The Control Handbook

Second Edition

CONTROL SYSTEM APPLICATIONS

Edited by

William S. Levine

CRC Press

Control System Applications

The Electrical Engineering Handbook Series

Series Editor

Richard C. Dorf

University of California, Davis

Titles Included in the Series

The Avionics Handbook, Second Edition, Cary R. Spitzer
The Biomedical Engineering Handbook, Third Edition, Joseph D. Bronzino
The Circuits and Filters Handbook, Third Edition, Wai-Kai Chen
The Communications Handbook, Second Edition, Jerry Gibson
The Computer Engineering Handbook, Vojin G. Oklobdzija
The Control Handbook, Second Edition, William S. Levine
CRC Handbook of Engineering Tables, Richard C. Dorf
Digital Avionics Handbook, Second Edition, Cary R. Spitzer
The Digital Signal Processing Handbook, Vijay K. Madisetti and Douglas Williams
The Electric Power Engineering Handbook, Second Edition, Leonard L. Grigsby
The Electrical Engineering Handbook, Third Edition, Richard C. Dorf
The Electronics Handbook, Second Edition, Jerry C. Whitaker
The Engineering Handbook, Third Edition, Richard C. Dorf
The Handbook of Ad Hoc Wireless Networks, Mohammad Ilyas
The Handbook of Formulas and Tables for Signal Processing, Alexander D. Poularikas
Handbook of Nanoscience, Engineering, and Technology, Second Edition, William A. Goddard, III, Donald W. Brenner, Sergey E. Lyshevski, and Gerald J. Iafrate
The Handbook of Optical Communication Networks, Mohammad Ilyas and Hussein T. Mouftah
The Industrial Electronics Handbook, J. David Irwin
The Measurement, Instrumentation, and Sensors Handbook, John G. Webster
The Mechanical Systems Design Handbook, Osita D.I. Nwokah and Yidirim Hurmuzlu
The Mechatronics Handbook, Second Edition, Robert H. Bishop
The Mobile Communications Handbook, Second Edition, Jerry D. Gibson
The Ocean Engineering Handbook, Ferial El-Hawary
The RF and Microwave Handbook, Second Edition, Mike Golio
The Technology Management Handbook, Richard C. Dorf
Transforms and Applications Handbook, Third Edition, Alexander D. Poularikas
The VLSI Handbook, Second Edition, Wai-Kai Chen

The Control Handbook

Second Edition

Edited by

William S. Levine

University of Maryland

College Park, MD, USA

Control System Fundamentals

Control System Applications

Control System Advanced Methods

Control System Applications

Edited by

William S. Levine

University of Maryland

College Park, MD, USA

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number: 978-1-4200-7360-7 (Hardback)

Library of Congress Cataloging-in-Publication Data

Control system applications / edited by William S. Levine. -- 2nd ed.
p. cm. -- (The electrical engineering handbook series)
Includes bibliographical references and index.
ISBN 978-1-4200-7360-7
1. Automatic control. 2. Control theory. I. Levine, W. S. II. Title.

TJ225.C66 2011
629.8--dc22 2010026364

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

SECTION III Industrial

SECTION IV Biological and Medical

SECTION V Electronics

SECTION VI Networks

SECTION VII Special Applications

Preface to the Second Edition

As you may know, the first edition of *The Control Handbook* was very well received. Many copies were sold and a gratifying number of people took the time to tell me that they found it useful. To the publisher, these are all reasons to do a second edition. To the editor of the first edition, these same facts are a modest disincentive. The risk that a second edition will not be as good as the first one is real and worrisome. I have tried very hard to insure that the second edition is at least as good as the first one was. I hope you agree that I have succeeded.

I have made two major changes in the second edition. The first is that all the *Applications* chapters are new. It is simply a fact of life in engineering that once a problem is solved, people are no longer as interested in it as they were when it was unsolved. I have tried to find especially inspiring and exciting applications for this second edition.

Secondly, it has become clear to me that organizing the *Applications* book by academic discipline is no longer sensible. Most control applications are interdisciplinary. For example, an automotive control system that involves sensors to convert mechanical signals into electrical ones, actuators that convert electrical signals into mechanical ones, several computers and a communication network to link sensors and actuators to the computers does not belong solely to any specific academic area. You will notice that the applications are now organized broadly by application areas, such as automotive and aerospace.

One aspect of this new organization has created a minor and, I think, amusing problem. Several wonderful applications did not fit into my new taxonomy. I originally grouped them under the title Miscellaneous. Several authors objected to the slightly pejorative nature of the term "miscellaneous." I agreed with them and, after some thinking, consulting with literate friends and with some of the library resources, I have renamed that section "Special Applications." Regardless of the name, they are all interesting and important and I hope you will read those articles as well as the ones that did fit my organizational scheme.

There has also been considerable progress in the areas covered in the *Advanced Methods* book. This is reflected in the roughly two dozen articles in this second edition that are completely new. Some of these are in two new sections, "Analysis and Design of Hybrid Systems" and "Networks and Networked Controls."

There have even been a few changes in the *Fundamentals*. Primarily, there is greater emphasis on sampling and discretization. This is because most control systems are now implemented digitally.

I have enjoyed editing this second edition and learned a great deal while I was doing it. I hope that you will enjoy reading it and learn a great deal from doing so.

William S. Levine

Acknowledgments

The people who were most crucial to the second edition were the authors of the articles. It took a great deal of work to write each of these articles and I doubt that I will ever be able to repay the authors for their efforts. I do thank them very much.

The members of the advisory/editorial board for the second edition were a very great help in choosing topics and finding authors. I thank them all. Two of them were especially helpful. Davor Hrovat took responsibility for the automotive applications and Richard Braatz was crucial in selecting the applications to industrial process control.

It is a great pleasure to be able to provide some recognition and to thank the people who helped bring this second edition of *The Control Handbook* into being. Nora Konopka, publisher of engineering and environmental sciences for Taylor & Francis/CRC Press, began encouraging me to create a second edition quite some time ago. Although it was not easy, she finally convinced me. Jessica Vakili and Kari Budyk, the project coordinators, were an enormous help in keeping track of potential authors as well as those who had committed to write an article. Syed Mohamad Shajahan, senior project executive at Techset, very capably handled all phases of production, while Richard Tressider, project editor for Taylor & Francis/CRC Press, provided direction, oversight, and quality control. Without all of them and their assistants, the second edition would probably never have appeared and, if it had, it would have been far inferior to what it is.

Most importantly, I thank my wife Shirley Johannesen Levine for everything she has done for me over the many years we have been married. It would not be possible to enumerate all the ways in which she has contributed to each and everything I have done, not just editing this second edition.

William S. Levine

Editorial Board

Editor

William S. Levine received B.S., M.S., and Ph.D. degrees from the Massachusetts Institute of Technology. He then joined the faculty of the University of Maryland, College Park where he is currently a research professor in the Department of Electrical and Computer Engineering. Throughout his career he has specialized in the design and analysis of control systems and related problems in estimation, filtering, and system modeling. Motivated by the desire to understand a collection of interesting controller designs, he has done a great deal of research on mammalian control of movement in collaboration with several neurophysiologists.

He is co-author of *Using MATLAB to Analyze and Design Control Systems*, March 1992. Second Edition, March 1995. He is the coeditor of *The Handbook of Networked and Embedded Control Systems,* published by Birkhauser in 2005. He is the editor of a series on control engineering for Birkhauser. He has been president of the IEEE Control Systems Society and the American Control Council. He is presently the chairman of the SIAM special interest group in control theory and its applications.

He is a fellow of the IEEE, a distinguished member of the IEEE Control Systems Society, and a recipient of the IEEE Third Millennium Medal. He and his collaborators received the Schroers Award for outstanding rotorcraft research in 1998. He and another group of collaborators received the award for outstanding paper in the *IEEE Transactions on Automatic Control*, entitled "Discrete-Time Point Processes in Urban Traffic Queue Estimation."

Contributors

Farhad Aghili
Division of Spacecraft Engineering
Canadian Space Agency
Saint-Hubert, Quebec, Canada

Juan C. Agüero
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan, New South Wales, Australia

Andrew Alleyne
Department of Mechanical Science and Engineering
University of Illinois at Urbana–Champaign
Urbana, Illinois

Anuradha M. Annaswamy
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts

Francis Assadian
Department of Automotive Engineering
Cranfield University
Cranfield, United Kingdom

John J. Baker
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan

Matthijs L. G. Boerlage
Renewable Energy Systems and Instrumentation
General Electric Global Research
Munich, Germany

Michael A. Bolender
Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Dominique Bonvin
Automatic Control Laboratory
Swiss Federal Institute of Technology in Lausanne
Lausanne, Switzerland

Francesco Borrelli
Department of Mechanical Engineering
University of California, Berkely
Berkeley, California

Richard D. Braatz
Department of Chemical Engineering
University of Illinois at Urbana–Champaign
Urbana, Illinois

Vikas Chandan
Department of Mechanical Science and Engineering
University of Illinois at Urbana–Champaign
Urbana, Illinois

Panagiotis D. Christofides
Department of Chemical and Biomolecular Engineering
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California

Francesco Alessandro Cuzzola
Danieli Automation
Buttrio, Italy

Raymond A. DeCarlo
Department of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

Josko Deur
Mechanical Engineering and Naval Architecture
University of Zagreb
Zagreb, Croatia

Jaspreet S. Dhupia
School of Mechanical and
Aerospace Engineering
Nanyang Technological University
Singapore

Stefano Di Cairano
Ford Motor Company
Dearborn, Michigan

David B. Doman
Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Thomas F. Edgar
Department of Chemical Engineering
University of Texas at Austin
Austin, Texas

Atilla Eryilmaz
Electrical and Computer
Engineering Department
The Ohio State University
Columbus, Ohio

Paolo Falcone
Signals and Systems Department
Chalmers University of Technology
Goteborg, Sweden

Thor I. Fossen
Department of Engineering Cybernetics
and
Centre for Ships and Ocean Structures
Norwegian University of Science and
Technology
Trondheim, Norway

Grégory François
Automatic Control Laboratory
Swiss Federal Institute of Technology in
Lausanne
Lausanne, Switzerland

Henri P. Gavin
Department of Civil and
Environmental Engineering
Duke University
Durham, North Carolina

Veysel Gazi
Department of Electrical and Electronics
Engineering
TOBB University of Economics and Technology
Ankara, Turkey

Hans P. Geering
Measurement and Control Laboratory
Swiss Federal Institute of Technology
Zurich, Switzerland

Alvaro E. Gil
Xerox Research Center
Webster, New York

Graham C. Goodwin
School of Electrical Engineering
and Computer Science
The University of Newcastle
Callaghan, New South Wales, Australia

Lino Guzzella
Swiss Federal Institute of Technology
Zurich, Switzerland

Michael A. Henson
Department of Chemical Engineering
University of Massachusetts Amherst
Amherst, Massachusetts

Raymond W. Holsapple
Control Science Center of Excellence
Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Seunghyuck Hong
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts

Karlene A. Hoo
Department of Chemical Engineering
Texas Tech University
Lubbock, Texas

Davor Hrovat
Research and Advanced Engineering
Ford Motor Company
Dearborn, Michigan

Gangshi Hu
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles
Los Angeles, California

Neera Jain
Department of Mechanical Science and Engineering
University of Illinois at Urbana–Champaign
Urbana, Illinois

Matthew R. James
College of Engineering and Computer Science
Australian National University
Canberra, Australia

Mrdjan Jankovic
Research and Advanced Engineering
Ford Motor Company
Dearborn, Michigan

Mustafa Khammash
Department of Mechanical Engineering
University of California, Santa Barbara
Santa Barbara, California

Ilya Kolmanovsky
Research and Advanced Engineering
Ford Motor Company
Dearborn, Michigan

Robert L. Kosut
SC Solutions
Sunnyvale, California

Rajesh Kumar
Department of Computer Science
Johns Hopkins University
Baltimore, Maryland

Katrina Lau
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan, New South Wales, Australia

Bin Li
Department of Mechanical Science and Engineering
University of Illinois at Urbana–Champaign
Urbana, Illinois

Mingheng Li
Department of Chemical and Materials Engineering
California State Polytechnic University
Pomona, California

Rongsheng (Ken) Li
The Boeing Company
El Segundo, California

Jianbo Lu
Research and Advanced Engineering
Ford Motor Company
Dearborn, Michigan

Stephen Magner
Research and Advanced Engineering
Ford Motor Company
Dearborn, Michigan

Amir J. Matlock
Department of Aerospace Engineering
University of Michigan
Ann Arbor, Michigan

Lalit K. Mestha
Xerox Research Center
Webster, New York

Roel J.E. Merry
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands

Marinus J. van de Molengraft
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands

Brian Munsky
CCS-3 and the Center for NonLinear Studies
Los Alamos National Lab
Los Alamos, New Mexico

Zoltan K. Nagy
Chemical Engineering Department
Loughborough University
Loughborough, United Kingdom

Jason C. Neely
Department of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

Babatunde Ogunnaike
Chemical Engineering
University of Delaware
Newark, Delaware

Michael W. Oppenheimer
Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Gerassimos Orkoulas
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles
Los Angeles, California

Rich Otten
Department of Mechanical Science and Engineering
University of Illinois at Urbana–Champaign
Urbana, Illinois

Thomas Parisini
Department of Electrical and Electronics Engineering
University of Trieste
Trieste, Italy

Kevin M. Passino
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, Ohio

Steven D. Pekarek
Department of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

Tristan Perez
School of Engineering
The University of Newcastle
Callaghan, New South Wales, Australia
and
Centre for Ships and Ocean Structures
Norwegian University of Science and Technology
Trondheim, Norway

Michael J. Piovoso
School of Graduate Professional Studies
Pennsylvania State University
Malvern, Pennsylvania

Giulio Ripaccioli
Department of Informatic Engineering
University of Siena
Siena, Italy

Charles E Rohrs
Rohrs Consulting
Newton, Massachusetts

Michael J. C. Ronde
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands

Melanie B. Rudoy
Department of Electrical and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts

Michael Santina
The Boeing Company
Seal Beach, California

Antonio Sciarretta
IFP Energies Nouvelles
Rueil-Malmaison, France

Jeff T. Scruggs
Department of Civil and Environmental Engineering
Duke University
Durham, North Carolina

Srinivas Shakkottai
Department of Electrical and Computer Engineering
Texas A & M University
College Station, Texas

Jason B. Siegel
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan

Eduardo I. Silva
Department of Electronic Engineering
Federico Santa María Technical University
Valparaíso, Chile

Masoud Soroush
Department of Chemical and Biological Engineering
Drexel University
Philadelphia, Pennsylvania

Anna G. Stefanopoulou
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan

Maarten Steinbuch
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, the Netherlands

Hongtei E. Tseng
Research and Advanced Engineering
Ford Motor Company

A. Galip Ulsoy
Department of Mechanical Engineering
University of Michigan
Ann Arbor, Michigan

M. Vidyasagar
Department of Bioengineering
The University of Texas at Dallas
Richardson, Texas

Meng Wang
School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan, New South Wales, Australia

Diana Yanakiev
Research and Advanced Engineering
Ford Motor Company
Dearborn, Michigan

Xinyu Zhang
Department of Chemical and Biomolecular Engineering
University of California, Los Angeles
Los Angeles, California

I

Automotive

1

Linear Parameter-Varying Control of Nonlinear Systems with Applications to Automotive and Aerospace Controls*

Hans P. Geering
Swiss Federal Institute of Technology

1.1 Introduction

In this chapter, a linear parameter-varying (LPV) plant $[A(\theta), B(\theta), C(\theta)]$ with the parameter vector θ is considered with continuously differentiable system matrices $A(\theta)$, $B(\theta)$, and $C(\theta)$. As described in Section 1.2, such a LPV plant description is typically obtained by linearizing the model of a nonlinear plant about a nominal trajectory. The control problem, which is considered in this chapter is finding a LPV continuous-time controller with the system matrices $F(\theta)$, $G(\theta)$, and $H(\theta)$ of its state–space model.

In Section 1.3, the control problem is formulated as an H_∞ problem using the mixed sensitivity approach. The shaping weights $W_e(\theta, s)$, $W_u(\theta, s)$, and $W_y(\theta, s)$ are allowed to be parameter-varying. The most appealing feature of this approach is that it yields a parameter-varying bandwidth $\omega_c(\theta)$ of the robust control system. Choosing appropriate shaping weights is described in Section 1.4. For more details about the design methodology, the reader is referred to [1–6].

* Parts reprinted from H. P. Geering, *Proceedings of the IEEE International Symposium on Industrial Electronics—ISIE 2005*, Dubrovnik, Croatia, June 20–23, 2005, pp. 241–246, © 2005. IEEE. With permission.

In Section 1.5, it is shown, how parameter-varying time-delays in the plant dynamics can be handled in the framework proposed in Sections 1.3 and 1.4. For more details, consult [5,7].

In Section 1.6, two applications in the area of automotive engine control are discussed. In the first application [4,5,8], the design of an LPV feedback controller for the fuel injection is shown, which is suitable over the whole operating envelope of engine.

In the second application [9,10], the philosophy of designing an LPV feedback controller is carried over to the problem of designing an additional LPV feedforward controller compensating the parameter-varying wall-wetting dynamics in the intake manifold of the port-injected gasoline engine.

In Section 1.7, the problem of LPV control of the short-period motion of an aircraft is discussed.

1.2 Statement of the Control Problem

We consider the following nonlinear time-invariant dynamic system ("plant") with the unconstrained input vector $U(t) \in R^m$, the state vector $X(t) \in R^n$, and the output vector $Y(t) \in R^p$:

$$\dot{X}(t) = f(X(t), U(t)),$$
$$Y(t) = g(X(t)),$$

where f and g are fairly "smooth" continuously differentiable functions.

Let us assume that we have found a reasonable or even optimal open-loop control strategy $U_{\text{nom}}(t)$ for a rather large time interval $t \in [0, T]$ (perhaps $T = \infty$), which theoretically generates the nominal state and output trajectories $X_{\text{nom}}(t)$ and $Y_{\text{nom}}(t)$, respectively.

In order to ensure that the actual state and output trajectories $X(t)$ and $Y(t)$ stay close to the nominal ones at all times, we augment the open-loop control $U_{\text{nom}}(t)$ with a (correcting) feedback part $u(t)$. Thus, the combined open-closed-loop input vector becomes

$$U(t) = U_{\text{nom}}(t) + u(t).$$

Assuming that the errors

$$x(t) = X(t) - X_{\text{nom}}(t) \quad \text{and} \quad y(t) = Y(t) - Y_{\text{nom}}(t)$$

of the state and output trajectories, respectively, can be kept minimum with small closed-loop corrections $u(t)$, allows us to design a linear (parameter-varying) output feedback controller based on the linearized dynamics of the plant:

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t),$$
$$y(t) = C(\theta)x(t),$$

where $A(\theta)$, $B(\theta)$, and $C(\theta)$ symbolically denote the following Jacobi matrices:

$$A(\theta) = \frac{\partial f}{\partial x}\big(X_{\text{nom}}(t), U_{\text{nom}}(t)\big),$$
$$B(\theta) = \frac{\partial f}{\partial u}\big(X_{\text{nom}}(t), U_{\text{nom}}(t)\big),$$
$$C(\theta) = \frac{\partial g}{\partial x}\big(X_{\text{nom}}(t)\big).$$

The symbol θ (or more precisely $\theta(t)$) denotes a parameter vector, by which the Jacobi matrices are parametrized; it contains the reference values $X_{\text{nom}}(t)$ and $U_{\text{nom}}(t)$ of the state and control vector, respectively, but it may also contain additional "exogenous" signals influencing the parameters of the

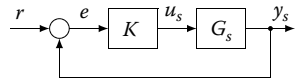

FIGURE 1.1 Schematic representation of the feedback control system.

nonlinear equations describing the dynamics of the plant (e.g., a temperature, which is not included in the model as a state variable).

By using the symbol θ rather than $\theta(t)$, we indicate that we base the design of the feedback controller on a time-invariant linearized plant at every instant t ("frozen linearized dynamics").

This leads us to posing the following problem of designing an LPV controller:

For all of the attainable values of the parameter vector θ, design a robust dynamic controller (with a suitable order n_c) with the state–space representation

$$\begin{aligned} z(t) &\in R^{n_c}, \\ \dot{z}(t) &= A_c(\theta)z(t) + B_c(\theta)e(t), \\ u_s(t) &= C_c(\theta)z(t), \end{aligned}$$

such that all of the specified quantitative, parameter-dependent performance, and robustness specifications are met (see Figure 1.1).

In Section 1.3, this rather general problem statement will be narrowed down to a suitable and transparent setting of H_∞ control and the solution will be presented.

1.3 LPV H_∞ Control

In this section, we consider the LPV time-invariant plant

$$\begin{aligned} \dot{x}_s(t) &= A_s(\theta)x_s(t) + B_s(\theta)u_s(t), \\ y_s(t) &= C_s(\theta)x_s(t) \end{aligned}$$

of order n_s. For the sake of simplicity, we assume that we have a "square" plant, that is, the number of output signals equals the number of input signals: $p_s = m_s$.

Furthermore, we assume that the input u_s, the state x_s, and the output y_s are suitably scaled, such that the singular values of the frequency response matrix $G_s(j\omega) = C_s[j\omega I - A_s]^{-1}B_s$ are not spread too wide apart.

For the design of the LPV time-invariant controller $K(\theta)$ depicted in Figure 1.1, we use the H_∞ method [1,2]. As a novel feature, we use parameter-dependent weights $W_\bullet(\theta, s)$. This allows in particular that we can adapt the bandwidth $\omega_c(\theta)$ of the closed-loop control system to the parameter-dependent properties of the plant!

Figure 1.2 shows the abstract schematic of the generic H_∞ control system. Again, $K(\theta)$ is the controller, which we want to design and $G(\theta, s)$ is the so-called augmented plant. The goal of the design is finding a compensator $K(\theta, s)$, such that the H_∞ norm from the auxiliary input w to the auxiliary output z is less

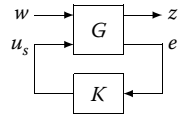

FIGURE 1.2 Schematic representation of the H_∞ control system.

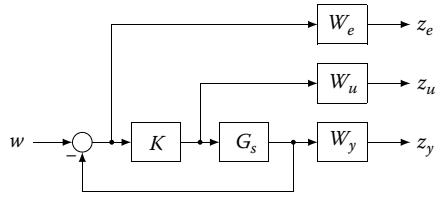

FIGURE 1.3 $S/KS/T$ weighting scheme.

than γ ($\gamma \leq 1$), that is,

$$\|T_{zw}(\theta, s)\|_\infty < \gamma \leq 1$$

for all of the attainable values of the constant parameter vector θ.

For the H_∞ design we choose the mixed-sensitivity approach. This allows us to shape the singular values of the sensitivity matrix $S(j\omega)$ and of the complementary sensitivity matrix $T(j\omega)$ of our control system (Figure 1.1), where

$$\begin{aligned} S(\theta, s) &= [I + G_s(\theta, s)K(\theta, s)]^{-1} \\ T(\theta, s) &= G_s(\theta, s)K(\theta, s)[I + G_s(\theta, s)K(\theta, s)]^{-1} \\ &= G_s(\theta, s)K(\theta, s)S(\theta, s). \end{aligned}$$

Thus, we choose the standard $S/KS/T$ weighting scheme as depicted in Figure 1.3. This yields the following transfer matrix:

$$T_{zw}(\theta, s) = \begin{bmatrix} W_e(\theta, s)S(\theta, s) \\ W_u(\theta, s)K(\theta, s)S(\theta, s) \\ W_y(\theta, s)T(\theta, s) \end{bmatrix}.$$

The augmented plant G (Figure 1.2) has the two input vectors w and u_s and the two output vectors z and e, where z consists of the three subvectors z_e, z_u, and z_y (Figure 1.3). Its schematic representation is shown in more detail in Figure 1.4.

In general, the four subsystems $G_s(\theta, s)$, $W_e(\theta, s)$, $W_u(\theta, s)$, and $W_y(\theta, s)$ are LPV time-invariant systems. By concatenating their individual state vectors into one state vector x, we can describe the dynamics of the augmented plant by the following state–space model:

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + \begin{bmatrix} B_1(\theta) & B_2(\theta) \end{bmatrix} \begin{bmatrix} w(t) \\ u_s(t) \end{bmatrix} \\ \begin{bmatrix} z(t) \\ e(t) \end{bmatrix} &= \begin{bmatrix} C_1(\theta) \\ C_2(\theta) \end{bmatrix} x(t) + \begin{bmatrix} D_{11}(\theta) & D_{12}(\theta) \\ D_{21}(\theta) & D_{22}(\theta) \end{bmatrix} \begin{bmatrix} w(t) \\ u_s(t) \end{bmatrix}. \end{aligned}$$

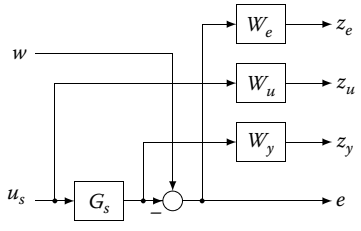

FIGURE 1.4 Schematic representation of the augmented plant.

The following conditions are necessary for the existence of a solution to the H_∞ control design problem*:

1. The weights W_e, W_u, and W_y are asymptotically stable.
2. The plant $[A_s, B_s]$ is stabilizable.
3. The plant $[A_s, C_s]$ is detectable.
4. The maximal singular value of D_{11} is sufficiently small: $\overline{\sigma}(D_{11}) < \gamma$.
5. Rank$(D_{12}) = m_s$, that is, there is a full feedthrough from u_s to z.
6. Rank$(D_{21}) = p_s = m_s$, that is, there is a full feedthrough to e from w.
7. The system $[A, B_1]$ has no uncontrollable poles on the imaginary axis.
8. The system $[A, C_1]$ has no undetectable poles on the imaginary axis.

Remarks

Condition 6 is automatically satisfied (see Figure 1.4). Condition 7 demands that the plant G_s has no poles on the imaginary axis. Condition 5 can be most easily satisfied by choosing W_u as a static system with a small, square feedthrough: $W_u(s) \equiv \varepsilon I$.

In order to present the solution to the H_∞ problem in a reasonably esthetic way, it is useful to introduce the following substitutions [3,4]:

$$
\begin{aligned}
B &= \begin{bmatrix} B_1 & B_2 \end{bmatrix} \\
D_{1\bullet} &= \begin{bmatrix} D_{11} & D_{12} \end{bmatrix} \\
\bar{R} &= \begin{bmatrix} D_{11}^{\mathrm{T}} D_{11} - \gamma^2 I & D_{11}^{\mathrm{T}} D_{12} \\ D_{12}^{\mathrm{T}} D_{11} & D_{12}^{\mathrm{T}} D_{12} \end{bmatrix} \\
\bar{S} &= B \bar{R}^{-1} B^{\mathrm{T}} \\
\bar{A} &= A - B \bar{R}^{-1} D_{1\bullet}^{\mathrm{T}} C_1 \\
\bar{Q} &= C_1^{\mathrm{T}} C_1 - C_1^{\mathrm{T}} D_{1\bullet} \bar{R}^{-1} D_{1\bullet}^{\mathrm{T}} C_1 \\
G &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \bar{R}^{-1} (B^{\mathrm{T}} K + D_{1\bullet}^{\mathrm{T}} C_1) \\
\widehat{R} &= I - \frac{1}{\gamma^2} D_{11}^{\mathrm{T}} D_{11} \\
\bar{\bar{R}} &= D_{21} \widehat{R}(\gamma)^{-1} D_{21}^{\mathrm{T}} \\
\bar{\bar{C}} &= C_2 - D_{21} G_1 + \frac{1}{\gamma^2} D_{21} \widehat{R}^{-1} D_{11}^{\mathrm{T}} D_{12} G_2 \\
\bar{\bar{S}} &= \bar{\bar{C}}^{\mathrm{T}} \bar{\bar{R}}^{-1} \bar{\bar{C}} - \frac{1}{\gamma^2} G_2^{\mathrm{T}} D_{12}^{\mathrm{T}} D_{12} G_2 - \frac{1}{\gamma^4} G_2^{\mathrm{T}} D_{12}^{\mathrm{T}} D_{11} \widehat{R}^{-1} D_{11}^{\mathrm{T}} D_{12} G_2 \\
\bar{\bar{A}} &= A - B_1 G_1 + \frac{1}{\gamma^2} B_1 \widehat{R}^{-1} D_{11}^{\mathrm{T}} D_{12} G_2 - B_1 \widehat{R}^{-1} D_{21}^{\mathrm{T}} \bar{\bar{R}}^{-1} \bar{\bar{C}} \\
\bar{\bar{Q}} &= B_1 \widehat{R}^{-1} B_1^{\mathrm{T}} - B_1 \widehat{R}^{-1} D_{21}^{\mathrm{T}} \bar{\bar{R}}^{-1} D_{21} \widehat{R}^{-1} B_1^{\mathrm{T}}.
\end{aligned}
$$

Remarks

The matrix K has the same dimension as A. The solution of the first algebraic matrix Riccati equation is given below. The matrix G has the same partitioning as B^{T}. The matrices $\bar{Q}$ and $\bar{\bar{Q}}$ will automatically be positive-semidefinite. Remember: all of these matrices are functions of the frozen parameter vector θ.

* In order to prevent cumbersome notation, the dependence on the parameter vector θ is dropped in the development of the results.

The solution of the H_∞ problem is described in the following.

Theorem 1.1:

The H_∞ problem has a solution for a given value $\gamma > 0$ if and only if the algebraic matrix Riccati equation

$$0 = K\bar{A} + \bar{A}^{\mathrm{T}}K - K\bar{S}K + \bar{Q}$$

has a positive-semidefinite stabilizing solution K and if and only if the algebraic matrix Riccati equation

$$0 = \bar{\bar{A}}P + P\bar{\bar{A}}^{\mathrm{T}} - P\bar{\bar{S}}P + \bar{\bar{Q}}$$

has a positive-semidefinite stabilizing solution P. ■

The controller $K(s)$ has the following state–space description:

$$\begin{aligned}\dot{z}(t) &= [A - B_1G_1 - B_2G_2 - H(C_2 - D_{21}G_1 - D_{22}G_2)]z(t) + He(t)\\ u_s(t) &= -G_2z(t)\end{aligned}$$

with the input matrix

$$H = \left[P\bar{\bar{C}}^{\mathrm{T}} + B_1\widehat{R}^{-1}D_{21}^{\mathrm{T}}\right]\bar{\bar{R}}^{-1}.$$

Remark

For sufficiently large values of γ, both of the algebraic matrix Riccati equations will have unique stabilizing solutions, such that $\bar{A} - \bar{S}K$ and $\bar{\bar{A}} - P\bar{\bar{S}}$ are stability matrices. We strive for a solution for $\gamma = 1$ for all of the attainable values of the parameter vector θ.

1.4 Choosing the LPV Weights $W_\bullet(\theta, s)$

For a parameter-independent SISO plant $G_s(s)$, a control engineer knows how to choose the bandwidth ω_c of the control system (Figure 1.1), that is, the cross-over frequency of the loop gain $|G_s(j\omega)K(j\omega)|$, after inspecting the magnitude plot $|G_s(j\omega)|$ of the plant.

He also knows how to choose quantitative specifications for performance via the sensitivity $|S(j\omega)|$ in the passband and for robustness via the complementary sensitivity $|T(j\omega)|$ in the rejection band and the peak values of $|S(j\omega)|$ (e.g., 3 dB ≈ 1.4) and $|T(j\omega)|$ (e.g., 1 dB ≈ 1.12) in the crossover region.

Since the weights W_e and W_y shape the sensitivity S and the complementary sensitivity T, respectively, the weights are chosen in the following way: (1) choose the functions $\overline{S}(\omega)$ and $\overline{T}(\omega)$ bounding the quantitative specifications for $|S(j\omega)|$ and $|T(j\omega)|$, respectively, from above, at all frequencies; (2) choose the weights W_e and W_y as the inverses of $\overline{S}$ and $\overline{T}$, respectively, such that $|W_e(j\omega)| \equiv 1/\overline{S}(\omega)$ and $|W_y(j\omega)| \equiv 1/\overline{T}(\omega)$ hold.

Example:

For the sake of simplicity, let $\overline{S}(\omega)$ correspond to a lead-lag element with $\overline{S}(0) = S_{\min} = 0.01$, $\overline{S}(\infty) = S_{\max} = 10$, and the corner frequencies $S_{\min}\omega_c$ and $S_{\max}\omega_c$; and let $\overline{T}(\omega)$ correspond to a lag-lead element with $\overline{T}(0) = T_{\max} = 10$, $T(\infty) = T_{\min} = 10^{-3}$, and the corner frequencies $\kappa\omega_c/T_{\max}$ and $\kappa\omega_c/T_{\min}$ with κ in the range $0.7 < \kappa \leq 2 \ldots 10$.

For a parameter-dependent SISO plant $G_s(\theta, s)$, for each value of the parameter vector θ, we proceed as described above in a *coherent* way. By coherent, we mean that we should "stress" the plant about equally strongly in each of the attainable operating points. Thus, we should probably have a pretty constant ratio $\omega_c(\theta)/\omega_n(\theta)$ of the chosen bandwidth of the closed-loop control system and the natural bandwidth $\omega_n(\theta)$ of the plant $G_s(\theta, s)$.

For a SISO plant, we obtain the following weights in the example described above:
For W_e the parameter-varying lag-lead element

$$W_e(\theta, s) = \frac{1}{\overline{S}_{\max}(\theta)} \frac{s + \overline{S}_{\max}(\theta)\omega_c(\theta)}{s + \overline{S}_{\min}(\theta)\omega_c(\theta)}$$

and for $W_y(s)$ the parameter varying lead-lag element

$$W_y(\theta, s) = \frac{1}{\overline{T}_{\min}(\theta)} \frac{s + \dfrac{\kappa(\theta)\omega_c(\theta)}{\overline{T}_{\max}(\theta)}}{s + \dfrac{\kappa(\theta)\omega_c(\theta)}{\overline{T}_{\min}(\theta)}}.$$

As already mentioned in Section 1.3, the weight W_u is usually chosen as a static element with a very small gain ε (e.g., $\varepsilon = 10^{-8}$) in order to automatically satisfy the condition 5:

$$W_u(\theta, s) \equiv \varepsilon.$$

In the case of a parameter-varying MIMO plant, the weights W_e, W_u, and W_y are square matrices because e, u_s, and y_s are vectors. In general, we use diagonal matrices with identical diagonal elements $w_e(\theta, s)$, $w_u(\theta, s)$, and $w_y(\theta, s)$, respectively.

If constant parameters $\overline{S}_{\min}$, $\overline{S}_{\max}$, $\overline{T}_{\max}$, $\overline{T}_{\min}$, and κ can be used, this is a good indication that the parameter dependence of the cross-over frequency $\omega_c(\theta)$ has been chosen in a coherent way.

1.5 Handling PV Time Delays

In many applications, modeling the linearized plant by a rational transfer matrix $G_s(\theta, s)$ only is not sufficient because the dynamics of the plant include significant, possibly parameter-varying, time delays $T_i(\theta)$.

A rational approximation of the transcendent transfer function e^{-sT} of a time delay can be obtained using the following Padé type approximation [5,7]:

$$e^{-sT} \approx \frac{\displaystyle\sum_{k=0}^{N} a_k(-sT)^k}{\displaystyle\sum_{k=0}^{N} a_k(sT)^k} \quad \text{with } a_k = \frac{(2N-k)!}{k!(N-k)!}.$$

The coefficients a_k can also be calculated recursively with the following scheme:

$$a_N = 1$$
$$a_{k-1} = \frac{k(2N+1-k)}{N+1-k} a_k \quad \text{for } k = N, \ldots, 1.$$

For choosing the order N of the approximation, we proceed along the following lines of thought [4]:

- The above Padé approximation is very good at low frequencies. The frequency ω^*, where the phase error is $\pi/6$ (30°), is $\omega^* \approx \frac{2N}{T}$.

- In order to suffer only a small loss of phase margin due to the Padé approximation error, we put ω^* way out into the rejection band. We propose $\omega^* = \alpha\omega_c$ with $\alpha = 30$.
- This leads to the following choice for the approximation order N:

$$N(\theta) = \left\lceil \frac{\alpha}{2}\omega_c(\theta)T(\theta) \right\rceil.$$

Obviously, choosing $\omega_c(\theta) \sim 1/T(\theta)$ would yield a parameter-independent approximation order N.

1.6 Applications in Automotive Engine Control

In this section, the LPV H_∞ methodology for the design of a LPV controller is applied to two problems of fuel control for a 4-stroke, spark-ignited, port-injected gasoline engine: In Section 1.6.1, we discuss the model-based feedback control, and in Section 1.6.2, the model-based feedforward control.

As usual, in control, the requirements for the accuracy of the mathematical models upon which the control designs are based differ significantly between feedback control and feedforward control.

For the design of a robust feedback control for an asymptotically stable plant, once we have chosen a controller, glibly speaking, it suffices to know the following data with good precision: the crossover frequency ω_c, the (sufficiently large) phase margin φ, and the (acceptable) direction of the tangent to the Nyquist curve at ω_c. In other words, a rather crude model of the plant just satisfying these requirements will do.

Conversely, for the design of a feedforward control, a rather precise model of the plant is needed because, essentially, the feedforward controller is supposed to invert the dynamics of the plant. The feedback part of the control scheme will then be mainly responsible for stability and robustness, besides further improving the command tracking performance.

1.6.1 Feedback Fuel Control

In this example, we want to design a feedback controller. Its task is keeping the air to fuel ratio of the mixture in the cylinders stoichiometric.

The fuel injection is governed by the control law $t_i = \beta(n, m)U$. Here, t_i is the duration of the injection impulse, n the engine speed, and m the mass of air in a cylinder (calculated by the input manifold observer). The function $\beta(n, m)$ is defined in such a way that the dimensionless control variable U is nominally 1 at every static operating point: $U_{\text{nom}}(n, m) \equiv 1$.

With a wide-range λ-sensor, the resulting air to fuel ratio is measured in the exhaust manifold of the engine. Its signal Λ is proportional to the air to fuel ratio and scaled such that $\Lambda = 1$ corresponds to stoichiometric. Hence, $\Lambda_{\text{nom}} \equiv 1$.

We want to find a robust compensator $K(\theta, s)$, such that small changes $u_s(t)$ in the control $U(t) = U_{\text{nom}} + u_s(t)$ will keep the errors $\lambda_s(t) = \Lambda(t) - \Lambda_{\text{nom}}$ minimum even in an arbitrary transient operation of the engine. Obviously, the parameter vector describing the operating point is $\theta(t) = [n(t), m(t)]$.

For modeling the linearized dynamics of the fuel path of the engine, the following phenomena must be considered: the wall-wetting dynamics in the intake manifold, turbulent mixing of the gas in the exhaust manifold, the dynamics of the λ-sensor, and (last but most important) the time delay between the injection of the fuel and the arrival of the corresponding mixture at the position of the λ-sensor. The simplest model we can get away with successfully is

$$G_s(\theta, s) = -e^{-sT(\theta)} \frac{1}{\tau(\theta)s + 1}.$$

The design of the controller proceeds in the following steps:

- Identify the functions $T(\theta)$ for the time delay and $\tau(\theta)$ for the time constant over the full operating envelope of the engine. This is done by measuring the step response of the engine to a step change in the amount of injected fuel over a sufficiently fine mesh of the parameters $\theta = [n, m]$.
- Choose a function $\omega_c(T, \tau)$ for the parameter-varying bandwidth of the control system. (Please note the change of variables!)
- Choose the weighting functions $W_e(T, \tau, s)$, $W_u(s)$, and $W_y(T, \tau, s)$.
- Choose an approximation order $N(T)$ for the Padé approximation of e^{-sT}. This yields a rational approximate transfer function $\widetilde{G}_s(T, \tau, s)$ of the plant.
- Solve the H_∞ problem for every pair (T, τ).
- Reduce the order of the resulting compensator $\widetilde{K}(T, \tau, s)$ by one successively, watching the resulting Nyquist curves and stop before the Nyquist curve deforms significantly. For practical purposes, the reduced order of the resulting final compensators should be constant over T and τ.
- Find a structurally suitable representation for the reduced-order transfer function $K(T, \tau, s)$, so that its parameters, say $k_i(T, \tau)$, can continuously and robustly be mapped over T and τ.

The engine control operates as follows in real time: At each control instant, that is, for each upcoming cylinder requesting its injection signal t_i, the instantaneous engine speed n and air mass m in the cylinder are available and the following steps are taken:

- Calculate $T(n, m)$ and $\tau(n, m)$.
- Calculate the parameters $k_i(T, \tau)$ of the continuous-time controller with the transfer function $K(T, \tau, s)$.
- Discretize the controller to discrete-time.
- Process one time step of the discrete-time controller and output the corresponding signal t_i.

Remark

Note the fine point here: As the engine speed changes, the time increment of the discrete-time controller changes.

For a BMW 1.8-liter 4-cylinder engine, the time delay T and the time constant τ were found to be in the ranges $T = 0.02 \ldots 1.0$ s and $\tau = 0.01 \ldots 0.5$ s over the full operating envelope of the engine. In [4], the bandwidth was chosen as $\omega_c(T, \tau) = \pi/6T$. With $\alpha = 30$, this resulted in the constant order $N = 8$ for the Padé approximation of the time delay.

For more details about these LPV feedback fuel control schemes, the reader is referred to [4,5,8,11], and [12, ch. 4.2.2].

1.6.2 Feedforward Fuel Control

In this example, we want to design a feedforward controller. Its task is inverting the wall-wetting dynamics of the intake manifold, such that, theoretically, the air to fuel ratio $\Lambda(t)$ never deviates from its nominal value $\Lambda_{\text{nom}} = 1$ in dynamic operation of the engine.

In 1981, Aquino published an empirical model for the wall-wetting dynamics [13], that is, for the dynamic mismatch between the mass m_{Fi} of fuel injected and the mass m_{Fo} of fuel reaching the cylinder:

$$m_{Fo}(s) = \left(1 - \kappa + \frac{\kappa}{s\tau + 1}\right) m_{Fi}(s).$$

This is a good model in the sense that it captures the balance of the fuel mass flow into the wall-wetting fuel puddle in the intake manifold and the fuel mass flow released by it again. The problem is, that for any given engine, the fraction κ and the time constant τ are strongly dependent on the temperatures of the air, of the fuel, and of the intake manifold wall, and the air mass flow $\dot{m}$, the intake manifold pressure, and so on.

Therefore, Aquino's model lends itself to the model-based design of a feedforward fuel controller. But its parameters should be derived using first physical principles.

Mathematical models derived by this approach and the corresponding designs of parameter-varying feedforward fuel controllers have been published in [9,10,23–25]. A summary of the model can be found in [12, ch. 2.4.2].

1.7 Application in Aircraft Flight Control

In this section, the LPV H_∞ methodology for the design of a LPV controller is applied to the control of the short-period motion of a small unmanned airplane.

For controlling an aircraft in a vertical plane, the following two physical control variables are available: The thrust F (for control in the "forward" direction) and the elevator angle δ_e (for angular control around the pitch axis).

In most cases, the airplane's pitch dynamics can be separated into a fast mode (from the elevator angle δ_e to the angle of attack α) and a slow mode (from the angle of attack α to the flight path angle γ). Therefore, for flight control, it is more useful to use the angle of attack as a control variable (instead of δ_e). In this case, the fast dynamics from δ_e to α should be considered in the flight control design as actuator dynamics from the commanded angle of attack α_{com} to the actual angle of attack of the aircraft in a suitable way. These "actuator dynamics" are usually associated with the notion of "short-period motion" because a step in the elevator angle produces a damped oscillatory response of the angle of attack.

Thus, we get the following control problem for the fast inner control loop of the overall control scheme, that is, the problem of controlling the short-period motion:

For the under-critically damped second-order parameter-varying system with the input δ_e, the output α, and the transfer function $G_{\alpha\delta_e}(\theta, s)$, find a parameter-varying controller with the transfer function $K(\theta, s)$, such that the command-following system

$$\alpha(s) = \frac{K(\theta, s)G_{\alpha\delta_e}(\theta, s)}{1 + K(\theta, s)G_{\alpha\delta_e}(\theta, s)}\,\alpha_{\text{com}}(s)$$

is robust and performs in a satisfactory way over the full operating envelope described by the parameter vector $\theta = (v, h)$ with the velocity $v = v_{\min} \ldots v_{\max}$ and the altitude $h = h_{\min} \ldots h_{\max}$.

In [14,15], a small unmanned airplane with a takeoff mass of 28 kg, and a wingspan of 3.1 m, and an operating envelope for the velocity of $v = 20 \ldots 100$ m/s and for the altitude of $h = 0 \ldots 800$ m has been investigated for robust, as well as fail-safe flight control.

The transfer function $G_{\alpha\delta_e}(\theta, s)$ can be written in the form

$$G_{\alpha\delta_e}(\theta, s) = G_{\alpha\delta_e}(\theta, 0)\,\frac{s_1(\theta)s_2(\theta)}{(s - s_1(\theta))(s - s_2(\theta))}.$$

Over the full flight envelope, the poles $s_1(\theta)$, $s_2(\theta)$ (in rad/s) and the steady-state gain $G_{\alpha\delta_e}(\theta, 0)$ can be parametrized with *very* high precision as follows:

$$s_{1,2}(v, h) = (c_1 + c_2 h)v \pm j(c_3 + c_4 h)v,$$

$$G_{\alpha\delta_e}(v, h, 0) = \frac{c_5}{1 + c_6 h}$$

with

$$c_1 = -1.47 \times 10^{-1}\ \text{rad/m}$$
$$c_2 = 1.37 \times 10^{-5}\ \text{rad/m}^2$$
$$c_3 = 7.01 \times 10^{-2}\ \text{rad/m}$$

$$c_4 = -3.08 \times 10^{-6} \text{ rad/m}^2$$
$$c_5 = 1.20$$
$$c_6 = -7.47 \times 10^{-5} \text{ m}^{-1}.$$

For manually piloting the unmanned airplane, it is desirable that the dynamic and static characteristics of the command following control from α_{com} to α be parameter-independent over the full flight envelope. This can easily be achieved by using the $S/KS/T$ weighting scheme (Figure 1.3) with parameter-independent weights W_e, W_u, and W_y.

$$W_e(\theta, s) = \frac{1}{\overline{S}_{\max}(\theta)} \frac{s + \overline{S}_{\max}(\theta)\omega_c(\theta)}{s + \overline{S}_{\min}(\theta)\omega_c(\theta)}$$

$$W_y(\theta, s) = \frac{1}{\overline{T}_{\min}(\theta)} \frac{s + \dfrac{\kappa(\theta)\omega_c(\theta)}{\overline{T}_{\max}(\theta)}}{s + \dfrac{\kappa(\theta)\omega_c(\theta)}{\overline{T}_{\min}(\theta)}}$$

$$W_u(\theta, s) \equiv \varepsilon.$$

For the following choice of the constant parameters of the weights:

$$\omega_c(\theta) \equiv 2 \text{ rad/s}$$
$$\kappa(\theta) \equiv 1.5$$
$$S_{\min}(\theta) \equiv 0.01$$
$$S_{\max}(\theta) \equiv 10$$
$$T_{\max}(\theta) \equiv 100$$
$$T_{\min}(\theta) \equiv 0.001$$
$$\varepsilon(\theta) \equiv 10^{-4},$$

a suitable unit step response from α_{com} to α with a rise time of about one second and no overshoot results for all $v = 20 \ldots 100$ m/s and $h = 0 \ldots 800$ m.

1.8 Conclusions

In this chapter, some concepts of LPV H_∞ control for a LPV plant $A(\theta), B(\theta), C(\theta)$ have been presented in detail. The salient feature is choosing a parameter-varying specification $\omega_c(\theta)$ for the bandwidth of the control system. Furthermore, parameter-varying weighting functions $W_\bullet(\theta, s)$ have been stipulated.

Applying these concepts has been discussed briefly for both LPV feedback and LPV feedforward control in the area of fuel control of an automotive engine, as well as for LPV control of the short-period motion in aircraft flight control.

In this chapter, the parameter θ has been assumed to be "frozen", that is, its truly time-varying nature, $\theta(t)$, has been neglected. Naturally, the question arises whether such an LVP control system will be asymptotically stable and sufficiently robust even during rapid changes of the parameter vector. Presently, there is a lot of research activity worldwide addressing the question of time-varying parameters, see [16–22] for instance. Suffice it to say here that in our examples about fuel injection of an automotive engine and about the control of the short-period motion of an airplane, the bandwidth of the dynamics of $\theta(t)$ is at least an order of magnitude smaller than the bandwidth of our control systems, even in severe transient operation of the engine or the airplane, respectively!

References

1. J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, State-space solutions to standard H_2 and H_∞ control problems, *IEEE Transactions on Automatic Control*, vol. 34, pp. 831–847, 1989.
2. U. Christen, *Engineering Aspects of H_∞ Control*, ETH dissertation no. 11433, Swiss Federal Institute of Technology, Zurich, Switzerland, 1996.
3. H. P. Geering, *Robuste Regelung*, 3rd ed., IMRT-Press, Institut für Mess- und Regeltechnik, ETH-Zentrum, Zurich, Switzerland, 2004.
4. H. P. Geering and C. A. Roduner, Entwurf robuster Regler mit der H_∞ Methode, *Bulletin SEV/VSE*, no. 3, pp. 55–58, 1999.
5. C. A. Roduner, *H_∞-Regelung linearer Systeme mit Totzeiten*, ETH dissertation no. 12337, Swiss Federal Institute of Technology, Zurich, Switzerland, 1997.
6. U. Christen, Calibratable model-based controllers, in *Proceedings of the IEEE Conference on Control Applications*, Glasgow, Scotland, October 2002, pp. 1056–1057.
7. J. Lam, Model reduction of delay systems using Padé approximants, *International Journal of Control*, vol. 57, no. 2, pp. 377–391, 1993.
8. C. A. Roduner, C. H. Onder, and H. P. Geering, Automated design of an air/fuel controller for an SI engine considering the three-way catalytic converter in the H_∞ approach, in *Proceedings of the 5th IEEE Mediterranean Conference on Control and Systems*, Paphos, Cyprus, July 1997, paper S5-1, pp. 1–7.
9. M. A. Locatelli, *Modeling and Compensation of the Fuel Path Dynamics of a Spark Ignited Engine*, ETH dissertation no. 15700, Swiss Federal Institute of Technology, Zurich, Switzerland, 2004.
10. M. Locatelli, C. H. Onder, and H. P. Geering, An easily tunable wall-wetting model for port fuel injection engines, in *SAE SP-1830: Modeling of Spark Ignition Engines*, March 2004, pp. 285–290.
11. E. Shafai, C. Roduner, and H. P. Geering, Indirect adaptive control of a three-way catalyst, in *SAE SP-1149: Electronic Engine Controls*, February 1996, pp. 185–193.
12. L. Guzzella and C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems.* London: Springer, 2004.
13. C. F. Aquino, Transient A/F characteristics of the 5 liter central fuel injection engine, *1981 SAE International Congress, SAE paper 810494*, Detroit, MI, March 1981.
14. M. R. Möckli, *Guidance and Control for Aerobatic Maneuvers of an Unmanned Airplane*, ETH dissertation no. 16586, Swiss Federal Institute of Technology, Zurich, Switzerland, 2006.
15. G. J. J. Ducard, *Fault-Tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*, ETH dissertation no. 17505, Swiss Federal Institute of Technology, Zurich, Switzerland, 2007.
16. J. S. Shamma and M. Athans, Analysis of gain scheduled control for nonlinear plants, *IEEE Transactions on Automatic Control*, vol. 35, pp. 898–907, 1990.
17. R. A. Hyde and K. Glover, The application of scheduled H_∞ controllers to a VSTOL aircraft, *IEEE Transactions on Automatic Control*, vol. 38, pp. 1021–1039, 1993.
18. G. Becker and A. Packard, Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback, *Systems & Control Letters*, vol. 23, pp. 205–215, 1994.
19. D. A. Lawrence and W. J. Rugh, Gain scheduling dynamic linear controllers for a nonlinear plant, *Automatica*, vol. 31, pp. 381–390, 1995.
20. P. Apkarian, P. Gahinet, and G. Becker, Self-scheduled H_∞ control of linear parameter-varying systems: A design example, *Automatica*, vol. 31, pp. 1251–1261, 1995.
21. P. Apkarian and R. J. Adams, Advanced gain-scheduling techniques for uncertain systems, *IEEE Transactions on Control Systems Technology*, vol. 6, pp. 21–32, 1998.
22. F. Bruzelius, *Linear Parameter-Varying Systems*, Ph.D. dissertation, Chalmers University of Technology, Göteborg, Sweden, 2004.
23. C. H. Onder and H. P. Geering, Measurement of the wall-wetting dynamics of a sequential injection spark ignition engine, in *SAE SP-1015: Fuel Systems for Fuel Economy and Emissions*, March 1994, pp. 45–51.
24. C. H. Onder, C. A. Roduner, M. R. Simons, and H. P. Geering, Wall-wetting parameters over the operating region of a sequential fuel injected SI engine, in *SAE SP-1357: Electronic Engine Controls: Diagnostics and Controls*, February 1998, pp. 123–131.
25. M. R. Simons, M. Locatelli, C. H. Onder, and H. P. Geering, A nonlinear wall-wetting model for the complete operating region of a sequential fuel injected SI engine, in *SAE SP-1511: Modeling of SI Engines*, March 2000, pp. 299–308.

2

Powertrain Control*

Davor Hrovat
Ford Motor Company

Mrdjan Jankovic
Ford Motor Company

Ilya Kolmanovsky
Ford Motor Company

Stephen Magner
Ford Motor Company

Diana Yanakiev
Ford Motor Company

2.1 Introduction

Automotive controls and associated microcomputers and embedded software represent one of the most active industrial R&D areas. Within this ever expanding area it is no surprise that powertrain controls have evolved as the most widespread and matured branch since, traditionally, the first microcomputer applications started with engine controls. For example, the first four generations of automotive computers found in Ford vehicles during the 1970s and early 1980s were all used for on-board engine controls (Powers, 1993). The first chassis/suspension and vehicle control applications then followed in the mid-1980s.

Early powertrain control applications were driven by U.S. regulatory requirements for improved fuel economy and reduced emissions. Additional benefits included improved functionality, performance, drivability, reliability, and reduced time-to-market as facilitated by the inherent flexibility of computers and associated software. Thus, in the mid-1970s, American automotive manufacturers introduced microprocessor-based engine control systems to meet the sometimes conflicting demands of high fuel economy and low emissions.

Present-day engine control systems contain many inputs (e.g., pressures, temperatures, rotational speeds, exhaust gas characteristics) and outputs (e.g., spark timing, exhaust gas recirculation, fuel-injector pulse widths, throttle position, valve/cam timing). The unique aspect of the automotive control is the requirement to develop systems that are relatively low in cost, which will be applied to several hundred thousand units in the field, that must work on automobiles with inherent manufacturing variability, which will be used by a spectrum of human operators, and are subject to irregular maintenance and varying operating conditions. This should be contrasted with the aircraft/spacecraft control problem, for which many of the sophisticated control techniques have been developed. In this case, we can enumerate nearly an opposite set of conditions.

* Figures from this chapter are available in color at http://www.crcpress.com/product/isbn/9781420073607

The software structure of the (embedded) controllers that have been developed to date is much like those in the other areas (i.e., aircraft controllers, process controllers) in that there exists an "outer-loop" operational mode structure, in this case, typically provided by a driver whose commands—for example, gas pedal position—are then interpreted as commands or reference signals for subsequent control agents. The latter typically consist of feedforward–feedback-adaptive (learning) modules. Traditionally, the feedforward or "open-loop" portion has been by far the most dominant with numerous logical constructs such as "if–then–else" contingency statements and related 2D and 3D tables, myriads of parameters, and online models of underlying physics and devices. More precisely, the feedforward component may include inverse models of relevant components and related physics such as nonlinear static formulas for flow across the throttle actuator, for example.

Assuming the existence of the microcomputer module with given chronometric and memory capabilities, the structured/disciplined approach to developing a total embedded control system typically involves the following major steps: (1) development of requirements; (2) development of appropriate linear and nonlinear plant models; (3) preliminary design with linear digital control system methods; (4) nonlinear simulation/controller design; and (5) hardware-in-the-loop/real-time simulation capability for identification, calibration, and verification including confirmation that the above chronometric and memory constraints have not been violated. This includes appropriate dynamometer and vehicle testing with the help of rapid prototyping, autocoding, and data collection and manipulation tools, as needed.

Typical powertrain control strategies contain several hundred thousand lines of C code and thousands of associated parameters and calibration variables requiring thousands of man-hours to calibrate, although the ongoing efforts in "self-calibrating" approaches and tools may substantially reduce this time-consuming task. In addition, different sensors and actuators—such as Electronic Throttle Control (ETC), Variable Cam Timing (VCT), and Universal Exhaust Gas Oxygen (UEGO) sensors—are constantly added and upgraded. So are the new functions and requirements. When computer memory and/or chronometric capabilities have been exhausted and new or improved functionality is needed, this can lead to development of new requirements for the next generation of engine computer modules. For example, in the period 1977–1982, Ford Motor Company introduced four generations of Engine Control Computers (EEC) of ever increasing capabilities. This evolution was needed to address more demanding fuel economy and pollution requirements, while at the same time for introducing new and/or more sophisticated functionality and for improved performance (Powers, 1993).

The five control strategy development steps mentioned above may not always be sequential and some of the steps can be omitted. Moreover, in practice, there are implied iteration loops between different steps. For example, one starts Step 1 with the best available requirements at the time, which can subsequently be refined as one progresses through Steps 2 through 5. Similarly, the models from Step 2 can be further refined as a result of Step 5 hardware implementation and testing.

In some cases, the detailed nonlinear models of a plant or component may already exist and can be used (with possible simplifications) to design a nonlinear plant-based controller directly. Alternatively, the detailed nonlinear model can be used to extract simplified linearized models with details that are relevant for the linear control system design within the bandwidth of interest. The model development, Step 2, may itself include models based on first principles (physics-based models) or semiempirical and identification-based models ("gray" and "black" boxes). Each approach has its advantages and disadvantages. The black and gray box models rely heavily on actual experimental data so that by default they are "validated" and typically require lesser time to develop. On the other hand, physically based models allow for (at least preliminary) controller design even before an actual hardware/plant is built and in the case of open-loop unstable systems they are needed to devise an initial stabilizing controller. Moreover, they can be an invaluable source of insight and critical information about the plant *modus operandi*, especially when dealing with sometimes elusive dynamic effects. In this context, they can influence the overall system design—both hardware and software—in a true symbiotic way.

Before we focus on concrete applications of powertrain controls, it is first important to enumerate what the main drivers and goals of such control systems are. This is briefly summarized in the next section.

2.2 Powertrain Controls and Related Attributes

Modern powertrains must satisfy numerous often competing requirements so that the design of a typical powertrain control system involves tradeoffs among a number of attributes (Hrovat and Powers, 1988, 1990). When viewed in a control theory context, the various attributes are categorized quantitatively as follows:

- *Emissions:* A set of terminal or final time inequality constraints (e.g., in case of gasoline engine, this would apply to key pollution components: NO_x, CO, and HC over certification drive cycles).
- *Fuel consumption:* A scalar quantity to be minimized over a drive cycle is usually the objective function to be minimized.
- *Driveability:* Expressed as constraints on key characteristic variables such as the damping ratio of dominant vibration modes or as one or more state variable inequality constraints, which must be satisfied at every instant on the time interval (e.g., wheel torque or vehicle acceleration should be within a certain prescribed band).
- *Performance:* Either part of the objective function or an intermediate point constraint, for example, achieve a specified 0–60 mph acceleration time.
- *Reliability:* As a part of the emission control system, the components in the computer control system (sensors, actuators, and computers) have up to 150,000 mile or 15-year warranty for Partial Zero-Emission Vehicle (PZEV)-certified vehicles. In the design process, reliability can enter as a sensitivity or robustness condition, for example, location of roots in the complex plane, or more explicitly as uncertainty bounds and weighted sensitivity/complimentary sensitivity bounds in the context of H_∞ or μ-synthesis and analysis methodology.
- *Cost:* The effects of cost are problem dependent. Typical ways that costs enter the problem quantitatively are increased weights on control variables in quadratic performance indices (which implies relatively lower-cost actuators) and output instead of state feedback (which implies fewer sensors but more software).
- *Packing:* Networking of computers and/or smart sensors and actuators requires distributed control theory and tradeoffs among data rates, task partitioning, and redundancy, among others.
- *Electromagnetic interference:* This is mainly a hardware problem, which is rarely treated explicitly in the analytic control design process.
- *Tamper-proof:* This is one of the reasons for computer control, and leads to adaptive/self-calibrating systems so that dealer adjustments are not required as the powertrain ages or changes.

To illustrate how control theoretic techniques are employed in the design of powertrain control systems, the examples of typical engine, transmission, and driveline controls will be reviewed along with some important related considerations such as drivability and diagnostics. This chapter concludes with a discussion of current trends in on-board computer control diagnostics.

2.3 Engine Control

Over the last decade, passenger vehicle emission regulations have become stricter by a factor between 5 (based on comparing Tier I and Tier II bin 5 EPA emission standards) and 30 (comparing Tier I and Tier II bin 2, i.e., Super Ultra Low Emission Vehicle, standards). This has forced automakers to increase the exhaust after-treatment catalyst's size and its precious metal loading, invent new procedures to start a cold engine, introduce a separate cold engine emission-reduction operating mode, and develop a system that accurately controls and coordinates air–fuel ratio, throttle position, spark timing, and VCT. In addition, the system operation has to be monitored by an on-board diagnostic (OBD) system that is designed to respond to any malfunction which can cause an increase in emissions larger than a specified threshold.

To improve fuel economy and performance, the automakers have added new engine devices and included new modes of operation. Today's production engines include ETC and VCT as the standard

hardware. Additional devices such as variable valve lift, variable displacement (also called cylinder deactivation or displacement-on-demand), charge motion control valves, intake manifold tuning, turbochargers, superchargers, and so on are also used in production applications.

Each device operation is computer controlled. Finding the steady-state set-point combinations that achieve the best tradeoff between fuel economy, peak torque/power, and emissions is the subject of the engine mapping and optimization process. Maintaining the device output to the desired set-point typically requires development of a local feedback control system for each one. Engines spend a significant fraction of time in transients. Because the optimization devices may interact in unexpected or undesirable ways, it is important to control and synchronize their transient behavior.

Figure 2.1 shows the view of an engine as the system to be controlled. The prominent feature is that the disturbance input, engine speed*, and ambient conditions are measured or known, while the performance variables, actual torque, emissions, and fuel efficiency are typically not available (except during laboratory testing). The reference set-point, engine torque demand, is available based on the accelerator pedal position. In the next three sections, we briefly review important paradigms of engine control system design: fuel consumption set-point optimization and feedback regulation. In each case, an advanced optimization or control method has been tried. In each case, experimental tests were run to confirm the achieved benefit.

2.3.1 Fuel-Consumption Optimization

In today's engines, several optimization devices are added to improve an attribute such as fuel economy. In most cases, the optimal set-point for each device varies with engine operating conditions and is usually found experimentally. Combining these devices has made it increasingly difficult and time consuming to map and calibrate such engines. The complexity increases not linearly, but exponentially with the number of devices, that is, degrees of freedom. Each additional degree of freedom typically increases the complexity in terms of mapping time and size of the calibration tables by a factor between 2 (for two position devices) and 3–10 (for continuously variable devices). For a high-degree-of-freedom (HDOF)

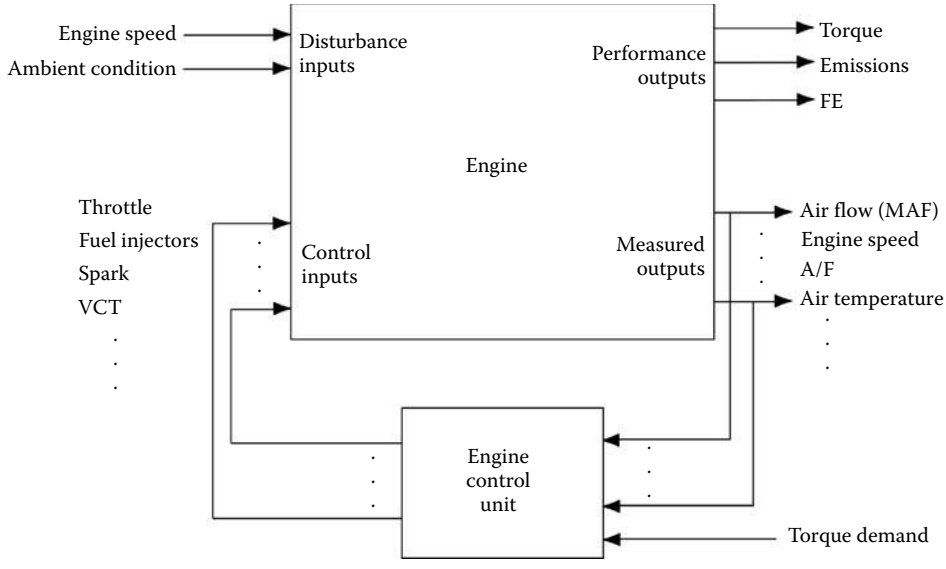

FIGURE 2.1 The input–output structure of a typical engine control system.

* Engine speed can be viewed as a disturbance input in some modes of operation and a system state in others (such as engine idle).

engine, the conventional process that generates the final calibration has become very time consuming. For example, in the dual-independent variable cam timing (diVCT) gasoline engine, which we shall use as the platform for the results in this section, the mapping time could increase by a factor of 30 or more over the conventional (non-VCT) engine or the process could end up sacrificing the potential benefit. From the product development point of view, either outcome is undesirable.

The diVCT engine has the intake and exhaust cam actuators that can be varied independently (Jankovic and Magner, 2002; Leone et al., 1996). A typical VCT hardware is shown in Figure 2.2. The intake valve opening (IVO) and exhaust valve closing (EVC) expressed in degrees after top dead center (ATDC) (TDC is at 360 deg crank in Figure 2.2) are considered the two independent degrees of freedom. The experimental 3.0L V6 engine under consideration has the range of −30 to 30 deg ATDC for IVO and 0 to 40 deg ATDC for EVC.

2.3.1.1 Engine Drive Cycle and Pointwise Optimization

We consider the problem of optimizing vehicle fuel efficiency while assuring that specific emissions regulations, defined over a drive cycle, are satisfied. A drive cycle, in which the vehicle speed and conditions are specified by regulations, is intended to evaluate vehicle emissions or fuel consumption under a generic

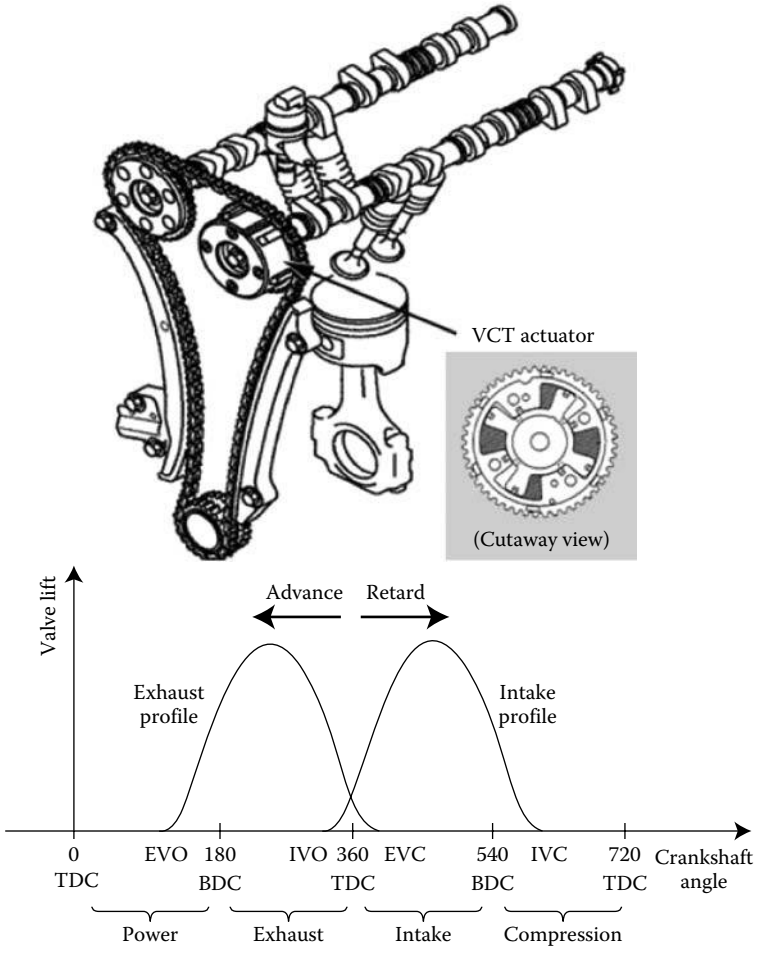

FIGURE 2.2 Valve lift profiles versus crank angle in a VCT engine.

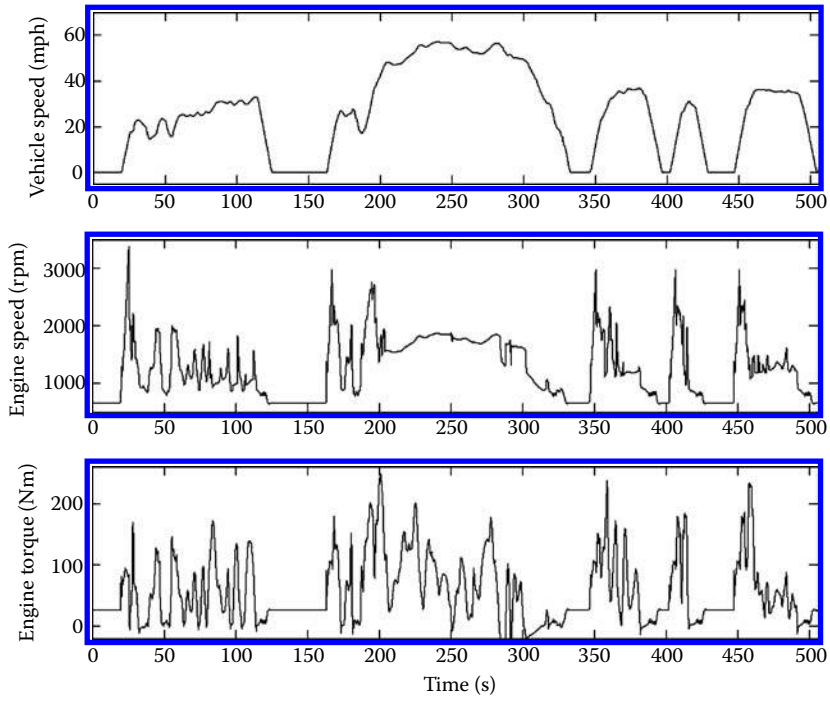

FIGURE 2.3 Vehicle speed, engine speed, and torque, during the first and last 505 s (bags 1 and 3) of the US75 drive cycle.

or particular drive pattern. The top plot in Figure 2.3 shows the vehicle speed profile during a part of the US75 drive cycle. Given the transmission shift schedule, the vehicle speed uniquely determines the engine speed and torque needed to follow the drive trace. Hence, from the point of view of engine optimization, engine speed (middle plot) and engine torque (bottom plot) are constrained variables.

Due to the presence of three-way catalysts, which become very efficient in removing regulated exhaust gases after light-off (50–100 s from a cold start), this optimization problem basically splits into two disjoint problems:

- Achieve a fast catalyst light-off while managing feedgas (engine-out) emissions.
- Optimize fuel economy after catalyst light-off.

In this section, we shall only consider the latter problem.

Given that the engine speed and torque are constrained by the drive cycle, and the air–fuel ratio is kept close to stoichiometry to assure high efficiency of the catalyst system, the variables one can use to optimize fuel consumption are IVO, EVC, and the spark timing (spk). Basically, we are looking for the best combination of these three variables at various speed and torque points of engine operation (see Figure 2.3). As the fuel consumption is evaluated at a fixed torque, it is often referred to as the brake-specific fuel consumption (BSFC). Figure 2.4 shows BSFC versus two optimization variables at a typical speed/torque operating point (1500 rpm engine speed, 62 Nm torque). The top plot shows BSFC versus IVO and EVC at the spark timing for best fuel economy (called maximum brake torque (MBT) spark). The bottom plot shows BSFC versus IVO and spark, at EVC $= 30°$. Hence, the dash curves in Figure 2.4a and b show the same set of mapped points. The plots have been obtained by the full-factorial mapping, which is prohibitively time consuming to generate. On the other hand, achieving the fuel economy potential of the diVCT technology requires accurate knowledge of these (hyper)surfaces under all operating conditions.

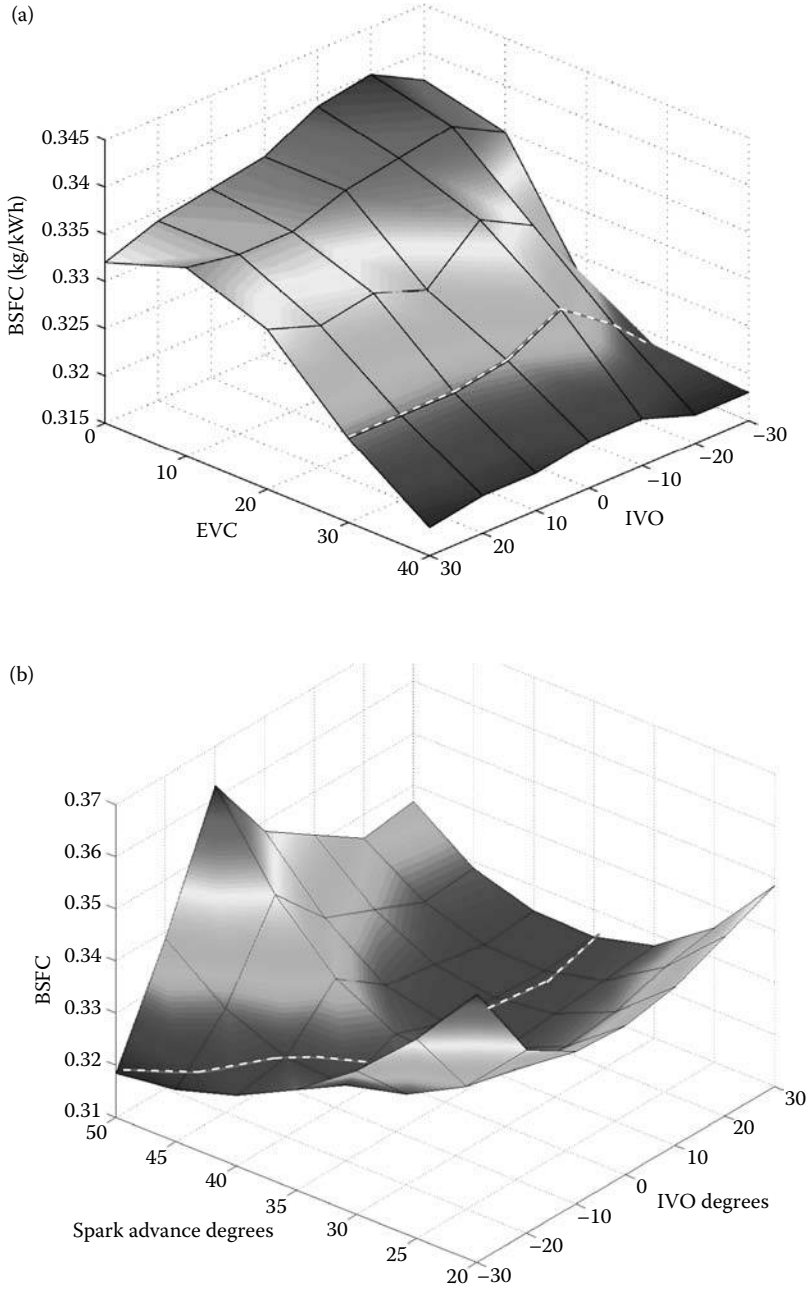

FIGURE 2.4 BSFC: (a) BSFC versus IVO and EVC at MBT spark; (b) BSFC versus IVO and spark at EVC = 30°.

2.3.1.2 Extremum Seeking (ES) Methods for Engine Optimization

To operate the engine with best fuel consumption under steady-state conditions only requires the knowledge of the optimal IVO–EVC pair, and the corresponding MBT spark timing. Several methods of obtaining these optimal combinations are available: ES with sinusoidal perturbation (Ariyur and Krstic, 2003), Direct Search methods such as Nealder–Mead (Wright, 1995; Kolda et al., 2003) and the Gradient Search methods (Box and Wilson, 1951; Spall, 1999; Teel, 2000). Such algorithms have already been used

for engine optimization. For example, Draper and Lee (1951) used the sinusoidal perturbation to optimize engine fuel consumption by varying air–fuel ratio and spark, while Dorey and Stuart (1994) used a gradient search to find the optimal spark setting.

ES is an iterative optimization process performed in real time on a physical system, that is, without a model being built and calibrated before. The function being optimized is the steady-state relationship between the system's input parameters and its performance output. The optimization function, in our case BSFC, denoted by $f(.)$ is usually called the response map. Since $f(.)$ is not known (otherwise, it could be optimized on a computer), ES controllers rely only on measurements to search for the optimum. Starting from some initial parameter values, the ES controller iteratively perturbs the parameters, monitors the response, and adjusts the parameters toward improved performance. This process runs according to some chosen optimization algorithm, usually as long as there is improvement.

Several ES algorithms that belong to the Gradient Search category have been developed and experimentally tested on a diVCT engine (Popovic et al., 2006). They include modified Box–Wilson, simultaneously perturbed stochastic approximation (SPSA) (Spall, 1999), and persistently exciting finite differences (PEFD) (Teel, 2000). The last two are closely related in implementation though not in theoretical underpinnings. Therefore, we shall describe how they work on our problem of optimizing the BSFC response map $f(x)$ with the optimization parameters vector $x = [x_1\ x_2\ x_3]^T$ consisting of IVO, EVC, and spark timing.

The SPSA and PEFD algorithms in 3D start with selection of one of the four perturbation directions:

$$v_1 = [1, 1, 1],\ v_2 = [-1, 1, 1],\ v_3 = [1, -1, 1],\ v_4 = [1, 1, -1].$$

SPSA selects the direction randomly and PEFD periodically in the round-robin fashion. The current values of the parameters $x(k)$ are perturbed in the direction of the selected vector v_i, by the amount controlled by the parameter λ, and the measurement of $f(x_k + \lambda v_i)$ is obtained. Because the map is generated by a dynamical system, the response of f to a change in input parameters can be measured only after a period of time, usually after large transients have died down. The measurements are usually noisy and may require filtering or averaging. For the BSFC optimization, we have waited for 1 s after the change in the set-point, and then averaged the measurement over the next 3 s.

Next, the current parameter value $x(k)$ is perturbed in the direction of $-v_i$ and the measurement of $f(x_k - \lambda v_i)$ is taken. The wait time and averaging are used again, and the total time of a parameter update step, dominated by the time needed to take the measurements, is found to be 8 s. With both the measurements in, the parameters are updated in the direction opposite to the directional derivative:

$$x_{k+1} = x_k - \alpha_k v_i \frac{f(x_k + \lambda v_i) - f(x_k - \lambda v_i)}{2\lambda},$$

where α_k is the time-varying step size for parameter update. After the new values of parameters are obtained, we repeat the procedure starting with the selection of the directional vector.

The experimental validation shown in Figure 2.5 is with the SPSA algorithm in a test cell where the dynamometer keeps the engine speed constant at 1500 rpm and the throttle is actuated to control the engine torque at 62 Nm. The top plot in Figure 2.5 shows the BSFC and the optimal parameter estimate versus time. The starting parameter estimate is $IVO = 0$, $EVC = 10$, $spk = 30$.

The noise-like jumps in BSFC are generated by the perturbations introduced by this method while the parameter plot only shows the estimates (without perturbation) so that they are smooth. After about 20 min, parameters have converged to the (local) minimum in BSFC. For comparison, generating the complete response surface, shown in Figure 2.5, takes about 15–20 h using the conventional engine mapping method. The initial and final points are indicated by arrows on the BSFC response surface (in IVO and EVC) as shown in the lower plot of Figure 2.5. In case it is not obvious, the minimum found by the algorithm is local. The (slightly lower) global minimum is on the other side of the ridge. Thus, in general, to find the global minimum, the algorithms will have to be run several times from different initial conditions.

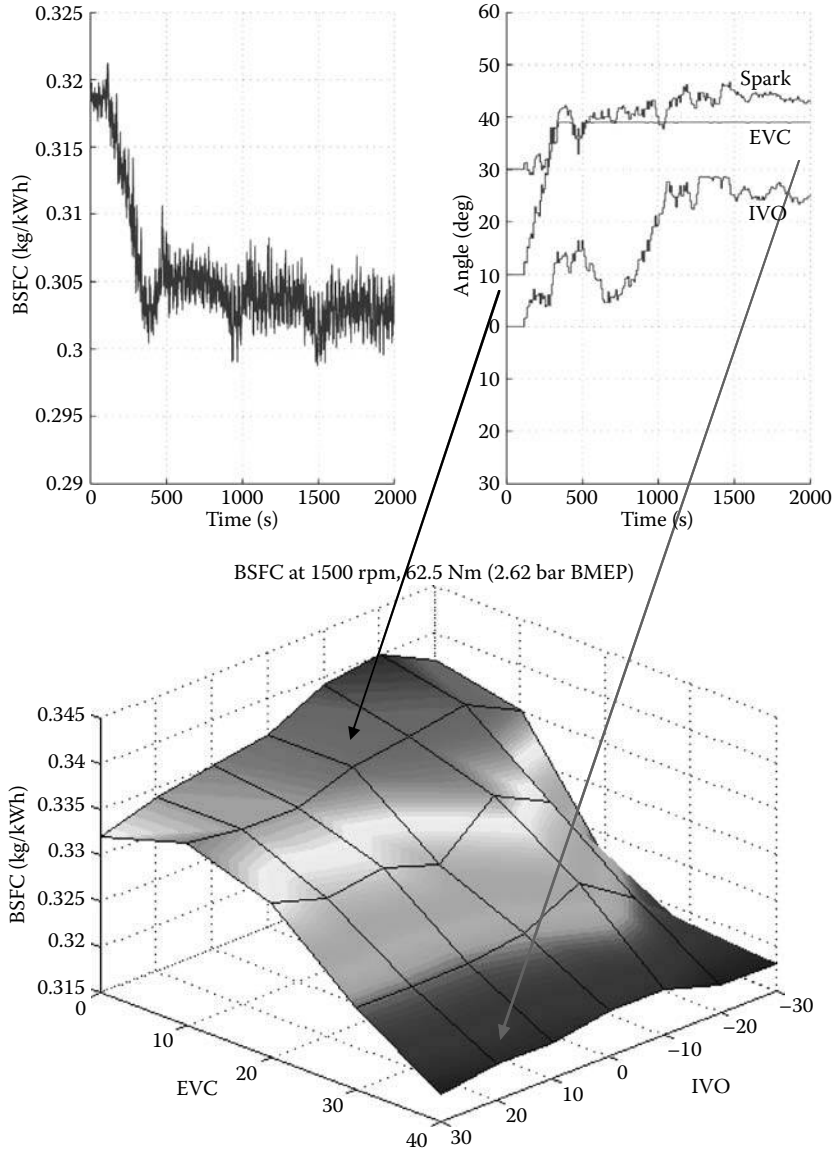

FIGURE 2.5 Parameter optimization run at 1500 rpm and 62 Nm: top plot—BSFC (left) and parameter values (right); bottom plot—the start and end points on the response surface.

Repeating the optimization at different speeds and torques provides the BSFC cam and spark tables for the entire engine operating range. More details about the performance of the optimization algorithms at different operating conditions (speed–torque pairs), comparison of performances, and discussion of some open issues can be found in Popovic et al. (2006).

2.3.1.3 Optimal Transient Scheduling

The previous subsection presented an efficient way to find the best BSFC parameter combinations at different speed and torque operating points. The outcome of each optimization run is a triplet of optimal parameter values, but no useful information about the complete response surface $f(x)$ is obtained.

If the engine is just run in steady state, the triplets of optimal parameters would be all that are needed. This is also acceptable if the engine speed and torque change relatively slowly, so that the optimization variables can always be at their optimal value. In such a case, the control strategy would schedule the desired (reference) values of the three parameters based on engine speed, N, and torque, Tq:

$$\begin{aligned} IVO\_ref &= Fn\_ivo(N, Tq), \\ EVC\_ref &= Fn\_evc(N, Tq), \\ spk\_ref &= Fn\_spk(N, Tq). \end{aligned} \tag{2.1}$$

Recall, however, that the engine speed and torque are constrained by the drive cycle, and, as one can see in Figure 2.3, are changing rapidly. Actual IVO and EVC, however, often trail their reference values by up to 0.5 s, while spark is practically instantaneous. Not synchronizing spark and VCT during transients could result in a large fuel economy penalty (up to 15%, based on Figure 2.4b) or even poor combustion quality and misfires in some cases. Hence, employing the scheduling approach (Equation 2.1), referred to as the lockstep scheduling, is in general not acceptable for transients. Instead, with spark being the fastest variable, it is usually scheduled on instantaneous (measured) values of the other four variables:

$$spk\_ref = Fn\_spk(N, Tq, IVO, EVC). \tag{2.2}$$

Note that the information needed to schedule spark according to Equation 2.2 is not available from the ES optimization. That is, ES test only provides the optimal triplet, but not the complete map needed to generate Equation 2.2.

2.3.1.4 Engine Mapping: Complexity/Fuel-Efficiency Tradeoff

As mentioned above, the full-factorial approach to mapping is too time consuming and the ES methods do not provide enough information to operate the engine under transient conditions. Hence, to generate the required parameter schedules, the automotive industry standard is to employ the design of experiments (DOE) methods (Montgomery, 2001; Edwards et al., 1999). The MATLAB® Model Based Calibration toolbox, based on DOE designs, is specifically directed toward automotive engine mapping and optimization tasks.

The main idea of a DOE method is to provide the designer with the best places to take measurements in the space of engine variables in order to support generating accurate response surfaces. For DOE mapping, the constrained variables (speed and torque), and the two optimization parameters (IVO and EVC) are lumped together while the spark timing is treated separately. That is, MBT spark timing is found by completing the so called "spark sweep" at each combination of the other four, which the DOE method selects to map. Then the MBT spark and the corresponding BSFC are fitted with a polynomial or radial basis function (RBF) regression. Once the regressions are available, finding the best BSFC points becomes an easy task (in particular if polynomials are used). The DOE designs take into account the regression method used (e.g., the degree of the polynomial) and the number of measurements to be taken. For example, the designer may specify that MBT spark and BSFC be regressed with a pth-order polynomial in N, Tq, IVO, and EVC:

$$\begin{aligned} spk &= \sum_{i+j+k+l+m=p} a_n \cdot N^i \cdot Tq^j \cdot IVO^k \cdot EVC^l \cdot 1^m, \\ BSFC &= \sum_{i+j+k+l+m=p} b_n \cdot N^i \cdot Tq^j \cdot IVO^k \cdot EVC^l \cdot 1^m. \end{aligned}$$

The number of points M to do the spark sweep is also picked as a tradeoff between accuracy and mapping time. The DOE tool then selects the locations for the M points (now in the 4-dimensional space) at which the measurements can be taken. As an illustration, we fix the engine speed, choose $p = 3$ and

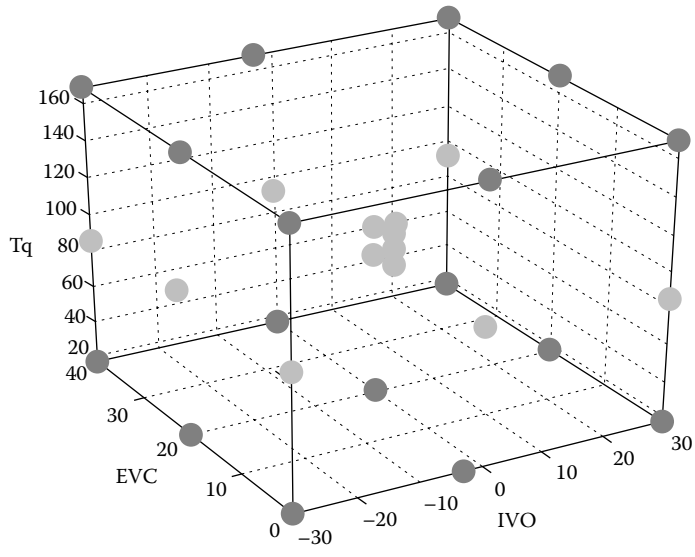

FIGURE 2.6 The pattern of mapping points for the V-optimal DOE in dimension 3 (Tq, IVO, and EVC); the shading corresponds to the vertical position of the point.

$M = 30$ and let the Model-Based Calibration V-optimal design provide the points at which the spark sweeps are conducted (Figure 2.6).

The efficiency of this approach and the tradeoffs between the accuracy and the number of mapped points have been evaluated on a model of the diVCT engine and are reported in Jankovic and Magner (2006). The evaluation was done by comparison to the full-factorial model serving as the benchmark. The full-factorial maps were only available over a given region (rectangle) in the operating space in which the engine spends a good amount of time when driven over a drive cycle such as US75 (Figure 2.3). The total fuel economy improvement potential (with full-factorial mapping) over the rectangle resulted in 3.11%, when compared to a fixed VCT engine (with fixed $IVO = -10$, $EVC = 10$ selected as typical values for a non-VCT engine). The full-factorial approach uses 630 spark sweeps over the rectangle to characterize the response surfaces (for comparison, the response surface in Figure 2.4 for one speed/torque point uses 35 sweeps). The V-optimal design with $M = 100$ sweeps in dimension 4 (N, Tq, IVO, and EVC) showed the BSFC improvement of 2.1%. This means that about a third of the potential benefit was not delivered due to the mapping constraint. Using other DOE designs (e.g., D-optimal design), regressions (RBFs), or more points did not produce an appreciable improvement in the complexity versus fuel-economy tradeoff. For example, using $M = 150$ improved the BSFC to 2.41% while increasing M over 200 sweeps showed diminished returns in terms of BSFC improvement. One explanation is that accuracy of DOE designs is known to be better at the center of the parameter space than around the edges, yet many (most) BSFC optimal points are on the edges of the IVO or EVC rectangle (see, e.g., Figure 2.4a).

2.3.1.5 Guided Mapping

The DOE designs treat the engine largely as a black box. Useful information about engine operation and efficiency is, however, available from the engine design phase (when the compression ratio, cam profiles, etc. parameters are selected). Alternatively, or in addition, the ES optimization can be used to quickly locate optimal (IVO, EVC, and spk) triplets at frequently used speed/torque operating points. Thus, if one is given *a priori* information that most of the optimal points are on an edge of the operating space, is there a way to modify the mapping process to get a better result than the black-box DOE? The answer to this question is presented in the following paragraphs.

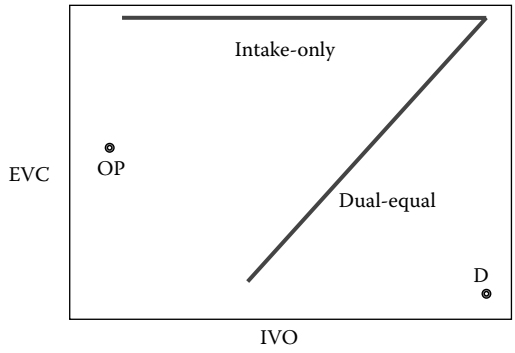

FIGURE 2.7 The pattern of features used for guided mapping of the diVCT engine.

The main idea behind the line mapping proposed in Jankovic and Magner (2004) is to use l lines to connect most optimal points and constrain the mapping to the lines. Scheduling of the optimal IVO and EVC in Equation 2.1, in general, would be constrained to the mapped lines. Hence, from the optimization perspective, the problem becomes that of mapping a lower dimensional space l times. If the lines are correctly selected, an improvement in the tradeoff between complexity and fuel economy can be achieved. For the problem at hand, it was found that more than 90% of the optimal points fall on two lines in the IVO–EVC plane: the dual-equal line and the intake-only line (Figure 2.7). Hence, the dual-independent VCT engine mapping task becomes that of characterizing (mapping) an engine with the dual-equal VCT and an engine with the intake-only VCT (both hardware choices correspond to engines actually used in production, although the location of the lines would be different). The two feature points shown in Figure 2.7, optimal performance (OP) and default (D), are there to support engine operation under special conditions (they are not needed for best BSFC).

The pattern of mapped points for the 2-line guided mapping, shown in Figure 2.8 in dimension 3 (again, with fixed engine speed), is very different from the V-optimal DOE in Figure 2.6. In the full 4-dimensions, with the full-factorial map on each point or line features, we obtain $M = 198$ sweeps (over the same rectangle considered above). The resulting BSFC benefit produced by comparison to the benchmark is 3.07% (close to the maximum 3.11%). If we employ the DOE on the features (basically mapping two lower-dimensional engines), with $M = 84$ we obtain BSFC improvement of 2.58%, still more favorable than the black-box DOE with almost double the number of points.

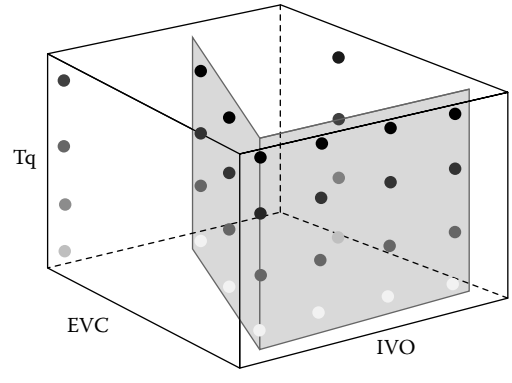

FIGURE 2.8 The pattern of mapping points for the 2-line guided mapping in dimension 3 (Tq, IVO, and EVC).

2.3.1.6 Inverse Distance Interpolation

The guided mapping addresses a deficiency of the black-box DOE mapping, but brings back the question of transient scheduling. That is, when the IVO and EVC traverse the space in between the mapped features during a transient, how is the engine controller going to compute the corresponding spark timing? One answer is to use kernel interpolation extended to use line features (Jankovic and Magner, 2004).

The interpolation, in general, deals with the problem of finding values of a function $Y(x)$, where $x \in R^m$, given the measured data pairs (X_i, Y_i), $i = 1, \ldots, M$, that typically come from setting the independent vector variable X at some predetermined or given set of values X_i, and taking the measurements of the dependent variable Y_i. The data pairs (X_i, Y_i) may come from taking engine measurements as specified by one of the mapping methods discussed above.

The full-factorial mapping provides a pair (X_i, Y_i) at each node of a hypercube providing the (hyper) look-up table structure. If this structure is stored in the controller memory, the spark timing at any point in the 4-dimensional (N, Tq, IVO, and EVC) space is done by the standard multilinear interpolation.

The parametric methods (such as polynomial regression) are usually computationally very efficient [few parameters represent a (hyper)surface]. Another advantage is that the DOE mapping produces a parametric regression anyway. On the other hand, parametric methods require a preselected model (e.g., a third-order polynomial) which may behave erratically in data-poor regions (this is why the DOE tools produce a fairly balanced grid of points to map), and are difficult to adjust for manual in-vehicle calibration.

In the IVO–EVC space, the guided mapping produces very irregularly spaced data points (Figure 2.8). In the N–Tq dimensions, the data points are regularly spaced, so the surfaces can be represented with look-up tables or parametric (polynomial) regressions. In the IVO–EVC dimensions, we consider nonparametric (kernel) methods as an alternative. They are computationally less efficient than parametric methods, but are model independent, more predictable in data-poor regions, and easier for direct calibration.

The kernel interpolation starts with a given symmetric function $K(u)$, called the kernel, and computes the interpolated value of Y at some value of x by

$$\hat{Y}(x) = \frac{\sum_{i=1}^{M} K(x - X_i) \times Y_i}{\sum_{i=1}^{M} K(x - X_i)}. \tag{2.3}$$

Various kernel functions have been considered in the literature including the parabolic "Epanechnikov kernel" and the "Gaussian kernel" that uses the Gaussian probability distribution function as the kernel (Hardle, 1989). Which is the best kernel function to use depends on the application. In our case, the sparse data structure and the desire that the interpolation surfaces on the feature (line or point) pass through the value at the feature suggest that the "inverse-distance kernel" be used:

$$K(u) = \frac{1}{u^2 + \varepsilon}, \tag{2.4}$$

where ε is a small constant employed to prevent division by 0. For our application, the interpolation formula (Equation 2.3) has been extended to work with points (*Xi*, *Yi*) and lines (L_i, Y_{Li}).

For the point features (such as OP and D in Figure 2.7), computing the kernel K_p is straightforward. By denoting the measured (IVO, EVC) pair by $x = (x_1, x_2)$, the kernel value at the *i*th data point is

$$K_p(x - X_i) = \frac{1}{(x_1 - X_{i1})^2 + (x_2 - X_{i2})^2 + \varepsilon}.$$

The corresponding Y_i is just the value of spark timing at X_i and the current N and Tq: $Y_i = spk_i = \text{Fn\_spk}_i(N, Tq)$ where, depending on the representation, $\text{Fn\_spk}_i(.)$ can be a polynomial, a look-up table, and so on.

To compute the kernel values at the feature line, $K_L(x, L_i)$, we parameterize each line segment by a parameter s, which takes values between s_{min} and s_{max}. Thus, x_1 and x_2 coordinates on the line segment must satisfy

$$x_1 = a_{i1}s + b_{i1},$$
$$x_2 = a_{i2}s + b_{i2}.$$

Instead of the distance to a feature point, the kernel now needs the distance to the feature line, that is, to the closest point on the line segment. The value of the parameter s at this point can be found by minimizing the squared distance

$$d_{Li}(x_i, x_2, s) = (x_1 - a_{i1}s - b_{i1})^2 + (x_2 - a_{i2}s - b_{i2})^2$$

with respect to s. The minimum must satisfy

$$\frac{\partial d_{Li}}{\partial s} = 2a_{i1}(x_1 - a_{i1}s - b_{i1}) + 2a_{i2}(x_2 - a_{i2}s - b_{i2}) = 0.$$

Solving the above equation for s, and including the limits on s, we obtain s_i^*, the value that corresponds to the closest point on the ith line segment to (x_1, x_2):

$$s_i^* = min\left\{s_{imax}, max\left\{s_{imin}, \frac{a_{i1}x_1 - b_{i1} + a_{i2}x_2 - b_{i2}}{a_{i1}^2 + a_{i2}^2}\right\}\right\}.$$

Now the value of the kernel evaluated at this line is

$$K_L(x, L_i) = \frac{1}{(x_1 - a_{i1}s_i^* - b_{i1})^2 + (x_2 - a_{i2}s_i^* - b_{i2})^2 + \varepsilon}.$$

The value of Y_{Li} on the ith line is obtained by standard methods: $Y_{Li} = spk_{Li} = \text{Fn\_spk}_{\text{Li}}(N, Tq, s_i^*)$. With the values of K_i, Y_i, K_{Li}, and Y_{Li} available, obtaining the spark at any point (N, Tq, IVO, and EVC), can be done by substituting the values into Equation 2.3.

This interpolation algorithm has been implemented in an experimental vehicle. The spark data computed by the in-vehicle inverse-distance interpolation method has been collected and compared to the spark timing generated by the full-factorial model run with the same inputs (also collected online). Figure 2.9 shows the spark timing comparison of the in-vehicle inverse-distance and full-factorial model over a 40 s run. The error is negligible—the total fuel economy loss is estimated at 0.04% due to spark not being equal to the benchmark (full factorial). As the cam timing schedule is also generated by the almost lossless 2-line guided mapping procedure (the one with $M = 198$), we expect that the combined mapping and interpolation provide an almost loss free schedule. This has been confirmed by running back-to-back vehicle tests comparing the fixed VCT schedule with the 2-line schedules/interpolation on a chassis dynamometer. The observed benefit of 2.97% (average of $4 + 4$ runs) is not far from the 3.11% model prediction. The fuel consumption in the tests was measured by emission analyzers using the standard method generally considered accurate, though vehicle testing may introduce additional factors that could confound the comparison.

2.3.2 Idle Speed Control

Idle Speed Control (ISC) is one of the key feedback control functionalities of modern gasoline and diesel engines (Hrovat and Sun, 1997; Hrovat and Powers, 1998). In spark-ignition (SI) gasoline engines, ISC manipulates the electronic throttle position and the spark timing (inputs) to maintain engine speed (controlled output) at a set-point when the driver's foot is off the accelerator pedal. See Figure 2.10. Measured (estimated) and unmeasured torque disturbances due to power steering, air-conditioning

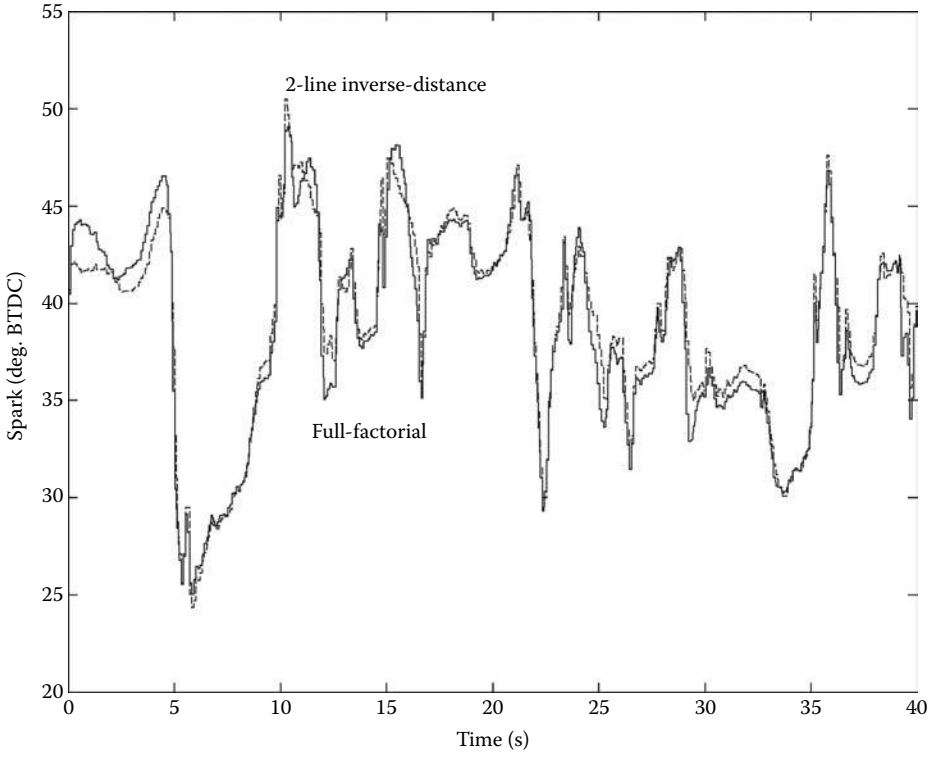

FIGURE 2.9 Comparison of the 2-line inverse-distance spark computed in the vehicle and the full-factorial model.

turning on and off, transmission engagement, and alternator loading changes must be rejected. The set-point is typically around 625 rpm in a drive gear and around 650 rpm in neutral, and it may be modified depending on the state of accessories, during transition into idle, and at cold engine and ambient temperatures. The engine speed can be treated as a measured output available for feedback control. Strict performance requirements are imposed to, whenever possible, contain the engine speed dip (maximum engine speed decrease when disturbances hit) within 80 rpm and engine speed flare (maximum engine speed increase when disturbances hit) within 120 rpm.

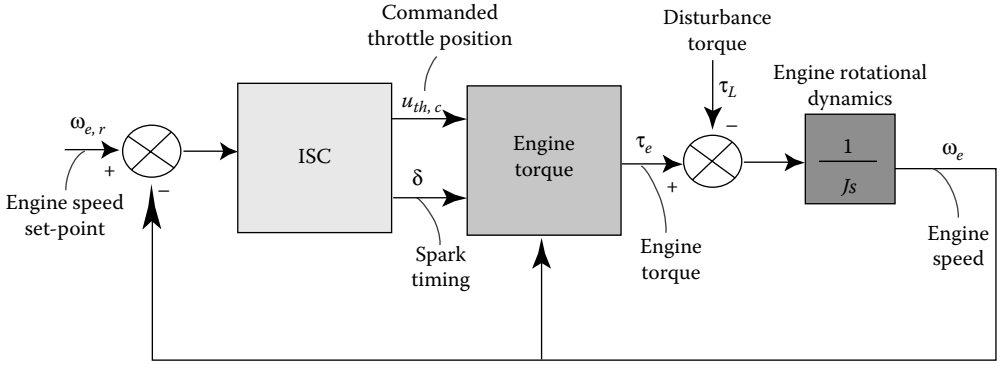

FIGURE 2.10 Gasoline engine ISC block diagram.

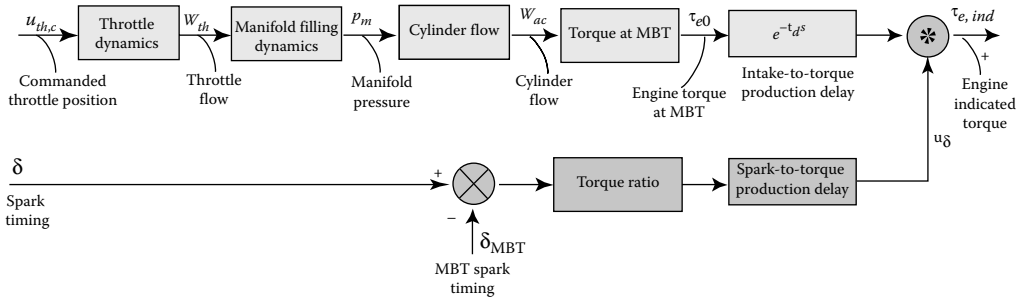

FIGURE 2.11 Gasoline engine indicated (combustion) torque.

Both the electronic throttle and the spark timing influence the engine torque and can compensate for the effect of disturbances. See Figure 2.11 for a block diagram of engine indicated torque production. The engine torque at the flywheel, $\tau_e = \tau_{e,ind} + \tau_{e,pump} + \tau_{e\,fric}$, is a sum of the indicated torque, pumping loss, τ_{pump}, and mechanical friction torque, $\tau_{e,fric}$.

Increasing engine throttle opening causes air flow through the throttle to increase. This, in turn, increases the intake manifold pressure and the air flow into the engine cylinders also increases. In response to the cylinder air flow increase, the air-to-fuel ratio controller increases the fuel flow, thereby maintaining the air-to-fuel ratio at a set-point. The engine torque increases as a result of the increase in the air and fuel cylinder charge, subject to the intake-to-power (IP) delay, that is, the delay between the intake stroke of the engine and torque production, which is about 360° of crankshaft revolution.

Adjusting the spark timing affects the start of combustion relative to the revolution of the crankshaft. The spark-to-torque production delay is small and adjusting the spark timing has an almost immediate effect on engine torque. Thus the spark timing may be viewed as a fast actuator as compared to the throttle angle modulation, which is a slow actuator.

The spark timing in deg of crankshaft revolution measured Before Top Dead Center or BTDC at which the maximum torque is produced is referred to as MBT spark timing. The spark timing is normally only retarded (made to occur later) relative to the MBT spark timing, which causes engine torque to decrease (see Figure 2.12). The maximum spark retard is limited by combustion stability constraints. To ensure that spark has a bidirectional authority over the engine torque, the set-point for spark timing in steady state is often retarded from MBT. This steady-state spark retard is referred to as a *spark reserve*. The need to operate with the spark reserve degrades fuel economy compared to the case when spark timing is maintained at MBT. There is about 90 deg of crankshaft revolution spark-to-power (SP) delay, which is often (but not always) neglected in comparison with the IP delay.

Historically, ISC is related to one of the oldest mechanical feedback control systems, the so-called Watt's governor (1787) which was used as a speed controller for a steam engine. Speed regulation using Programmable Logic Controllers has been used for years in stationary engines utilized as components in power generation and transport systems such as natural gas pipelines. The ISC in automotive systems has been a popular case study in the control literature; see, for example, a survey article by Hrovat and Sun (1997). In older gasoline engines with a mechanical throttle (i.e., throttle mechanically connected to driver gas pedal), a dedicated actuator called the air-bypass valve was used for ISC. In modern engines, with electronic throttle, the air-bypass valve has been eliminated. This did not simplify the control problem as the electronic throttle designed for regular driving operates only in a narrow range during idle where small imperfections may have large impacts.

Despite the advances made in ISC over the years, further improvements are of significant interest as they can translate into vehicle attribute improvements, in particular, better fuel economy. The need to maintain spark reserve, in particular, results in fuel consumption penalty as compared to the case when spark timing is maintained at MBT. The IP delay is one of the key factors, which causes conservative tuning of the throttle to engine speed control loop and thus prevents the elimination of the spark reserve.

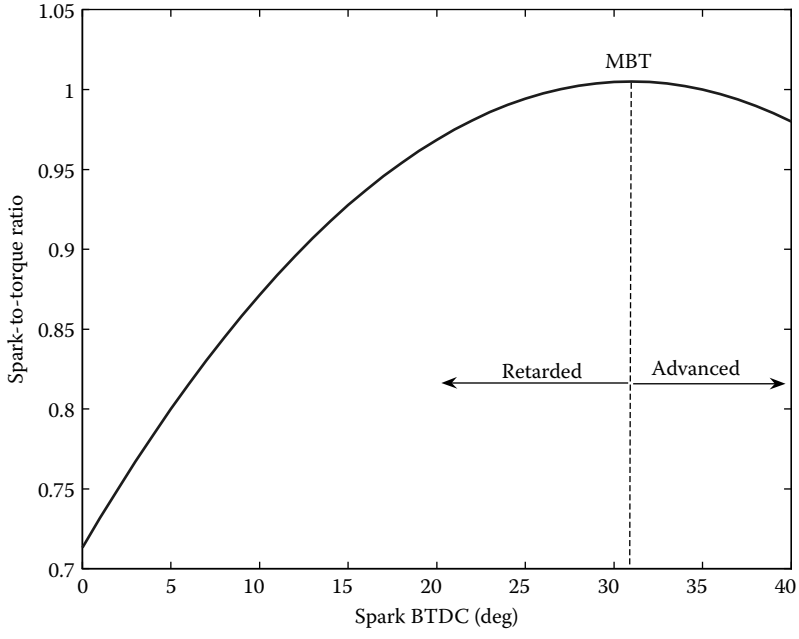

FIGURE 2.12 Fraction reduction in engine indicated torque when retarding spark timing from MBT.

Another opportunity to reduce fuel consumption at idle is to lower the idle speed set-point, provided this can be accommodated by vehicle accessories. Lowering the idle speed set-point complicates the problem in two ways. First, the IP delay, t_d, is inversely proportional to engine speed where ω_e is engine speed in rad/s, and becomes larger as the engine speed decreases. Second, the engine speed excursions, especially dips, must be tighter controlled as engine speed drops below the so-called "fishhook" point can result in a rapid increase in engine friction, which, combined with the increased delay, can lead to instability and engine stall. Note that the closed-loop system must operate flawlessly despite variability caused by aging and manufacturing tolerances, and due to operation across a wide range of environmental and operating conditions (temperature, pressure, humidity, dust levels, etc.). As a particular example, deposits on the throttle body (the so-called "throttle sludging") change the effective throttle flow area, and, if not properly handled in the development of the controller, can result in engine stalls during idling.

Concluding this overview, we note that feedback control of engine speed may be employed in other operating conditions, for example, during transmission shifts in vehicles with automatic transmission. In diesel engines (not considered in detail in this section), the ISC is achieved by adjusting the fuel flow to regulate engine speed. The transport delay is much smaller in that compression ignition case, as the direct injection event causes the start of combustion and the fuel quantity can be modified just before injection.

2.3.2.1 Open-Loop Model and Properties

The plant model reflects the rotational dynamics of the engine and the intake manifold filling and emptying dynamics:

$$\dot{\omega}_e = \frac{1}{J}\left(\tau_{e0}(t - t_d)u_\delta + \tau_{e,pump} + \tau_{e\,fric} - \tau_L\right),$$

$$\dot{\tau}_{e0} = -k_2\frac{RT}{V}\left(\frac{\omega_e}{k_1}\tau_{e0} - k_3 u_{th}\right),$$

where, referring to Figures 2.10 and 2.11, J is the effective crankshaft inertia, τ_{e0} is the engine torque at MBT, u_δ is the spark influence on indicated torque, τ_L is the disturbance (load) torque, u_{th} is the

throttle position, R is the gas constant, T is the intake manifold temperature, V is the volume of the intake manifold, and k_0, k_1, k_2, k_3 are model parameters such that the cylinder air flow is given by $W_{ac} = \frac{k_2}{k_1} p_1 \omega_e + k_0$, the engine torque at MBT is given by $\tau_{e0} = \frac{k_1}{\omega_e} W_{ac}$, the throttle flow, assuming choked conditions, is given by $W_{th} = k_3 u_{th}$. The step response of the engine speed to a change in throttle position exhibits a second-order lightly damped character (damping ratio of about 0.5) and a time delay, where the amount of damping decreases for unloaded conditions (e.g., transmission in "neutral").

The conventional engine ISC is based on feedforward, Proportional-plus-Integral-plus-Derivative (PID) feedback for air flow and Proportional-plus-Derivative (PD) feedback for spark timing. The feedforward term accounts for estimated accessory load disturbances. The feedback gains may depend nonlinearly on the engine speed error to provide faster response when the error is large. The measured engine speed error is filtered with a rolling average (low-pass) filter before being fed into the proportional and derivative terms. The feedback gains may be made functions of operating conditions (engine coolant temperature, transmission in neutral or drive, air-conditioning on or off), and the derivative term may be suspended during extended flare conditions. A simple integrator antiwindup strategy is employed, where the integrator updates are suspended if the integrator state exceeds predefined limits. A simple adaptive function is implemented to learn the value of the integrator into Keep-Alive-Memory and subsequently apply it as a feedforward. The tuning (calibration) of the controller must account for the component variability and apparently the recommended practice is to tune the ISC gains for the "slow end" of component variability range while maintaining adequate stability margins. From the standpoint of control design, the PID controller approach is simple and understandable without significant training.

In the remainder of this section we discuss three control system design approaches based on advanced control methods:

- Nonlinear control which offers insights into system architecture.
- Adaptive control approach which assumes very little prior system knowledge and handles time delay through prediction.
- Model predictive control (MPC) approach, which utilizes linear model-based prediction and constrained control optimization, and results in a piecewise affine control law.

The salient features of these applications will be discussed and illustrated through vehicle experimental results.

2.3.2.2 Nonlinear Control

While effective ISC can be based on linear design techniques, nonlinear control approaches can offer insight into the architecture of the control system:

- The spark timing can be controlled to compensate for the time delay, that is, $\tau_{e0}(t - t_d)u_\delta(t) = \tau_{e0}(t)$, where $\tau_{e0}(t)$ is a prescribed control input. With this choice of fast spark timing control, and provided it is within the spark authority, the delay may not have to be considered in the design of the throttle control loop.
- The control law of Proportional-plus-Integral type for $\tau_{e0}(t)$ is designed to stabilize the engine speed, $\omega_e(t)$, to its set-point, r. This design can be based on the "$L_g V$" (Sepulchre et al., 1997) or Speed-Gradient (Fradkov, 1979) design techniques. An unknown input observer (Kolmanovsky and Yanakiev, 2008; Stotsky et al., 2000) or an adaptive Kalman filter (Pavkovic et al., 2009) may be used to estimate τ_L and use this estimate for the feedforward compensation.
- The backstepping approach (Sepulchre et al., 1997) can be applied to derive a control law for u_{th} (Kolmanovsky and Yanakiev, 2008; Stotsky et al., 2000). To avoid numerical differentiation of the control law for $\tau_{e0}(t)$ designed in the first stage, the backstepping approach based on the dynamic surface control can be applied (Kolmanovsky and Yanakiev, 2008).

Note that the overall control system resulting from this analysis has a cascade structure, with the inner loop controlling the throttle to bring estimated engine indicated torque to the set-point determined by

the outer loop from the engine speed tracking requirements. See Kolmanovsky and Yanakiev (2008) for an experimental evaluation of this controller.

2.3.2.3 Adaptive Posicast Control

Adaptive Posicast Controller (APC) is a nonlinear controller for time-delay systems designed from the adaptive control viewpoint. It is related to the ideas of the classical Smith predictor and finite spectrum assignment (Niculescu and Annaswamy, 2003). The open-loop system is viewed as linear and time-invariant, with a delay at the input. The parameters of the open-loop transfer function may be unknown except for the known values of the delay and the relative degree (excess of poles over zeros). The delay-free part of the plant transfer function can be unstable but is assumed not to have any transmission zeros in the closed right half plane even though under certain assumptions, known transmission zeros in the closed right half plane can be treated with extra design steps.

In the application of APC to ISC considered in Yildiz et al. (2007), the control input, $u(t)$, is the deviation of the throttle air flow command from nominal, and it is given by the following expression:

$$u(t) = \theta_1^T(t)\omega_1(t) + \theta_2^T(t)\omega_2(t) + \theta_4^T(t)r + \int_{-t_d}^{0} \lambda(t,\sigma)u(t+\sigma)\,d\sigma,$$

$$\dot{\omega}_1 = \Lambda_0\omega_1 + l \cdot u(t-t_d),$$

$$\dot{\omega}_2 = \Lambda_0\omega_2 + l \cdot y(t),$$

$$e(t) = y(t) - y_m(t),$$

$$\theta = \begin{bmatrix}\theta_1 & \theta_2 & \theta_4\end{bmatrix}^T,$$

$$\Omega = \begin{bmatrix}\omega_1 & \omega_2 & r\end{bmatrix}^T,$$

$$\dot{\theta} = -\Gamma\, e(t)\Omega(t-t_d),$$

$$\frac{\partial\lambda(t,\tau)}{\partial t} = -\Gamma_\lambda e(t)u(t-t_d+\tau), \quad -t_d \le \tau \le 0.$$

Here, y denotes the deviation of the engine speed from the nominal value, r is the set-point for this deviation, t_d denotes the time delay, $y_m(t)$ is the output of the reference model with the input $r(t-t_d)$ and with the number of poles equal to the relative degree of the open-loop transfer function, (Λ_0, l) is a controllable pair of the same dimension as the number of the plant poles and with the dynamics faster than that of the reference model, $\theta(t)$ denotes the adapted parameters (integral term is approximated by a sum of the past control input weighted with weights $\lambda(t,\sigma_i)$ in the actual implementation), and Γ, Γ_λ are gain matrices. The controller was enhanced with the following additional features (Yildiz et al., 2007):

- The sigma-modification approach was used to ensure robustness and prevent parameter drift.
- Antiwindup logic was added to handle actuator saturation.
- Procedures were given to select initial estimates of adaptive parameters based on approximate knowledge of plant model parameters and to select the learning gains based on the desired adaptation speed.
- Robustness to uncertainties in the knowledge in the delay value has been experimentally demonstrated.

Figures 2.13 and 2.14 illustrate the results achieved with this control approach in the vehicle for power steering disturbance rejection. Note that in terms of the engine speed excursions and integral of the error, the APC compares favorably with the baseline PID controller, despite the fact that little explicit knowledge of the system model was required for the development of the controller.

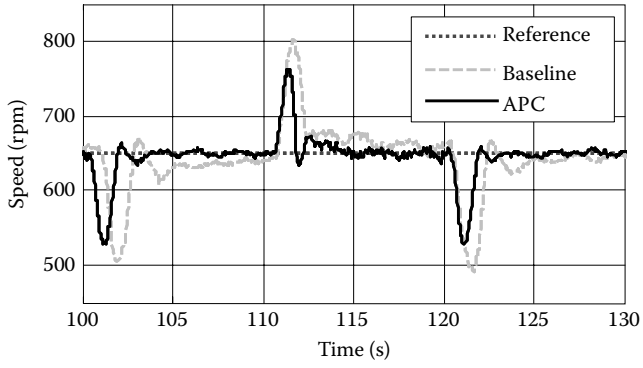

FIGURE 2.13 Rejection of power steering disturbance by the APC at 650 rpm.

2.3.2.4 Model Predictive Control

The MPC is of interest for powertrain applications because of its capability for near-optimal control of the system while enforcing state and control constraints, dealing with delays and handling hybrid system models with both continuously valued and discrete-valued (categorical) decision variables. In the event of system parameter changes or failures, the MPC law can be reconfigured by changing the model used in the prediction.

The MPC has emerged from and has been used for years in the chemical process industry where processes are slow, computing power is abundant and tuning can often be performed individually for each process being controlled. The trends toward increasing on-board computing power and evolving design methods enable the use of MPC in mass production automotive applications.

The application of MPC to ISC with load disturbance preview has been first proposed in (Hrovat, 1996) which demonstrated feasibility and benefits of the approach using simulations. Recent advances in MPC methods and tools have enabled an experimental implementation for this problem (Di Cairano et al., 2008) in a vehicle. Extensive testing and analysis of the MPC solution has been conducted, suggesting that MPC can be a realistic technology for future automotive applications. These developments are now briefly reviewed.

Recall that, for ISC in gasoline engines, the objective is to regulate the engine speed deviation, y, from the set-point, r, using either a scalar control input (throttle position) or a vector control input (throttle

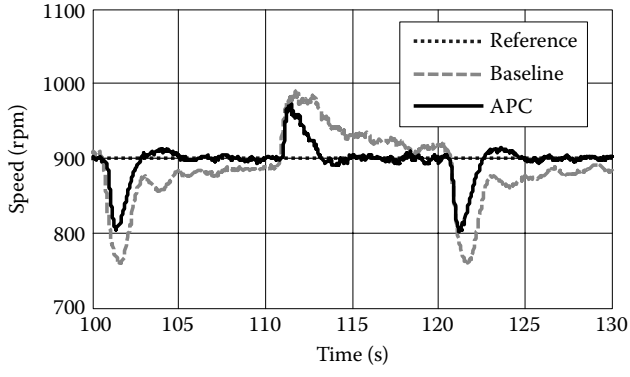

FIGURE 2.14 Rejection of power steering disturbance at 900 rpm by APC.

position and spark timing). The MPC formulation is based on the following optimization problem:

$$\omega\sigma^2 + \sum_{i=0}^{N-1} ||y(i|k) - r(k)||_2^Q + ||\Delta u(i|k)||_2^R \rightarrow \underset{\sigma, u(k)}{min},$$

subject to

$$\begin{aligned}
x(i+1|k) &= Ax(i|k) + Bu(i|k), \quad i = 0, 1, \ldots, N-1,\\
y(i|k) &= Cx(i|k), \quad i = 0, 1, \ldots, N-1,\\
u_{min} &\le u(i|k) \le u_{max}, \quad i = 0, 1, \ldots, N-1,\\
\Delta u_{min} &\le \Delta u(i|k) \le \Delta u_{max}, \quad i = 0, 1, \ldots, N-1,\\
y_{min} - \sigma &\le y(i|k) \le y_{max} + \sigma, \quad i = 0, 1, \ldots, N_c - 1,\\
u(i|k) &= u(N_u - 1|k), \quad i = N_u - 1, \ldots, N-1,\\
u(-1|k) &= u(k-1),\\
x(0|k) &= \hat{x}(k), \quad \sigma \ge 0,
\end{aligned}$$

where $y(i|k)$, $u(i|k)$, and $x(i|k)$ denote, respectively, the output, the control input, and the state predicted i steps ahead from the time instant k and $r(k)$ denotes the set-point for the measured output. A linear model with a realization (A, B, C, D) can be identified for the prediction from the experimental engine response data. Limits on the control input amplitude (u_{min}, u_{max}) and its rate of change $(\Delta u_{min}, \Delta u_{max})$ are imposed. A slack variable, σ, was introduced to relax the output constraints (with limits y_{min}, y_{max}) and is penalized in the cost function with a weight ω. The cost also penalizes the deviation of the output from the set-point and the change in the control input. Note that $\|y - r\|_2^Q = (y-r)^T Q(y-r)$, $\|\Delta u\|_2^R = (\Delta u)^T R(\Delta u)$. The optimization problem reduces to a Quadratic Program (QP) problem in the variables $u(i|k), i = 1, \ldots, N_u - 1$ and σ. This QP problem parametrically depends on the state, $x(0|k)$, and using the approach of Bemporad et al. (2002) can be solved offline, thereby providing an explicit MPC law in the form of a piecewise-affine function of the estimated state, $\hat{x}(k)$, defined over polyhedral region partitioning:

$$\begin{aligned}
&u(k) = F_i\hat{x}(k) + G_i u(k-1) + T_i r(k),\\
&i \in \{1, 2, \ldots, N_r\} : H_i\hat{x}(k) + J_i u(k-1) + K_i r(k) \le 0.
\end{aligned}$$

The matrices in this expression can be computed numerically using Hybrid Toolbox for MATLAB (Bemporad, 2003) or the Multiparametric Toolbox (Kvasnica et al., 2004). The key enabler for ISC application is the explicit MPC solution that avoids the need for implementing online optimization in software to compute the control law.

The MPC design in Di Cairano et al. (2008) was based on a discrete-time engine model identified from in-vehicle throttle and spark timing step response data. The IP delay was modeled as an input delay and handled by augmenting additional states to the model. An integrator on the engine speed was also augmented to the model to provide zero steady-state error. A Kalman filter was introduced for estimating the state and the load torque disturbance, and the tuning of this Kalman filter has been shown to have significant influence on the disturbance rejection performance of the controller. The prediction horizon was comparable to the settling time of the open-loop response of the system ($N = 30$ time steps of 30 ms), but shorter control horizon ($N_u = 3$) and constraint horizon ($N_c = 3$) were selected for the best tradeoff between the response and complexity of the solution.

Output constraints on engine speed deviation were handled as soft with introduction of a slack variable to avoid infeasibility (infeasibility refers to the situation in which no solution exists satisfying constraints) for large disturbances. The output constraint has provided extra flexibility in shaping the transient response and faster recovery when the actual and/or predicted speed tracking error was large.

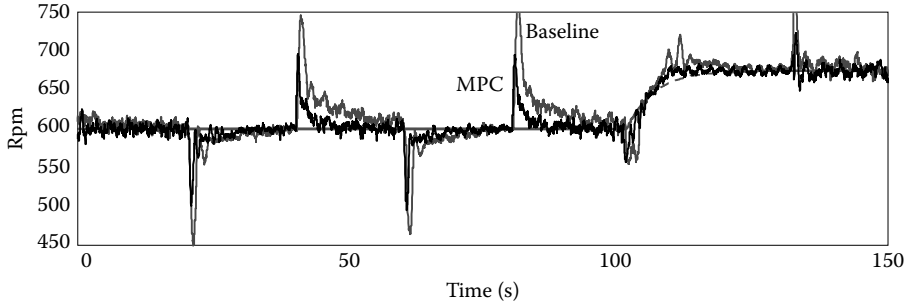

FIGURE 2.15 Power steering load rejection and set-point tracking with the MPC controller compared to the baseline controller.

Figure 2.15 illustrates the performance of the throttle-only MPC controller in the vehicle, with seven polyhedral regions, $N_r = 7$. The conventional spark controller was retained with this design. The worst-case Electronic Control Unit (ECU) computational loading was only 0.05% of ECU capability at idle, which is manageable for implementation.

The MPC design for coordinated control of both throttle and spark timing was obtained with $N_r = 131$ and was also tested in the vehicle. The resulting controller is able to maintain near-optimal actuator coordination in presence of saturation (in particular, on the spark timing range), with the worst-case ECU computational loading under 5%. The performance was improved compared to the simpler MPC solution but at the cost of increased worst-case computing effort.

2.3.3 Closed-Loop Air–Fuel Ratio Control

The after-treatment systems (three-way catalysts) in gasoline engines are exceptionally efficient in removing emissions of hydrocarbons (HC), carbon monoxide (CO), and nitrogen oxides (NO_x) from the tailpipe exhaust gas. The catalyst efficiency of well over 99% is in fact needed to meet today's stringent regulations for these three emission categories. The high catalyst efficiency is achieved as long as the engine operates at the stoichiometric air to fuel ratio—the value at which there is just enough air for complete oxidation of all the fuel (carbon and hydrogen). For the conventional gasoline fuel, the stoichiometric air–fuel ratio is around 14.6 or, equivalently, and the stoichiometric fuel-to-air ratio (FAR) is 0.069.

The catalytic convertors are also capable of storing and releasing oxygen. The oxygen storage capability allows temporary efficient operation off stoichiometry. As long as there is some oxygen available, the catalytic convertor will continue oxidizing HC and CO into H_2O and CO_2. Conversely, as long as there are free oxygen storage locations available, NO_x will get reduced and the released oxygen stored (with N_2 coming out of the tailpipe). Thanks to this mechanism, the tail pipe emissions correlate to the "area under the curve" of the FAR signal. The larger the area, the larger the potential for breakthrough of a pollutant. If FAR stays rich (lean) long enough, the emissions will increase significantly after the oxygen storage is empty (full).

2.3.3.1 FAR Regulation System

In the automotive gasoline engine applications, engine torque is controlled by regulating the amount of air that enters the cylinders, while the fuel–air ratio is controlled by adjusting the amount of fuel injected. To maintain the FAR as close to stoichiometry as possible, automotive engines employ an elaborate feedforward system coupled with a closed-loop system that consists of the inner closed-loop

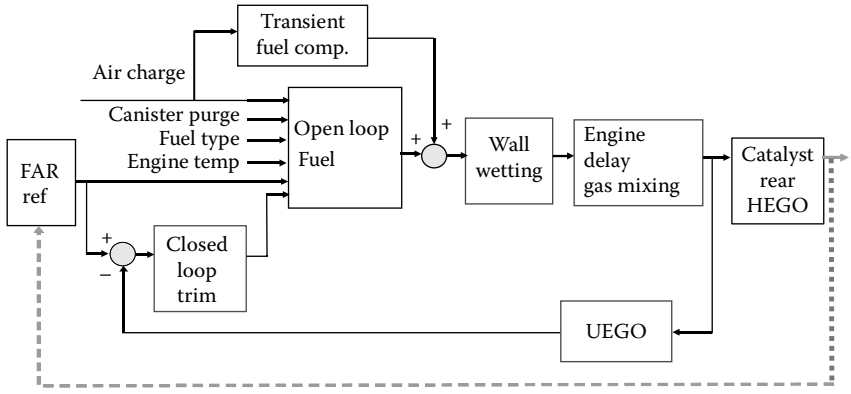

FIGURE 2.16 A block diagram of an automotive fuel control system.

FAR controller and the outer regulation loop. The fuel control system, shown in Figure 2.16, can be decomposed into several parts:

- Cylinder air charge entering the cylinder during the induction phase of the engine cycle is estimated. The fuel is determined to provide a desired fuel–air ratio (usually stoichiometric). This system must take into account fuel from other sources (canister purge), type of fuel (gasoline fuel with added ethanol), and so on.
- Transient fuel compensation that anticipates the wall fuel puddle effect on the mass of fuel that enters the cylinder (see Guzzella and Onder, 2004 Section 2.4.2).
- Inner loop controller "trims" the fuel to achieve the desired FAR at the sensor located upstream of the catalyst. The sensor typically measures excess oxygen in the combusted gas, but can be devised to provide the signal that measures fuel–air ratio. This chapter assumes the use of such a sensor known as the UEGO sensor.
- The reference fuel–air ratio is determined based on engine operating demands, cold start combustion stability, or catalyst efficiency. Under normal driving conditions, an outer loop controller feeds the inner UEGO feedback controller with the modulated FAR reference signal. Operation of the outer loop, shown by dotted lines, is beyond the scope of this chapter (Peyton-Jones et al., 2006).

A significant research interest and substantial literature have been devoted to the components of the above-described system. The interested reader is directed to the two books on engine controls (Guzzella and Onder, 2004; Kiencke and Nielsen, 2000), and the references therein. In the remainder of this chapter, we shall focus on the inner loop FAR regulation and, in particular, on potential for improvements of the closed-loop system performance with the Smith Predictor compensation for the time delay.

2.3.3.2 Linear System Model

If the trim is computed in FAR units (we used normalized FAR, $\phi = \text{FAR}/\text{FAR\_stoic}$), the inner-loop signal path from the output of the controller to the UEGO sensor (see Figure 2.16) includes multiplication by the air-charge estimate to get to the injected fuel, wall-wetting dynamics, cylinder air and fuel mixing, combustion, exhaust, and finally travel to the UEGO sensor. The physics of this process is quite complex. The wall-wetting effect tends to be minimal once the cylinder port reaches nominal operating temperature. The rest of the system, which will represent our open-loop plant P, is usually modeled as a lumped parameter system that just includes a time delay and a first-order lag (Guzzella and Onder, 2004):

$$P(s) = \frac{e^{-\tau_d s}}{T_c s + 1}. \tag{2.5}$$

In the plant transfer function, τ_d is the time delay and T_c is the time constant. The time delay and the time constant will vary with engine speed and the engine load (the normalized air charge); see Jankovic and Kolmanovsky, 2009. At a given engine speed and load, τ_d and T_c are assumed constant.

To evaluate the quality of this model for closed-loop system design, we have compared the responses to step disturbance of an engine and of the corresponding simple model. The tests shown in Figures 2.17 and 2.18 were run at 1000 rpm, at the engine load of 0.16 (corresponding to transmission in neutral at idle, that is, 0 torque). For these conditions, the model used the time delay of 0.25 s and the time constant of 0.30 s. Figure 2.17 shows five responses of a fuel–air ratio to the injector slope change by 20%, which produces a step in the amount of injected fuel. The bold black trace is the response of the model. In the top plot, the injector slope is decreased, resulting in a lean FAR. In the lower plot, the injector slope is increased and the system now responds rich. In both cases the disturbance is rejected (after some time) by the standard PI controller. The gains of the PI controller are very conservative ($Kp = 0.18$, $Ki = 0.7$), resulting in the closed-loop system bandwidth of about 0.8 rad/s.

Figure 2.18 shows the engine (two runs) and model responses under the same conditions except that the PI gains are much higher: $Kp = 1.2$, $Ki = 4.5$, for the bandwidth of 5 rad/s. The ratio of Kp/Ki is kept the same as in the low gain case. Two things can be observed from Figure 2.18. First, the system with the high PI gains and no delay compensation is on the verge of instability. Second, the model seems to adequately capture the onset of instability.

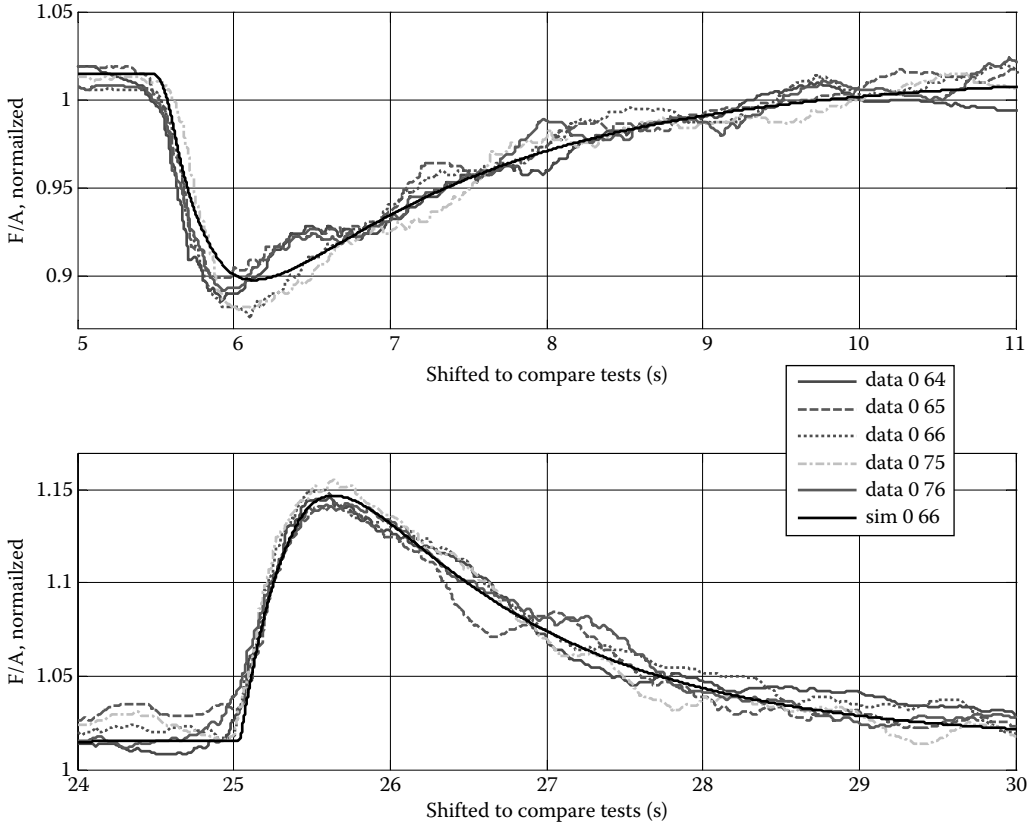

FIGURE 2.17 Comparison of engine and model disturbance response with a low-gain PI controller (no delay compensation).

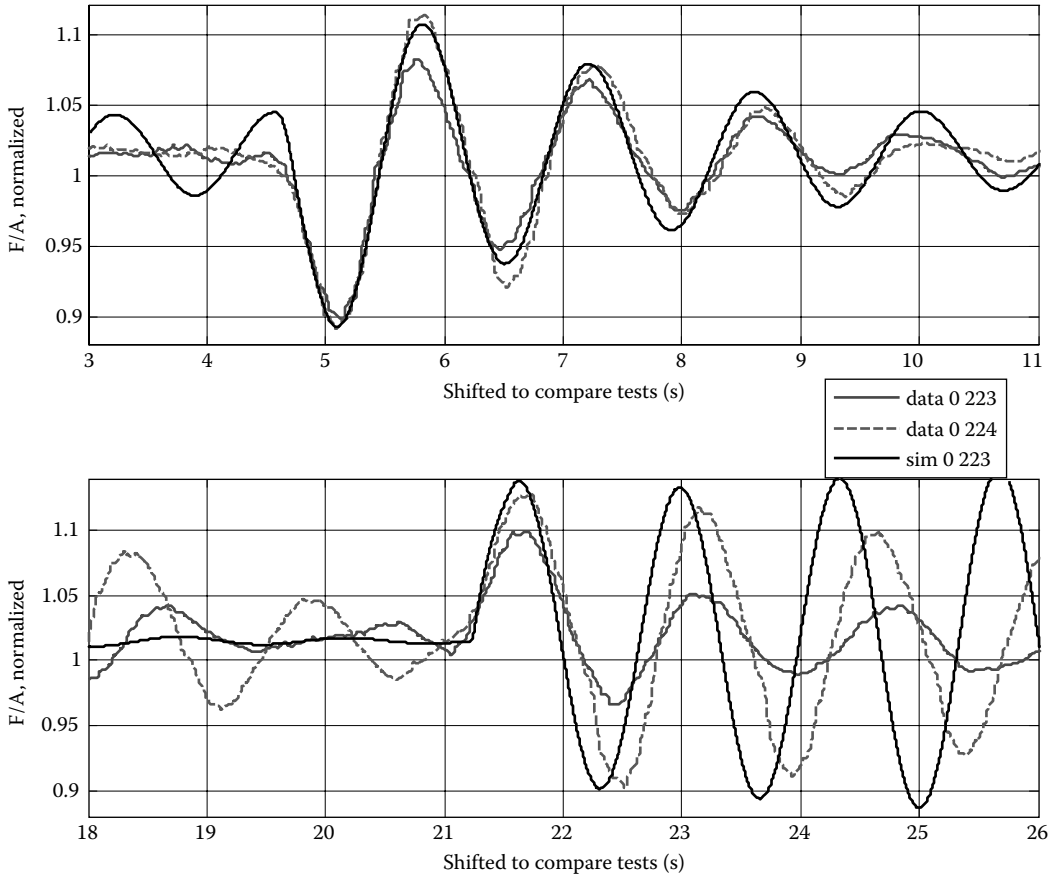

FIGURE 2.18 Comparison of the engine and model step with a high-gain PI controller (no delay compensation).

2.3.3.3 Smith Predictor

Faced with the bandwidth limitation due to time delay and the resulting slow disturbance rejection, one can try a delay compensation method such as the Smith Predictor. The controller structure is shown in Figure 2.19. With the PI controller transfer function given by $(Kp + Ki/s)$, the transfer function from the reference tracking error, $e = r - y = \text{FAR}_{\text{ref}} - \text{FAR}$, to the controller output u is

$$C(s) = \frac{u(s)}{e(s)} = \frac{(T_c s + 1)(K_p s + K_i)}{T_c s^2 + s + (K_p s + K_i)(1 - e^{-\tau_d s})}. \tag{2.6}$$

With this controller, the inner closed-loop transfer function shown in Figure 2.19 becomes

$$\frac{y(s)}{r(s)} = \frac{(K_p s + K_i)e^{-\tau_d s}}{T_c s^2 + s + (K_p s + K_i)}. \tag{2.7}$$

Except for the $e^{-\tau_d s}$ (delay) term, Equation 2.7 is exactly the same as the closed-loop transfer function obtained by using the PI controller for the first-order lag without delay. In other words, the Smith Predictor pulls the delay out of the feedback loop, removing the speed of response limitation to reference tracking.

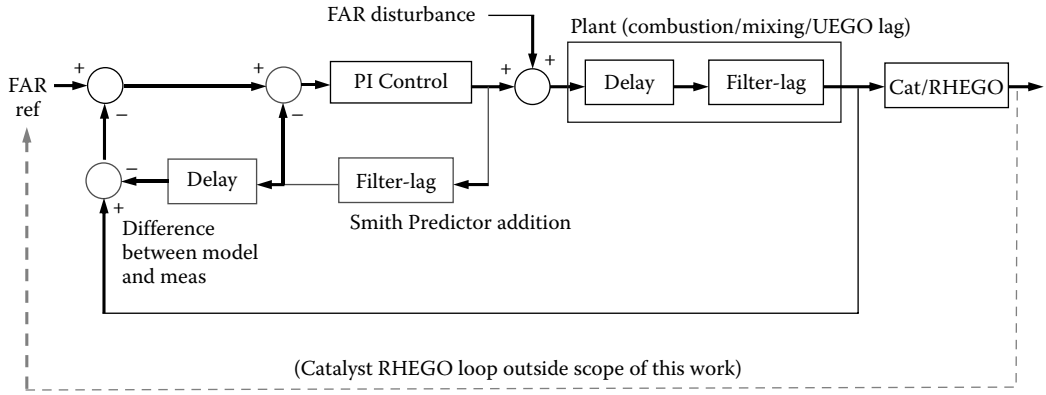

FIGURE 2.19 Block diagram of a Smith Predictor controller for the UEGO inner loop control system.

It is well known that the Smith Predictor tends to provide good results for reference tracking. On the other hand, its effectiveness for disturbance rejection depends on the plant open-loop poles (see, e.g., Chapter 10 (Z.J. Palmor) in the *Control Handbook*, 1996 (Levine, 1996) or in the current edition of this handbook, Section 9.8 of *Control System Fundamentals*). Nevertheless, the controller of the same structure has been proposed in Nakagawa et al. (2002) and Alfieri et al. (2007) for fuel–air regulation in gasoline and diesel engines, respectively, and we have also observed a tangible improvement in disturbance rejection as reported here.

With the Smith Predictor in place, we have dialed much higher PI gains ($Kp = 3.6$, $Ki = 12$) than those shown in Figure 2.17 or Figure 2.18. The bandwidth to the reference tracking is now 12 rad/s. Figure 2.20 shows that the disturbance response, obtained under the same conditions used for Figures 2.17 and 2.18, is about 4–5 times faster than in Figure 2.17. The small oscillations (visible also in Figure 2.17) are due to engine speed excitation in neutral idle. The Smith Predictor controller provides much faster response than what would be possible without delay compensation. To evaluate the impact on catalyst efficiency, the two controllers have been compared on the emission cycle. The conventional PI has production-like tuning with gain scheduled *Kp* and *Ki*. The Smith Predictor has fixed PI gains, reduced by 50% from those shown in Figure 2.20, while the delay τ_d and the time constant T_c for Equation 2.5 are scheduled on speed and load. Figure 2.21 shows the comparison over the first 150 s after the cold start sampled by the emission analyzer at 1 s rate. The tighter regulation by the Smith Predictor results in the corresponding improvement in catalyst efficiency. Figure 2.22 shows catalyst efficiency improvement in HC and NO_x, which means that the improvement is due to tighter FAR control, not rich or lean FAR bias. This is critical, because the majority of emissions over the cycle come out of the tail pipe in the first 100 s, before the catalyst starts functioning with high efficiency.

2.4 Transmission Controls

Automotive transmissions are an important part of the automotive powertrain. Their main function is to facilitate a better match between the engine and the vehicle. In particular, since a typical IC engine has a relatively low torque at low engine speeds, there is a need to amplify this torque significantly when transferring the engine torque to the wheels when the vehicle is at low speeds. For example, during a launch from standstill the transmission will typically be in the first gear which is characterized by the largest speed ratio, that is, largest amplification of the engine torque. This large torque will eventually be applied at the wheels in order to provide the largest possible acceleration propelling the vehicle through the

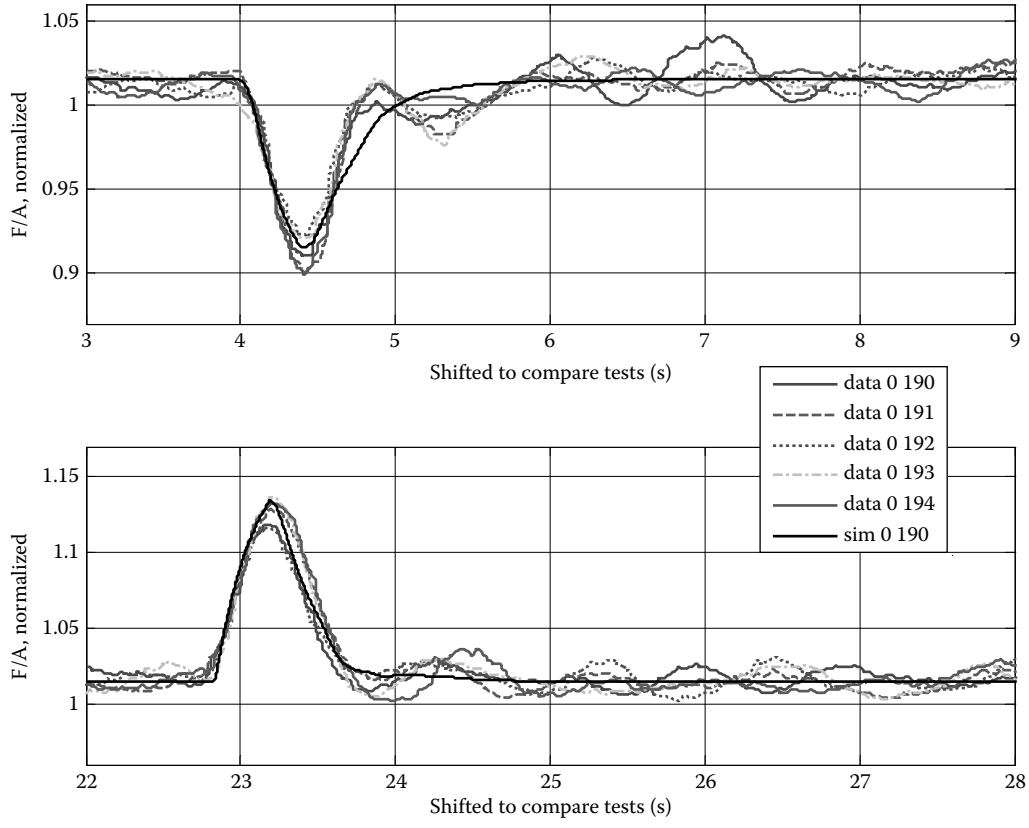

FIGURE 2.20 FAR disturbance rejection with the Smith Predictor.

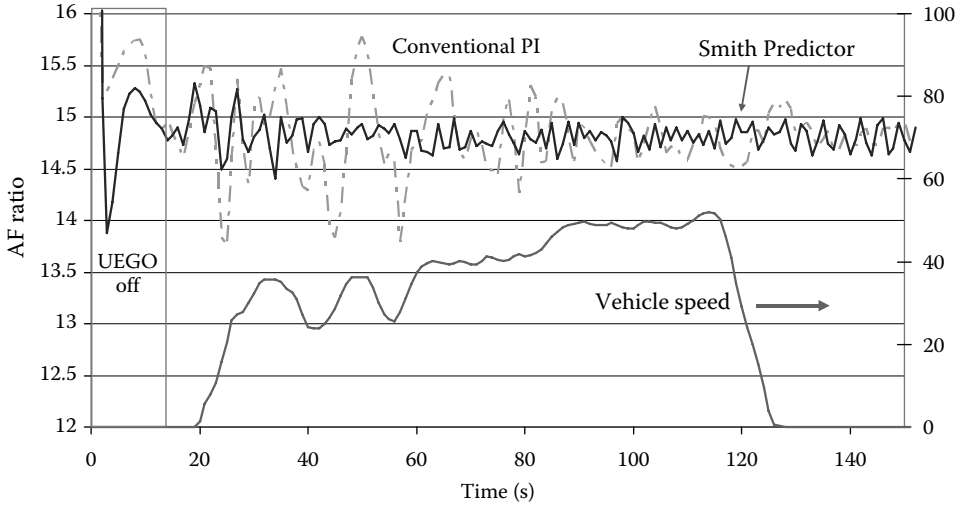

FIGURE 2.21 Comparison of the air–fuel regulation conventional PI and Smith Predictor controllers.

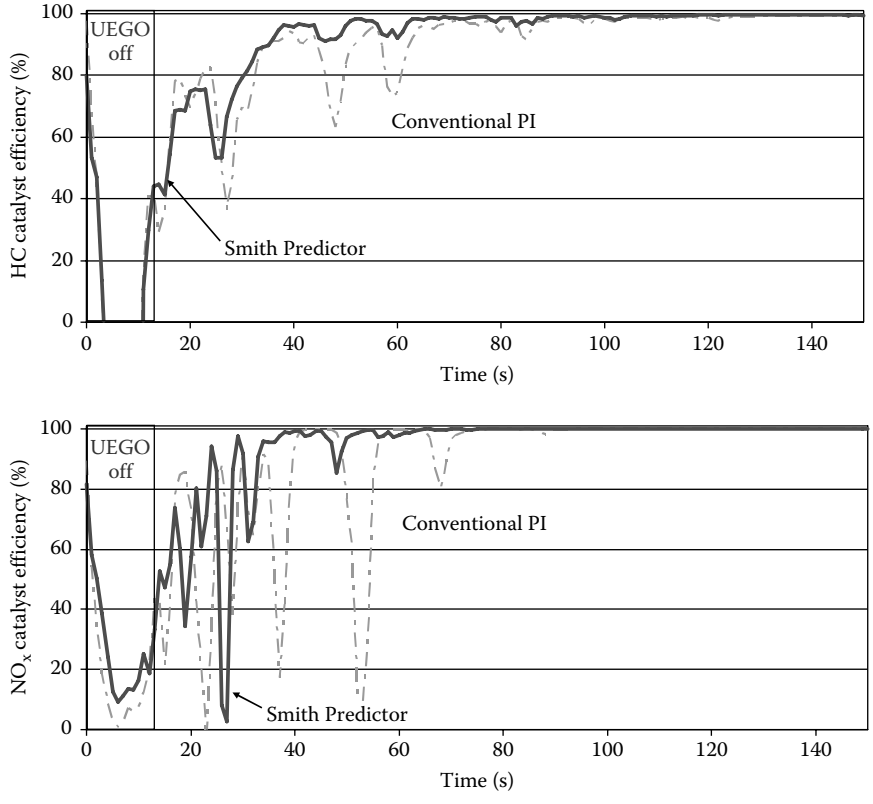

FIGURE 2.22 Comparison of HC (top) and NO_x (bottom) catalyst efficiencies between PI and Smith Predictor controllers.

launch. Depending on the vehicle class there are different numbers of transmission gears and associated torque ratio multiplications. While heavy-duty trucks may have up to dozens or more gears in order to accelerate their large mass especially when loaded, modern passenger vehicles, on the other hand, have typically 4–6 gears with the current trend toward a higher number of gears (7 and 8) in order to facilitate improved vehicle performance, fuel economy, and drivability/smoothness.

There are several kinds of transmissions. One distinguishes between manual and automatic transmissions, where different transmission actions are performed automatically without driver intervention. Among the automatic transmissions, there are discrete-ratio and continuously variable (or even infinitely variable) transmissions. The discrete-ratio transmissions have a finite number of gears used to achieve the above engine-vehicle matching. On the other hand, continuously variable transmissions, as their name implies, continuously change the transmission gear ratio to maintain engine operation at the optimal efficiency. One special class of continuously variable transmissions is found in some Hybrid Electric Vehicles (HEVs) where a continuous change in gear ratio is obtained through a single (or multiple) planetary gear set appropriately linked to an IC engine, motors, and generators (Kuang and Hrovat, 2003). In some of these cases, when there are multiple planetary gears, one can think of a hybrid transmission which combines discrete ratio changes along with continuously variable ones.

The discrete-ratio automatic transmissions are the most widespread in the United States. They achieve two basic functions: one is to shift the gear in order to bring the engine into a more favorable operating region and the other is to facilitate a launch of a vehicle from a standstill position. The discrete ratio transmissions can be divided into two classes. The more traditional automatic transmissions are of the

planetary type where different parts of a planetary set (ring, sun, and carrier) are connected to different clutches, bands, and one-way clutches, which are engaged or disengaged at appropriate times in order to perform a shift. The launch is achieved with the help of a special device, the so-called torque converter, which conveniently and smoothly couples the engine with the driven wheels starting from zero vehicle speed onwards (Hrovat et al., 2000). Another class of discrete-ratio automatic transmissions, with layshaft architecture, has been introduced into production more recently. They are similar to the traditional manual transmissions with the exception that they have two clutches as opposed to the single clutch that is found in manual transmissions. With the two clutches one can achieve smooth, uninterrupted power shift (Hrovat and Powers, 1998; Hrovat et al., 2000). These transmissions are also known as dual clutch transmissions (DCTs) or power shift transmissions.

Each of the above transmission classes has its own control challenges. While in the past, these challenges were addressed through hardware measures, such as a valve body consisting of a maze of passages with associated hydraulics, spools and valves, most of the modern-day automatic transmissions rely to a greater extent on microcomputers and embedded controls. For a typical discrete-ratio automatic transmission, the main control challenge is in the area of smooth and crisp/fast shifts and responsive and smooth launch. Since the DCTs or Power Shift transmissions typically do not have a torque converter, the launch control in this case is especially challenging. Here one can use both feedforward and feedback control. Feedforward control typically relies on detailed models of the engine and clutch torque production, whereas the feedback control, based on desired slip or engine and transmission input shaft speed, is used to compensate for any model inaccuracies and plant parameter changes (Hrovat and Powers, 1998; Hrovat et al., 2000).

There are many shifts in a typical modern 5–8 speed automatic transmission, and this includes numerous upshifts and downshifts between gears (e.g., 1–2 upshift, 5–2 downshift). Moreover, these shifts can occur under power conditions (i.e., with pressed gas pedal), in which case they are called power-on upshifts and downshifts, or these shifts can occur during coast-down (i.e., with closed gas pedal), in which case they are called power-off upshifts and downshifts. Since the whole shift lasts a relatively short time (typically well below 1 s) it is an interesting and challenging control problem of a servo type with important prestaging and poststaging phases. For example, in the case of 1–2 power-on upshift, there are two main phases (Hrovat and Powers, 1998; Hrovat et al., 2000). In the first or the so-called torque phase, the torque is transferred between two power paths corresponding to the first and second gear. In the case of layshaft or DCT transmissions, the torque transfer is done between two clutches, one attached to the first gear path ("odd clutch") and the second attached to the second gear path ("even clutch"). This transfer occurs within a few tenths of a second, and it is important that it is done in a precise manner to reduce any drivability shocks that could be felt by the vehicle occupants.

At present, the torque transfer is done mostly open-loop since there are no robust low-cost torque sensors suitable for production vehicles. This open-loop control design relies on appropriate embedded models of transmission dynamics and prestaging, that is, appropriate stroking of oncoming and off-going clutches before the beginning of the torque phase. Once the torque transfer has been completed, the off-going clutch, that is, the odd clutch, is fully opened and the oncoming, or even clutch, clutch is then used to initiate and control the second phase of the shift, the so-called inertia or speed-ratio change phase. During this phase, the engine speed should decrease from the first gear level to the second gear level, which typically amounts to hundreds of rpm change in less than half a second. This is usually done in a closed-loop manner, using either velocity or acceleration command profile, with appropriate filtering of the command to ensure smooth "landing" to a new operating level. Many different control techniques can be used for closed-loop speed or slip control implementation, from classical PID with adjustable parameters to "modern" robust controls such as H-infinity (Hibino et al., 2009). The shift control as well as the launch control can be further enhanced using observers for critical actuation variables such as clutch pressure (Hrovat and Powers, 1998). Additional details about this and other aspects of transmission control—such as torque converter bypass clutch slip control and transmission engagement—can be found in (Deur et al., 2006; Hibino et al., 2009; Hrovat and Powers, 1998; Hrovat et al., 2000) and references

therein along with examples of actual implementations and tests, which demonstrate the flexibility and potential of computer control where one can easily change the duration of the shift in a controlled manner from sporty and fast to smooth and comfortable.

2.5 Drivability

Good drivability is a key vehicle attribute since it directly contributes to customer satisfaction and overall ability to effectively drive and control the vehicle. Drivability encompasses various aspects of vehicle operation including launch from standstill, gas pedal tip-in and back-out response, transmission shifts, and general vehicle response to driver commands imposed via different actuators. In the more strict sense of longitudinal (fore-aft) dynamics, the latter include gas pedal, brakes, shift lever, and similar, where one typically expects a firm and brisk response without excessive delays and oscillations/tremors, and smooth shifts in case of transmission interventions. In a more general sense of drivability that includes both lateral as well as vertical motion of a vehicle, the actuators include steering wheel and possible knobs for different (semi)active suspension settings. In this section we will limit the treatment of drivability to vehicle longitudinal motion only and give two examples of using control principles and associated embedded software to improve and enhance the drivability experience.

2.5.1 Tip-In/Back-Out Drivability

In this first example, we focus on improving vehicle drivability during typical gas pedal operations when relatively fast gas pedal applications—either in the direction of increased (tip-in) or decreased (back-out) engine torque—can lead to noticeable vehicle longitudinal oscillations. In particular, tip-in and back-out response of a typical manual transmission powertrain is characterized by slightly damped, fore-aft oscillations known as "shuffle mode" vibrations (Hrovat and Tobler, 1991; Hrovat et al., 2000). While originally limited to manual transmissions, this type of response also extends to other transmissions that can behave like manual transmissions under similar driving conditions. This includes Automatic Shift Manual (ASM), Power Shift (Dual Clutch) transmissions, and HEV powertrains. Similar to the manual transmission, all these type of transmissions are typically characterized by the absence of a torque converter, which provides adequate amount of shuffle mode damping needed for good drivability.

It should be pointed out that the main focus of the present example is on improving, that is, increasing, the shuffle mode damping, which is one of the main characteristics of good tip-in drivability. In addition, we are limiting ourselves to tip-ins from one positive torque level to the next, higher level. Under this condition, the drive train does not cross various backlashes that are present in the drive train. However, even when such crossing is present due to torque reversals during tip-ins (and back-outs), having increased shuffle mode damping is still very beneficial, all the more so since the shuffle mode excitation level after crossing the backlashes is now even larger.

An effective way to improve tip-in/back-out drivability consists of increasing the shuffle mode damping through structural modifications and/or active measures such as engine controls through spark advance/retard changes (Jansz et al., 1999). In order to obtain an overall picture about possible improvements by using the spark intervention, we will first neglect the spark actuation limits. Later we will treat these limits with soft constraints on control action within the linear-quadratic (LQ) optimization framework.

2.5.1.1 Model Description

A simple, control-oriented vehicle schematic relevant for the above tip-in/back-out dynamics is shown in Figure 2.23. It includes the powertrain unit consisting of engine and transmission along with key driveline components—differential and half shafts that transfer the engine torque to the wheels. The corresponding

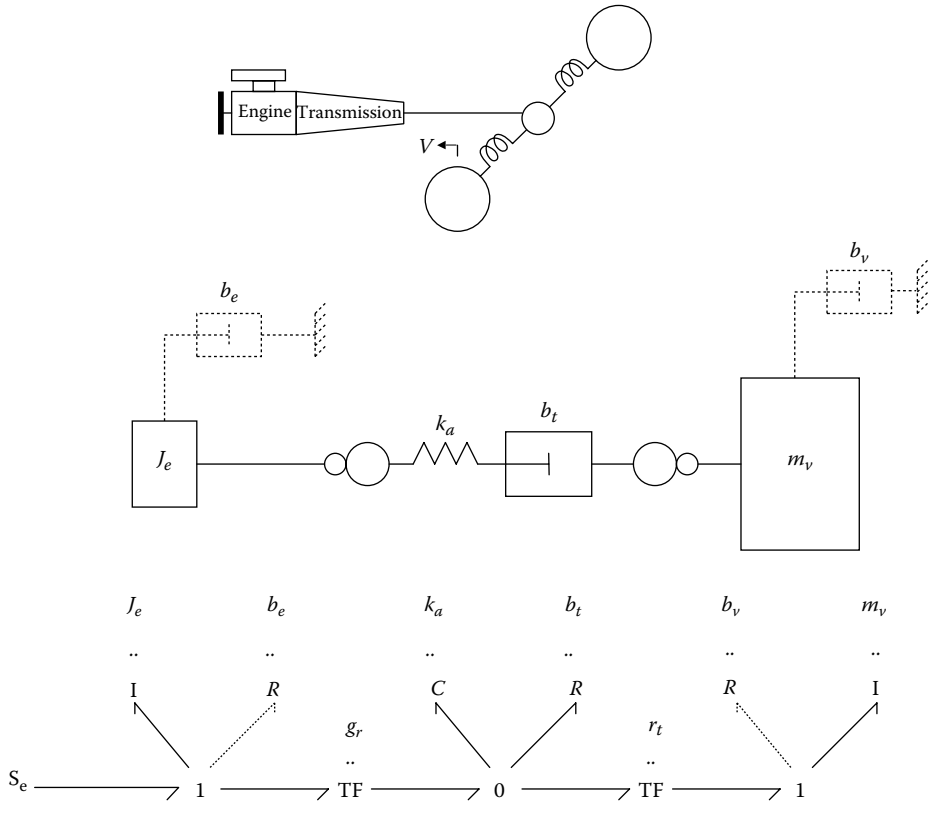

FIGURE 2.23 Powertrain configuration relevant for tip-in/back-out drivability, and corresponding linear-motion schematic and bond graph model.

model consists of three states (see below) and includes key elements necessary to capture the dominant, shuffle mode of powertrain/vehicle vibrations (Hrovat and Tobler, 1991; Hrovat et al., 2000). This includes effective engine-transmission inertia J_e, transformer (gear set) with the gear ratio g_r, effective/combined half-shaft stiffness k_a, tire resistance with an effective damping coefficient b_t, and vehicle inertia J_v represented by vehicle mass m_v. Note that the dominant shuffle mode damping is provided by tires that act *in series* with the dominant shuffle mode compliance provided by the half shafts. This in-series configuration is often missing in the literature, which typically shows the two acting in parallel. It can be shown that a given shuffle mode damping ratio can be represented by either configuration—serial or parallel. However, only the serial one will reflect the correct trends so that, for example, when the effective tire damping coefficient increases, the overall shuffle mode damping ratio decreases!

Additional engine and vehicle damping are also shown by dashed lines in Figure 2.23, since they are not essential for the present problem. That is, unless stated otherwise, it will be assumed that the engine damping b_e and vehicle damping b_v, due to air resistance and similar, are negligible. Similarly, we neglect the structural damping in various driveline components, which can lead to a slightly more conservative approach. The control is provided via an engine torque source S_e, which, in turn, can be provided via an electronic throttle actuator with associated manifold dynamics and/or spark advance/retard through appropriate actuation/computation delay (Jansz et al., 1999). For the purpose of the present study, these additional dynamic effects will be neglected since the main focus is on the driveline subsystem and suppressing excessive shuffle mode vibrations. These effects can be added later for final verification and implementation.

The bond graph (Karnopp et al., 2006) corresponding to the above model is also shown in Figure 2.23 where the transformers TF represent transmission and final drive with the total gear ratio g_r and tire with the transformer ratio equal to tire (loaded) ratio r_t. From this representation, it can be seen that the system can be described by three states: engine speed, ω in rad/s, half-shaft deflection, θ in rad, and vehicle speed, V in m/s. The corresponding state equations are

$$\begin{bmatrix} \dot{\omega} \\ \dot{\theta} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} -b_e/J_e & -k_s/J_e/g_r & 0 \\ 1/g_r & -k_s/b_t & -1/r_t \\ 0 & k_s/m_v/r_t & -b_v \end{bmatrix} \begin{bmatrix} \omega \\ \theta \\ V \end{bmatrix} + \begin{bmatrix} 1/J_e \\ 0 \\ 0 \end{bmatrix} \tau_{eng}. \tag{2.8}$$

The following data, characteristic of a compact-to-medium size car operating in second gear, will be and in our subsequent numerical examples: $J_e = 0.164$ kgm^2, $g_r = 8.159$, $b_t = 1400$ Nm s/rad, $k_s = 5038$ Nm/rad, $r_t = 0.27$ m, and $m_v = 1445$ kg. Note that the engine torque τ_{eng} [Nm] in Equation 2.8 simultaneously acts as system reference input imposed via the accelerator pedal, as well as the control input implemented via spark retard/advance control strategy to be designed in the next subsection. The system output of primary interest is the half-shaft torque $\tau_s = k_s\theta$, since this is what is primarily felt by the vehicle occupants. Thus the corresponding C and D matrices are

$$C = \begin{bmatrix} 0 & k_s & 0 \end{bmatrix}, \quad D = 0, \tag{2.9}$$

where the sole system output is equal to half-shaft torque, which is the main indicator of tip-in/back-out drivability and drive comfort felt by vehicle occupants.

2.5.1.2 Control Design

Two different control design approaches will be used in the present study. A state-space control based on pole placement for improved shuffle mode damping will be considered first. This will be followed by an LQ-optimal design incorporating soft constraints on control input (engine torque) resulting in a similar level of shuffle mode damping improvements.

2.5.1.2.1 Pole Placement Control

To establish a benchmark for a good tip-in response, we will first design a state-space-based controller using a pole-placement technique, assuming that no constraints are imposed on spark actuation control.

The main objective of the control design is to increase the shuffle mode damping, since the open-loop plant has very low damping in this mode. Indeed, the corresponding open-loop poles are at 0, and -1.7993 ± 22.4957i. The marginally stable pole at zero corresponds to the rigid body mode of the entire vehicle moving forward with a velocity V. If needed (primarily for numerical purposes) the damping of this mode could be increased by introducing either a non-zero engine damping, b_e, or vehicle damping, b_v. The complex conjugate pair of poles is of primary interest for the present drivability study since it results in a relatively low damping ratio of only 0.08 or 8%. The corresponding step response is shown in Figure 2.24 as a very oscillatory torque trace resulting in unacceptable drivability since the whole vehicle would "shuffle" in a back-and-forth direction as the driver would apply a slightly more energetic tip-ins and back-outs of gas pedal.

In order to improve the drivability we need to significantly increase the shuffle mode damping. To this end we choose the shuffle mode closed-loop poles at -11.2838 ± 19.5441i, which corresponds to a significantly larger damping ratio of 0.5 or 50%. This ratio was chosen instead of the more common 0.7, since it is desirable to have some overshoot on an accelerator pedal tip-in, giving an impression of a more aggressive or "peppy" powertrain. At the same time, the shuffle mode natural frequency was left unaltered at 22.5675 rad/s or about 3.6 Hz, which is typical of similar vehicles operating in lower gears, such as second gear considered in this example. The remaining "rigid body" pole was left at its open-loop location at zero.

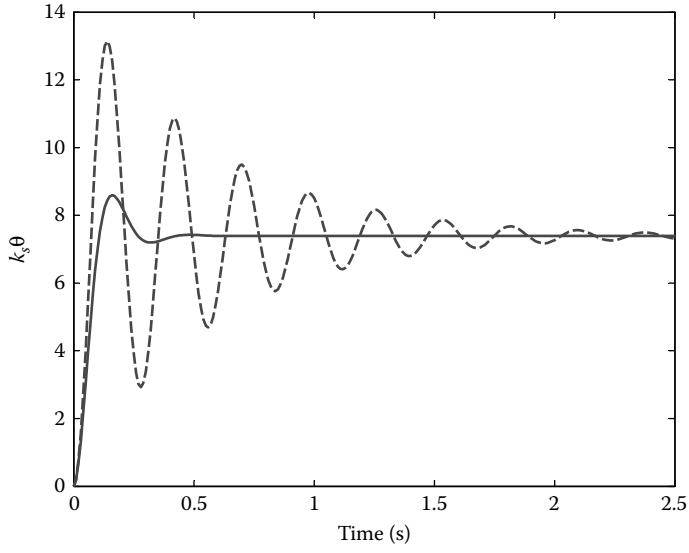

FIGURE 2.24 Step responses: Open-loop (dashed) and closed-loop with the pole placement controller (solid).

Using the above setup and standard CACSD tools such as MATLAB (MATLAB User's Guide, 1998) we obtain the closed-loop control gains as,

$$K = \begin{bmatrix} 3.11091 & -91.3369 & -94.0071 \end{bmatrix}. \tag{2.10}$$

The performance of the closed-loop system is compared in Figure 2.24 with the corresponding open-loop case for engine input consisting of a unit step in torque. It can be seen that the closed-loop response has much better damping with a desirable, slight overshoot as postulated above.

From Equation 2.10, observe that the ratio $K_3/K_1 = -30.2185$ is exactly equal to the ratio of g_r/r_t. This implies that the first and the third state, when properly reflected, are subject to the same gain so that only their difference is relevant to the controller. Note also that the gain on the half-shaft windup state x_2 is small relative to the. combined shaft stiffness $k_s = 5038$ Nm/rad and thus it can be neglected. All this implies that—at least w.r.t. the engine—the controller tries to emulate a "virtual damper" shown in Figure 2.25 acting on the relative speed across the drivetrain between the engine and the wheels (reflected upstream). Note that, to achieve better damping of the shuffle mode, there is no need for a "skyhook-like" damping term often used for optimal active suspension design (Hrovat, 1997). This "virtual damper" concept was also used for improved shift control (Hrovat et al., 2001); the above state-space based analysis and related LQ-optimal considerations discussed below confirm the effectiveness of this approach.

2.5.1.2.2 LQ Control

In order to prepare for the LQ-based control system design, we normalized the above model with the help of insight gained from the pole-placement results and bond graph modeling. The resulting schematic and bond graph model are shown in Figure 2.25. From them one can obtain the following state equations:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -b'_e/J'_e & -k_s/J'_e & 0 \\ 1 & -k_s/b_t & -1 \\ 0 & +k_s/J'_v & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1/J'_e \\ 0 \\ 0 \end{bmatrix} u, \tag{2.11}$$

with $J'_e = J_e\, g_r^2$, $b'_e = b_e\, g_r^2$, $J'_v = J_v\, r_t^2$, and $u = \tau'_e = \tau_e\, g_r$. The engine torque consists of an open-loop part and a closed-loop control part that is implemented via spark. The spark authority limits will be reflected

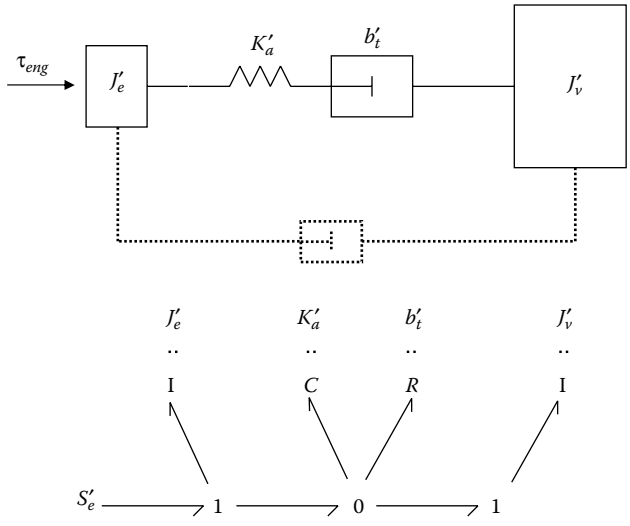

FIGURE 2.25 System model, showing the "virtual dampler" action of the controller.

in the LQ performance index through a soft constraint on control, that is, a quadratic term in u with the weighting R.

Next we chose the state weight matrix $\boldsymbol{Q}$ in the LQ performance index. Knowing that the main objective is to increase the shuffle mode damping, and with the insight gained from the above pole-placement design, we chose the state weighting matrix that will mainly penalize the relative speed $(x_1 - x_3)$ between the engine inertia (reflected downstream) and vehicle velocity (reflected upstream); this will be done via the quadratic term $q_1 (x_1 - x_3)^2$. In addition, we include a relatively small term that will penalize the half-shaft deflection, which is a representative for wheel torque that will be felt by vehicle occupants. This will be accomplished through a quadratic term, $q_2\theta^2$. Thus, the weighting matrix $\boldsymbol{Q}$ will have the following form

$$Q = \begin{bmatrix} q_1 & 0 & -q_1 \\ 0 & q_2 & 0 \\ -q_1 & 0 & q_1 \end{bmatrix} \times 10^5, \tag{2.12}$$

and the control weighting R will be set to $R = 1$.

We will next use this LQ problem formulation with vehicle data from Section 2.5.1.1,

$$A = \begin{bmatrix} 0 & -461.4668 & 0 \\ 1 & -3.5986 & -1 \\ 0 & 47.8259 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0916 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 5038 & 0 \end{bmatrix}, \tag{2.13}$$

and, after a few iterations, we find

$$Q = \begin{bmatrix} 0.5916 & 0 & -0.5916 \\ 0 & 0.001 & 0 \\ -0.5916 & 0 & 0.5916 \end{bmatrix} \times 10^5. \tag{2.14}$$

Based on the above data, the following LQ control gains have been obtained:

$$K_{CL} = \begin{bmatrix} 210.1106 & -687.632 & -210.1106 \end{bmatrix}. \tag{2.15}$$

Note that the control gains K_{CL1} and K_{CL3} have the same absolute value and different signs. This is expected since we are now dealing with the normalized state representation (Equation 2.11) where the two speed states have been properly reflected, that is, normalized. With control gains (Equation 2.15) the

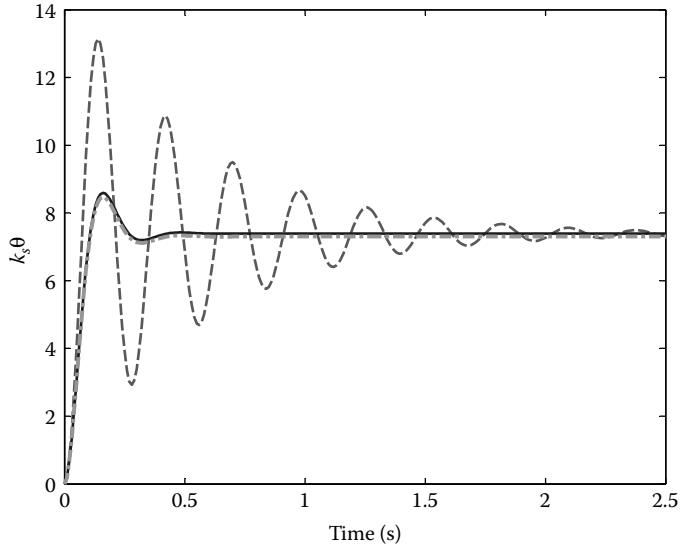

FIGURE 2.26 Step responses: Open-loop (dashed), closed-loop with the pole placement controller (solid), and closed-loop with the LQ controller (dash-dotted).

resulting closed poles are

$$p = -11.4224 \pm 19.6238i, \quad -2 \times 10^{-15}. \tag{2.16}$$

These poles are close to the previously discussed pole-placement case. This can also be seen from the corresponding step response traces shown in Figure 2.26, where the two closed-loop responses—one for pole placement and another for LQ case—are very similar and both significantly better damped than the original open-loop case.

2.5.1.3 Discussion

The above examples show that a simple but insightful model of an automotive drive train can be used to capture some of the key characteristics of tip-in/back-out dynamics leading to shuffle mode for-aft vehicle oscillations. Using this model and its normalized counterpart, it was possible to design a simple but effective and practical feedback controller that can substantially improve the shuffle mode damping and thus reduce the excessive vehicle oscillations that can lead to unacceptable drivability. The controller essentially amounts to having a virtual damper between the two main inertias elements while acting on the engine inertia only. The corresponding implementation does not require additional state observers if it is assumed that one could neglect the relatively small gain on the half-shaft deflection state. Indeed, in practice, the latter term should be small since any attempt to significantly change the effective stiffness via software, that is, algorithm intervention may not be efficient. In this case, it may be more appropriate to consider hardware changes by designing half shafts of different diameters and similar structural alterations. This is an example of implied interplay that often exists in practice between control/software and structural/hardware interventions where both are used, as appropriate, in a true sense of "system design."

2.5.2 Cancellation of VCT-Induced Air/Torque Disturbance

The discussion in Section 2.3 does not exhaust the set of transient issues brought about by adding optimization devices. For example, rapid VCT movement may produce torque surges and sags, resulting in unacceptable behavior. Figure 2.27 shows the VCT transient effect in a dual-equal VCT engine equipped

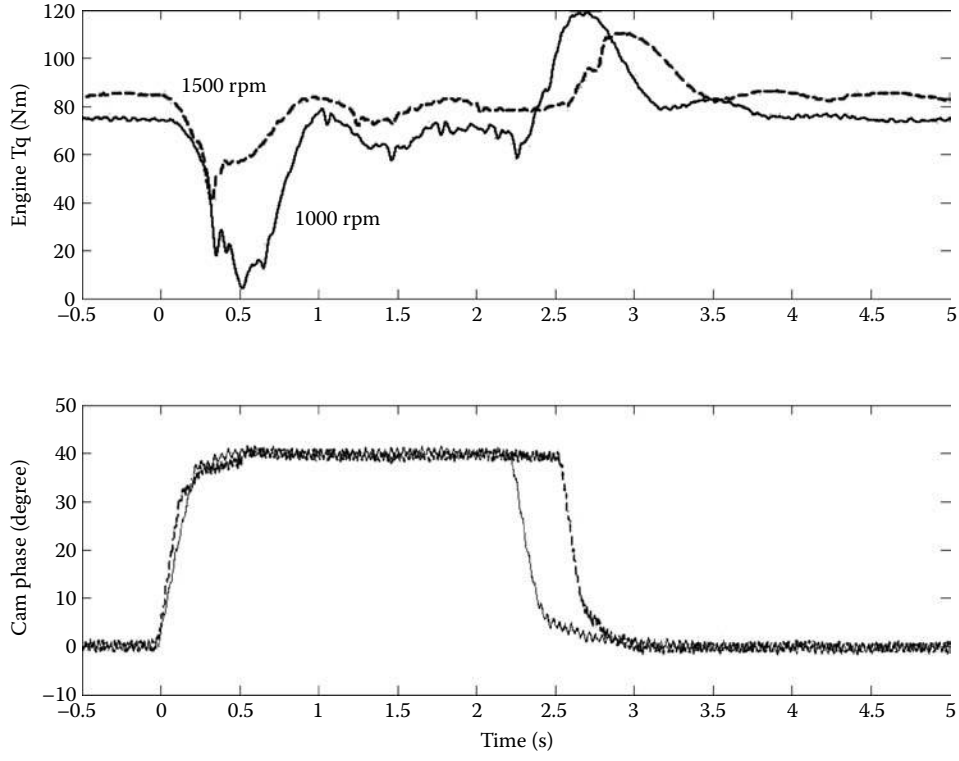

FIGURE 2.27 Engine torque response (top plot) to VCT transients (bottom plot) at 1000 and 1500 rpm.

with the single-degree-of-freedom VCT device (approximately coinciding with the dual-equal line in Figure 2.7). The plots illustrate a significant effect of cam timing that varies with engine speed.

Without a control system to remove this effect and restore acceptable vehicle drivability, the VCT schedules will have to be modified and the response slowed down. The end result is a degraded BSFC performance compared to the optimal one produced by the process described in Section 2.3.1.

The feedforward controller for this problem was proposed in Jankovic et al. (1998), the experimental results were reported in Jankovic et al. (2000), and the problem was recast in the format of nonlinear disturbance decoupling in Jankovic (2002). Here we shall provide a brief review of the method and the results.

2.5.2.1 System Model

The model of the system is the standard manifold filling equation (see, e.g., Section 2.3 in Guzzella and Onder, 2004) obtained from the ideal gas law: $PV = mRT$; P: pressure, V: volume, m: air mass, R: gas constant for air, and T: temperature. Under the standard isothermal assumption*, the intake manifold pressure rate of change is

$$\dot{P} = K_m(W_\theta - W_{cyl}), \tag{2.17}$$

where W_θ is the throttle mass flow rate, W_{cyl} is the cylinder mass flow rate, and $K_m = RT/V$. The throttle flow is the function of the throttle angle θ, and the throttle pressure ratio P/P_{amb}, with the additional

* Improved model accuracy is achievable if one replaces the isothermal with polytropic assumption for the intake manifold temperature as discussed in Deur et al. (2004) and the references therein.

dependence on ambient conditions suppressed:

$$W_\theta = g(\theta)\Psi(P/P_{amb}). \tag{2.18}$$

The subsonic correction factor Ψ is also standard (see, e.g., Guzzella and Onder, 2004; Jankovic, 2002). The throttle characteristic $g(.)$ is nonlinear, but invertible. In the dual-equal VCT engines, the flow into the cylinder is an affine function of manifold pressure (see Figure 2.6 in Jankovic and Magner, 2002):

$$W_{cyl} = \alpha_1(\zeta_{cam})P + \alpha_2(\zeta_{cam}), \tag{2.19}$$

with ζ_{cam} denoting the VCT position, and the slope α_1 and the offset α_2 dependence on engine speed being suppressed.

The effect of VCT on torque observed in Figure 2.27 is caused by its effect on the cylinder air flow W_{cyl}, given by Equation 2.19. If this impact is removed, the effect on torque response would be removed as well. An important feature of this problem is the difference between the two actuators that effectively control the engine air flow. The throttle is electrically actuated (ETC) and is fast, accurate, and very repeatable. The VCT mechanism is hydraulically actuated (see Section 2.3.1) and may be much slower than the throttle. Hence, the decision is made to command VCT based on the external signals, speed, and (desired) torque as in Equation 2.1, and suppress the air and torque disturbances by modulating ETC.

For a feedback solution, we note that the actual "performance" variables, the air-charge or torque are not measured. The next upstream variable, the manifold pressure can be measured, but the needed system bandwidth turned out to be very high and required high sampling rates. One reason is that, as the pressure approaches the ambient, its dynamics becomes very fast due to the large sensitivity of the throttle mass air flow to manifold pressure under such conditions. The VCT position is known and its impact on air/torque can be estimated. Thus, we have decided to consider it the disturbance to be cancelled (decoupled from the performance output) by a feedforward controller.

2.5.2.2 Feedforward Disturbance Decoupling

To design a controller that actuates the throttle to cancel the disturbance caused by VCT, we turn to the paradigm of disturbance decoupling [see Section 4.6 in Isidori (1989)]. Given a nonlinear system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + p(x)w, \\ y &= h(x), \end{aligned} \tag{2.20}$$

with x being the state, u the control input, and w a disturbance input, we consider the problem of finding a control law for u to reject the influence of the disturbance w on the output y. The key role in deciding if the disturbance decoupling problem is solvable is played by the concept of the "relative degree." The relative degree, denoted here by r, from an input variable u (or w) to the output y, tells us how many times we need to differentiate the output until the input u (respectively w) appears on the right-hand side. Thus, relative degree 1 means that y does not directly depend on u, but that $\dot{y}$ does. With the disturbance known (measured), the disturbance decoupling problem is solvable if the relative degree from the input to the output is not larger than that from the disturbance to the output. With the output of interest being the cylinder air flow W_{cyl}, it is clear that the relative degree from ζ_{cam} to the output is lower than that from the control input θ and the disturbance decoupling is not solvable. If we, however, model the VCT actuator as the first-order lag with time constant T_{VCT},

$$\dot{\zeta}_{cam} = -\frac{1}{T_{VCT}}(\zeta_{cam} - \zeta_{ref}), \tag{2.21}$$

and consider ζ_{ref}, which is a known quantity, the disturbance, we have the relative degree 1 from both the control and the disturbance input. That is,

$$\dot{W}_{cyl} = \alpha_1\dot{P} + \dot{\alpha}_1 P + \dot{\alpha}_2 = \alpha_1 K_m\left[g(\theta)\Psi(P/P_{amb}) - W_{cyl}\right] + \left(\frac{\partial\alpha_1}{\partial\zeta_{cam}}P + \frac{\partial\alpha_1}{\partial\zeta_{cam}}\right)\dot{\zeta}_{cam}.$$

Given that the throttle characteristic $g(.)$ is invertible and $\alpha_1 K_m \neq 0$, one can use the input θ to assign the desired response characteristic to the cylinder air flow. One possibility is to force W_{cyl} to respond like a reference model corresponding to $\zeta_{cam} = 0$. That is, we want W_{cyl} to track W^0_{cyl} generated by

$$\begin{aligned} \dot{P}^0 &= K_m(W^0_\theta - W^0_{cyl}), \\ W^0_{cyl} &= \alpha_1(0)P^0 + \alpha_1(0). \end{aligned} \tag{2.22}$$

Note that the throttle flow W^0_θ is generated by the conventional throttle movement θ^0. The solution for θ that makes W_{cyl} track W^0_{cyl} is given by

$$\theta = g^{-1}\left(\frac{\alpha_1(0)}{\alpha_1(\zeta_{cam})}\frac{g(\theta^0)\Psi\left(\frac{P^0}{P_{amb}}\right)}{\Psi(P/P_{amb})} - \frac{\left(\frac{\partial\alpha_1}{\partial\zeta_{cam}}P + \frac{\partial\alpha_1}{\partial\zeta_{cam}}\right)}{\alpha_1(\zeta_{cam})K_m\Psi(P/P_{amb})}\dot{\zeta}_{cam} + \frac{\alpha_1(\zeta_{cam})P + \alpha_2(\zeta_{cam}) - \frac{\alpha_1(0)}{\alpha_1(\zeta_{cam})}W^0_{cyl}}{\Psi(P/P_{amb})}\right). \tag{2.23}$$

For the implementation, this expression is simplified by assuming that the manifold pressure P is "correct," that is, it produces the desired cylinder air flow corresponding to zero VCT with the current (nonzero) value for ζ_{cam}:

$$P = \frac{(W^0_{cyl} - \alpha_2(\zeta_{cam}))}{\alpha_1(\zeta_{cam})}.$$

This makes the controller (Equation 2.23) completely feedforward. Using the measured value for P would have closed an unintentional feedback loop, introducing a risk of instability at pressures close to ambient.

The experimental results shown below are obtained by using the approximate derivative to generate $\dot{\zeta}_{cam}$ needed in Equation 2.23. The other option, to use Equation 2.21, was also tried successfully. Figure 2.28 shows the improvement in the engine torque response obtained by the disturbance decoupling system. In this test, only the VCT is changed while the reference model variables in Equation 2.22 stayed constant. The top plot compares the torque response between the disturbance decoupling system off (dash curve) and on (solid curve). Note that the controller achieves nearly flat torque response. That is, it decouples the torque (and the air flow) from the VCT disturbance. The corresponding throttle position traces are shown in the middle plot. The VCT movement (the disturbance) is shown in the bottom plot. A significant improvement was also observed when the reference model is not commanded to stay constant but is responding to a torque command change. For additional details and more experimental test results, see Jankovic et al. (2000) and Jankovic (2002).

The VCT impact on torque and vehicle drivability in this application is so severe that the optimal VCT schedules could not be used over a part of the operating region and the reference command ζ_{ref} had to be filtered to slow down VCT response. Implementation of the disturbance decoupling controller allowed restoration of optimal schedules and removal of filters resulting in cycle fuel economy improvement by more than 1%.

2.6 Diagnostics

Vehicles are subject to regulations that require most emission relevant components and systems to be monitored by the engine control unit (ECU) and any detected malfunction reported by setting a predefined code. The monitored items include all sensors, actuators, and devices on the engine, including catalytic convertors, fuel tanks, and evaporation control system (e.g., vapor canisters).

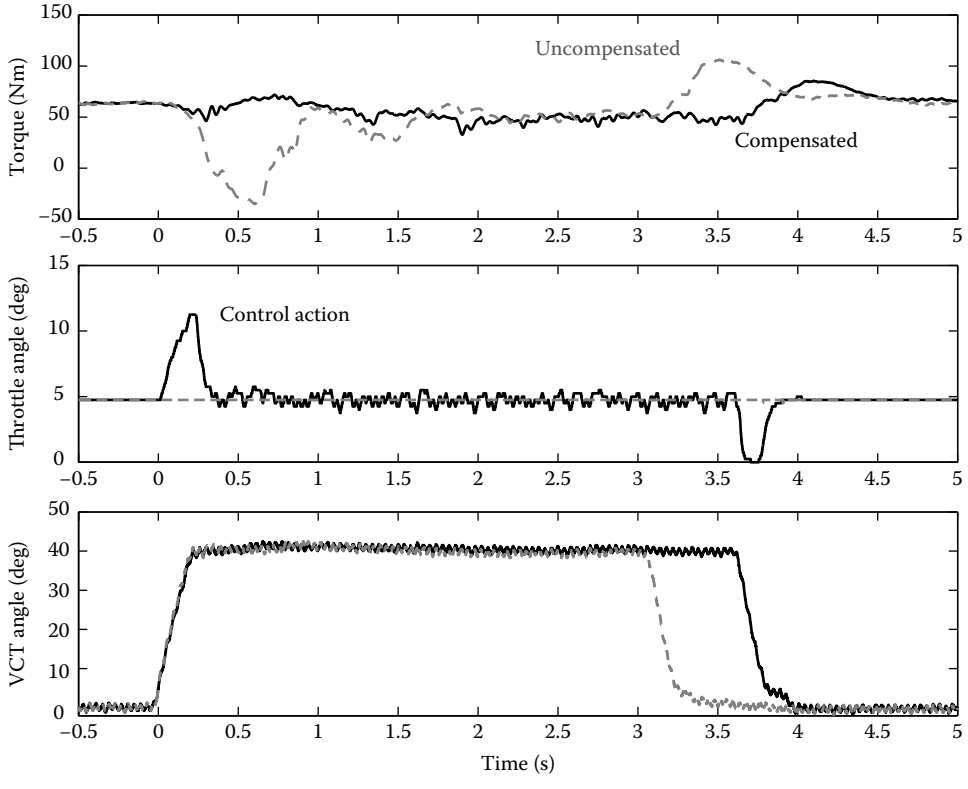

FIGURE 2.28 The comparison of system responses with the disturbance decoupling system off (dash) and on (solid).

2.6.1 Misfire Detection

Misfire detection is a substantial element of the automotive OBD-II system, first introduced in production for the 1994 model year. A misfire is defined as a condition during which a significant part of a cylinder's air–fuel mixture fails to ignite. Misfires, which result in increased emissions, must be diagnosed to meet legal requirements to correctly set the Malfunction Indicator Light (MIL) in the vehicle.

A conventional approach to misfire detection is to monitor the crankshaft accelerations resulting from the individual cylinder firings. The crankshaft accelerations are computed from the crankshaft position sensor, more specifically from the passage times between the teeth on the engine pulse wheel. Depending on the engine speed and location of the crankshaft position sensor, which may be at the front or rear of the engine*, the normally firing and misfiring cylinders may not be easily separable; see Figures 2.29a through c. This is especially true at higher engine speeds and for engines with a high number of cylinders.

Machine learning methods, in particular, artificial neural networks (ANNs) can provide an effective solution for the robust misfire detection for engines with a large number of cylinders. Using in-vehicle data with induced misfires, the neural network can be trained to correctly detect misfire patterns. Figure 2.30 illustrates schematically the neural network structure.

* *Note:* The rear location is more challenging due to torsional oscillations.

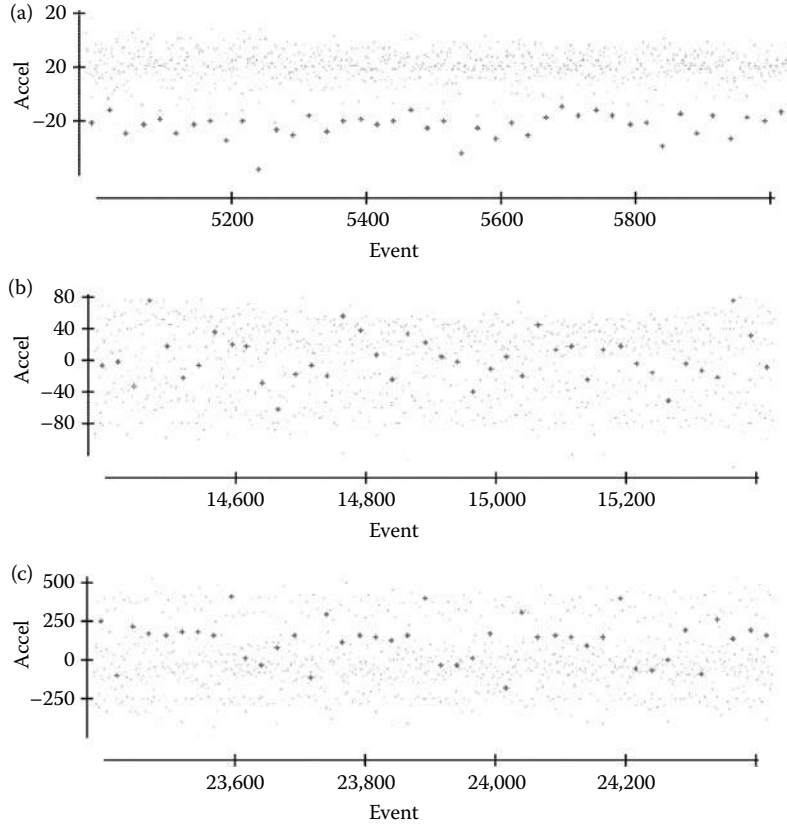

FIGURE 2.29 Crankshaft accelerations at (a) 1700 rpm, (b) 4000 rpm, and (c) 6500 rpm. Pluses correspond to engine misfires.

The effective ANN training can be based on the multistream Extended Kalman Filter (EKF) approach (Feldkamp and Puskorius, 1998) applied to the recurrent ANNs structure:

- With the EKF approach, the weights in ANNs are viewed as constant parameters in a dynamic system with a nonlinear output; these states can be estimated from input and output measurements, representing data used for ANN training.
- With the multistream training, the update of the weights is performed simultaneously from several data streams formed at random from the original dataset. This simultaneous update is achieved by considering the ANN as a nonlinear dynamic system with the vector output of each element being an exact replica of the ANN output and the total number of outputs is equal to the number of streams used for training.

Figure 2.31 demonstrates that the ANN solution can effectively separate misfire conditions from nonmisfire conditions.

2.6.2 VCT Monitoring

Beginning in 2006, the California Air Resource Board (CARB) requires that Low Emission Vehicles have monitors for their VCT systems (if so equipped). These monitors should run continuously to detect a malfunction in actuator response which would produce emissions that exceed 1.5 times the applicable standard if the vehicle was run on an emission test cycle such as FTP 75.

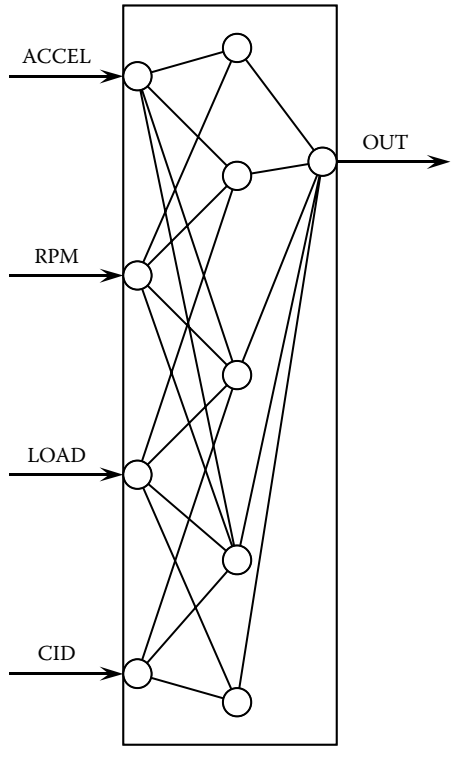

FIGURE 2.30 A schematic representation of an ANN for Misfire Detection. Crankshaft acceleration signal, engine speed signal, load signal (representative of engine air charge relative to maximum air charge at a given engine speed), and cylinder number are the inputs.

The monitor may have to detect a malfunction while running a different drive pattern than FTP 75. Hence, the regulations allow that some trips do not provide sufficient information to detect degradation. Instead, they require reporting a "performance rate" that tells how often the conditions, which allow the monitor to detect a malfunction, are met. The performance rate is determined as the ratio of the number of trips that meet the detection standard (the "numerator") to the number of "eligible" trips meeting certain conditions on duration, vehicle speed, and ambient conditions (the "denominator"). For a VCT threshold monitor, the performance rate is required to exceed 0.336, which means that the system should be able to detect a potential malfunction of the VCT system in more than one third of the eligible trips. This section reviews the approach from the U.S. Patent (Magner et al., 2007) to address these requirements.

2.6.2.1 Threshold Monitor

VCT affects engine feed-gas emissions through its effects on burned gas dilution and combustion quality. VCT faults can result in an increase in one or more of the regulated gas components through the following three mechanisms:

1. Cam position error resulting in the engine not retaining enough residual dilution gas (burned gas from the previous cycle), which may increase NO_x emissions. In the engine considered here (dual-equal VCT), over-advanced VCT causes lower than expected residual.

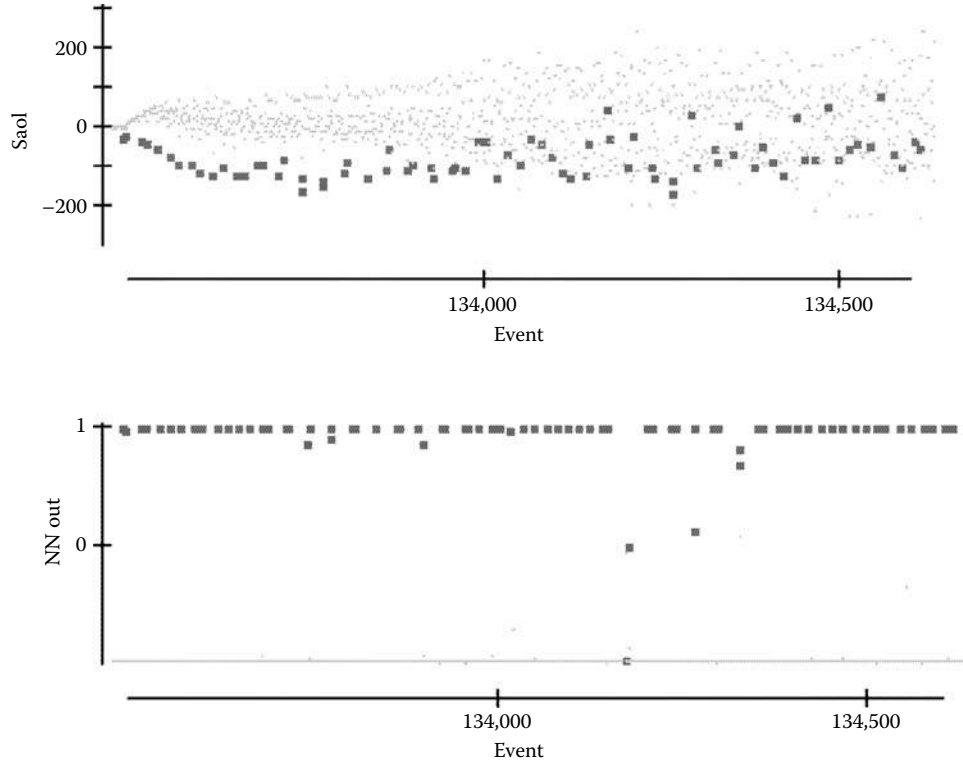

FIGURE 2.31 Performance of ANN on a test with increasing engine speed. Top: Accelerations with engine misfires indicated by red dots. Bottom: ANN output showing that misfire conditions are easily separable from the nominal combustion event.

2. Cam position error resulting in the engine receiving too much residual due to overretarded VCT. At lower loads and engine speeds, too much residual may result in partial burn that increases HC and CO emissions.
3. Different cam positions in the two banks of a V-engine may produce increased emissions because spark and (open-loop) fuel are typically not computed for each bank separately and could be incorrect for both.

A properly operating VCT system will have some time lag between desired (*cam_ph_d*) and measured (*cam_act*) cam position. Thus, we do not wish to penalize the VCT system that operates normally, as indicated by the *cam_expect* signal. Figure 2.32 provides an example of how several nonideal characteristics such as an offset, slow response, and higher-frequency oscillations can be characterized in terms of their impact on emissions.

The vehicle emissions over the cycle are a cumulative measure. That is, a more severe, but brief VCT degradation is likely to have a lower impact on test emissions than a less severe, but much longer lasting one. Hence, we found that the metric that best correlates with emission production is the integral of the VCT error over the cycle. It is expected that a small deviation from an ideal VCT response results in little if any emissions increase, while a large deviation results in disproportionately larger emissions increase. In response, instead of accumulating (integrating) the absolute value of the error we will accumulate the square error. Note that simple integration is unacceptable since even a near-perfect VCT operation will, after a sufficiently long time, reach an integrated level that is above any error limit. An integrator with a "forgetting" factor is thus employed. The recursive implementation of this algorithm is practically the same as a first-order low-pass filter and is implemented as such. Hence,

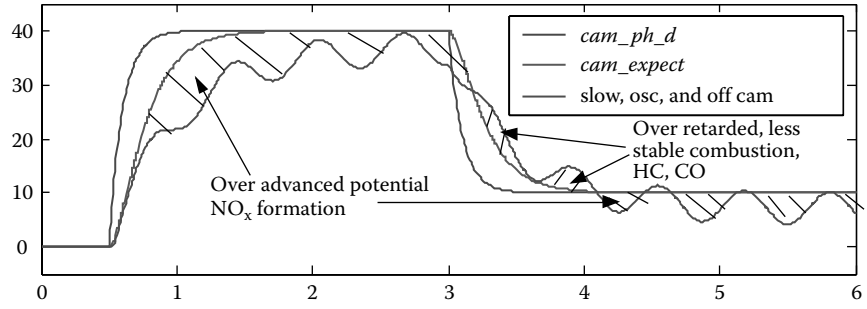

FIGURE 2.32 Illustration of overadvanced and overretarded errors.

each type of error (overadvanced, overretarded, and the difference between the banks in a V-engine) will have a low-pass filter accumulator producing the error index. Each index is to be compared with a predetermined, calibratable threshold that is used to set an OBD code indicating malfunction that could result in emissions exceeding 1.5 times the regulation on the emission cycle. The structure of the system is shown in Figure 2.33.

To illustrate the operation of the diagnostic system we have introduced an (artificial) VCT fault in which one bank is prevented from retarding beyond 26 (crank) degrees. Such a fault could occur if the camshaft in this bank is offset by two teeth in the advance direction (factory mis-build), therefore limiting the retard by the end stop. Figure 2.34 shows vehicle speed (top) on the FTP-75 cycle. The middle plot in Figure 2.34 shows the VCT positions on the two banks in which one tracks the desired, while the other is limited to 26 deg retard (and hence suffers from overadvance error). The bottom plot shows the accumulated value of the overadvanced index for the two banks. The large difference between the index values for the two banks (by a factor of 30–50) illustrates good detection capability of the system. In this case, the OBD code, indicating the malfunction, would have been set at about 250 s after start.

2.6.2.2 Performance Rate Monitor

Another important function of the VCT threshold monitor is to determine the "performance rate," or how often the vehicle is driven under conditions that allow the threshold monitor to detect a malfunction.

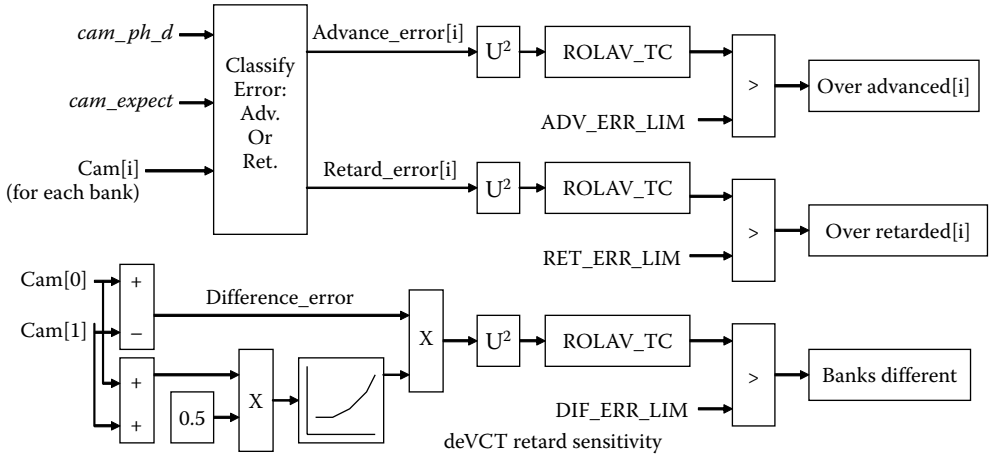

FIGURE 2.33 Signal processing from each cam error to setting the OBD codes.

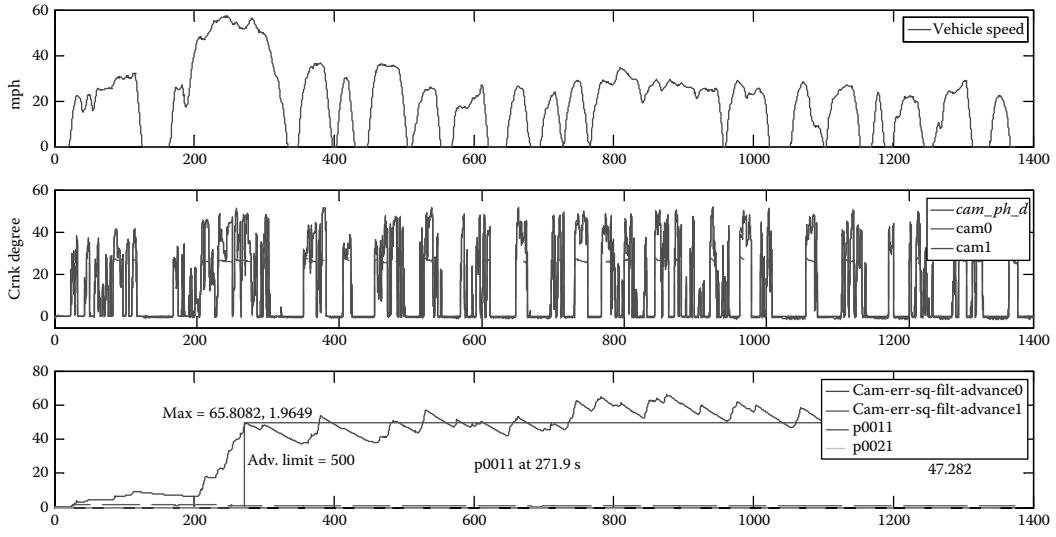

FIGURE 2.34 Bank 0 cannot retard past 26 deg, creating an overadvanced error on this bank.

Intuitively, it is clear that if the scheduler does not ask the VCT actuators to change the position, it may not be possible to tell if they are working correctly. In general, the performance rate numerator should be incremented only when the VCT excitation is sufficient so that a malfunction can be detected.

We would like to argue that this problem is related to a parameter identification problem and use the notion of "persistence of excitation," which can be computed from the measured (or known) signals, to tell if the parameters (and therefore actuator malfunction) can be robustly estimated/detected. Now consider the problem of identifying the parameters θ_1 and θ_2 from measurements of the signals y and x that change with time instant i and are related by the following equality (ε is a random noise):

$$y(i) = \theta_1 + \theta_2 * x(i) + \varepsilon(i). \tag{2.24}$$

Let us relate the signal x with the cam phase desired, and y with the actual cam position. Note that in the presence of noise, the parameters cannot be estimated unless $x(i)$ varies over sufficiently large range. That is, we do not want cam desired signal [i.e., $x(i)$] to stay at one value with only short excursions that do not allow the threshold monitor to detect a fault because there is not enough time for the error to accumulate. Another way to relate the fault detection problem to line parameter estimation is to consider under what conditions on measured signals can we distinguish the case of normally functioning actuator (*cam* $\approx$ *cam_ph_des*; i.e., $\theta_1 \approx 1$ and $\theta_2 \approx 0$) from, say, a stuck actuator (*cam* $\approx$ constant; i.e., $\theta_1 \approx 0$ and $\theta_2 \approx$ constant).

The condition for estimating the two parameters in Equation 2.24 is well known and can be obtained from the persistence of excitation determinant [cf. (Ioannou and Sun, 1996)]. To prevent summation (integration) to infinity, and to make sure we can catch a fault that occurs after a long fault-free drive, the old measurements of $x(i)$ and $y(i)$ [that presumably correspond to different (old) parameter values] are discounted. This is done by introducing a forgetting factor λ. After taking n measurements, the persistence of excitation conditions for identifying the two parameters is given by

$$det \begin{bmatrix} \sum_{i=1}^{n} \lambda^{n-i} & \sum_{i=1}^{n} \lambda^{n-i} x(i) \\ \sum_{i=1}^{n} \lambda^{n-i} x(i) & \sum_{i=1}^{n} \lambda^{n-i} x^2(i) \end{bmatrix} > E,$$

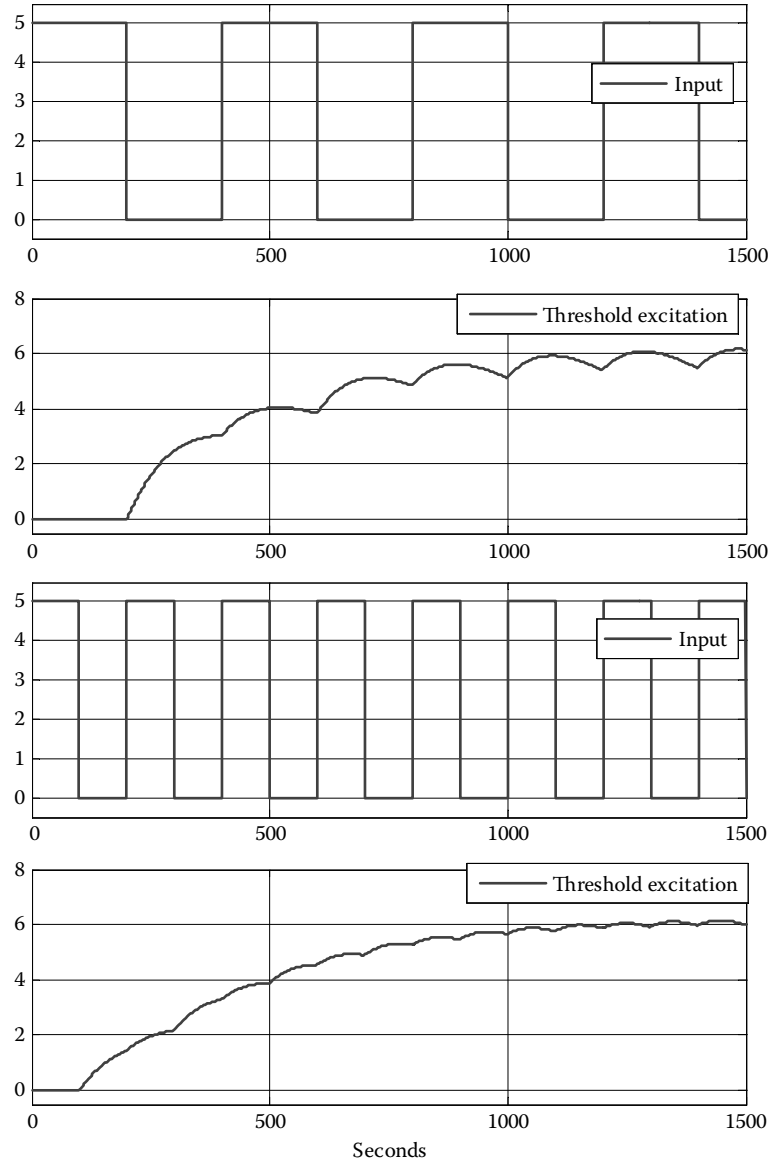

FIGURE 2.35 The same target excitation level is reached if the range of VCT is fixed, but the frequency of the input varies.

where E is the excitation threshold selected such that, under most common driving conditions, the performance rate does not increment before the threshold monitor is ready to declare a malfunction (if one is present).

An example of how this target excitation monitor operates is given in Figure 2.35. For an input, such as *cam_ph_d*, the frequency of change of the input (in this case the target value) plays no essential role in the final value of the determinant denoted by *te_excitation*. The set of plots at the top are for the lower frequency input. The value of the determinant ultimately reaches the same plateau in both cases.

2.7 Conclusion

A modern automotive powertrain relies on a computer control system for its operation. The control system assures desired response to driver's demand, optimal settings for various actuators to improve fuel economy and emissions, smooth transitions between operating points, handling of nonstandard situations, and system diagnostics. Increasing stringency of requirements, growing engine and transmission hardware complexity, and increasing computational power brought the need and the opportunity for the development and implementation of advanced control and estimation algorithms and associated optimization and tuning methods.

The purpose of this chapter was to illustrate several of these new developments and discuss the observed benefits. The examples considered covered a wide range of problems. They included mapping and optimization of HDOF engines, which is performed during powertrain development phase. Feedback regulation is illustrated by the idle speed control and air–fuel ratio control problems. Handling powertrain transients that affect vehicle drivability was illustrated by a feedback design for damping driveline oscillations and feedforward disturbance rejection in VCT engines. The problems related to engine emission diagnostics were also considered, including algorithms for the neural network misfire detection and for the VCT threshold monitor.

The chapter provided an illustration for a variety of optimization and advanced control methods that can be applied to powertrain systems in the vehicles or during the calibration phase. They range from online gradient search, through feedback controllers with delay compensation, linear quadratic, nonlinear, adaptive, and Model Predictive Control and Artificial Neural Networks. Other opportunities exist to apply advanced ideas in control and estimation theory to these and other powertrain control and diagnostics problems.

The implementation of controllers based on advanced control techniques in the final products depends on many factors, including demonstrated performance improvements compared to existing controllers, robustness to numerous uncertainties and abnormal conditions, computational complexity, calibration complexity and efficiency, and usability by software and calibration engineers that may not be experts in advanced control. The methods featured in this chapter have either been implemented in the final product or, in the opinion of the authors, represent promising directions for future implementation.

References

Alfieri E., Amstutz A., Onder C.H., and Guzzella L., 2007. Automatic design and parameterization of a model-based controller applied to the AF-ratio control of a diesel engine, *Proceedings of American Control Conference*, New York, NY.

Ariyur K.B. and Krstic M., 2003. *Real-Time Optimization by Extremum Seeking Methods*, Hoboken, NJ: John Wiley & Sons.

Bemporad A., 2003. *Hybrid Toolbox—User's Guide*, http://www.dii.unisi.it/hybrid/toolbox.

Bemporad A., Morari M., Dua V., and Pistikopoulos E., 2002. The explicit linear quadratic regulator for constrained systems, *Automatica*, 38(1), 3–20.

Box G.E.P. and Wilson K.B., 1951. On the experimental attainment of optimum conditions, *Journal of the Royal Statistical Society, Series B*, 13, 1–38.

Deur J., Magner S., Jankovic M., and Hrovat D., 2004. Influence of intake manifold heat transfer effects on accuracy of SI engine air charge prediction, *Proceedings of ASME IMECE*, Anaheim, CA.

Deur J., Petric J., Asgari J., and Hrovat D., 2006. Recent advances in control-oriented modeling of automotive power train dynamics, *IEEE Transactions on Mechatronics*, 11(5), 513–523.

Di Cairano S., Yanakiev D., Bemporad A., Kolmanovsky I.V., and Hrovat D., 2008. An MPC design flow for automotive control and applications to idle speed regulation, *Proceedings of IEEE Conference on Decision and Control*, Mexico, pp. 5686–5691.

Dorey R.E. and Stuart G., 1994. Self-tuning control applied to the in-vehicle calibration of a spark ignition engine, *Conference on Control Applications*, Glasgow, UK, pp. 121–126.

Draper C.S. and Li Y.T., 1951. Principles of optimalizing control systems and an application to the internal combustion engine, *ASME*, 160, 1–16.
Edwards S.P., Grove D.M., and Wynn H.P. (eds), 1999. *Statistics for Engine Optimization*, London: Professional Engineering Publishing.
Feldkamp L. and Puskorius, G., 1998. A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering, and classification, *Proceedings of the IEEE*, 86(11), 2259–2277.
Fradkov A.L., 1979. Speed-gradient control scheme and its application in adaptive control problems, *Automation and Remote Control*, 40(9), 90–101.
Guzzella L. and Onder C.H., 2004. *Introduction to Modeling and Control of Internal Combustion Engine Systems*, Berlin: Springer-Verlag.
Hardle W., 1989. *Applied Nonparametric Regression*, Cambridge: Cambridge University Press.
Hibino, R., Osawa, M., Kono, K., and Yoshizawa, K., 2009. Robust and simplified design of slip control system for torque converter lock-up clutch, *ASME Journal of Dynamic Systems, Measurement, & Control*, 131(1).
Hrovat D., 1996. MPC-based idle speed control for IC engine, *Proceedings of FISITA Conference*, Prague, Czech Republic.
Hrovat D., 1997. Survey of advanced suspension developments and related optimal control applications, *Automatica*, 33, 1781–1817.
Hrovat D., Asgari J., and Fodor M., 2000. Automotive mechatronic systems, *Mechatronic Systems, Techniques and Applications: Vol. 2—Transportations and Vehicle Systems*, C.T. Leondes (Ed.), pp. 1–98, Amsterdam: Gordon and Breach Science Publishers.
Hrovat D., Asgari J., and Fodor M., 2001. Vehicle shift quality improvement using a supplemental torque source, *U.S. Patent* 6,193,628.
Hrovat D. and Powers W., 1988. Computer control systems for automotive powertrains, *IEEE Control Systems Magazine*, August 3–10.
Hrovat D. and Powers W., 1990. Modeling and control of automotive powertrains, *Control and Dynamic Systems*, 37, 33–64.
Hrovat D. and Sun J., 1997. Models and control methodologies for IC engine idle speed control design, *Control Engineering Practice*, 5(8), 1093–1100.
Hrovat D. and Tobler W.E., 1991. Bond graph modeling of automotive power trains, *The Journal of the Franklin Institute*, Special Issue on Current Topics in Bond Graph Related Research, 328(5/6), 623–662.
Ioannou P. and Sun J., 1996. *Robust Adaptive Control*, Englewood Cliffs, NJ: Prentice-Hall.
Isidori A., 1989. *Nonlinear Control Systems*, 2nd ed., Berlin: Springer-Verlag.
Jankovic M., 2002. Nonlinear control in automotive engine applications, *Proceedings of 15th MTNS Conference*, South Band, IN.
Jankovic M., Frischmuth F., Stefanopoulou A., and Cook J.A., 1998. Torque management of engines with variable cam timing, *IEEE Control Systems Magazine*, 18, 34–42.
Jankovic M. and Kolmanovsky I., 2009. Developments in control of time-delay systems for automotive powertrain applications, *Delay Differential Equations—Recent Advances and New Directions*, B. Balachandran, T. Kalmár-Nagy, and D. Gilsinn (Eds), Berlin: Springer-Verlag.
Jankovic M., and Magner S., 2002. Variable cam timing: Consequences to automotive engine control design, *Proceedings of 15th IFAC World Congress*, Barcelona, Spain.
Jankovic M., and Magner S., 2004. Optimization and scheduling for automotive powertrains, *Proceedings of American Control Conference*, Boston, MA.
Jankovic M., and Magner S., 2006. Fuel economy optimization in automotive engines, *Proceedings of American Control Conference*, Minneapolis, MN.
Jankovic M., Magner S., Hsieh S., and Koncsol J., 2000. Transient effects and torque control of engines with variable cam timing, *Proceedings of American Control Conference*, Chicago, IL.
Jansz N.M., DeLaSallle S.A., Jansz M.A., Willey J., and Light D.A., 1999. Development of drivability for the Ford Focus: A systematic approach using CAE, *Proceedings of 1999 European Automotive Congress*, Barcelona, Spain.
Karnopp D.C., Margolis D.L., and Rosenberg R.C., 2006. *System Dynamics*, 4th ed., New York: John Wiley & Sons.
Kiencke U. and Nielsen L., 2000. *Automotive Control Systems*, Berlin: Springer-Verlag.
Kolmanovsky I. and Yanakiev D., 2008. Speed gradient control of nonlinear systems and its applications to automotive engine control, *Transactions of SICE*, 47(3), 160–168.
Kolda T.G., Lewis R.M., and Torczon V., 2003. Optimization by direct search: New perspectives on some classical and modern methods, *SIAM Review*, 45, 385–482.
Kuang M. and Hrovat D., 2003. Hybrid Electric Vehicle powertrain modeling and validation, *Proceedings of the 20th International Electric Vehicle Symposium (EVS),* Long Beach, CA.

Kvasnica M., Grieder P., Baotic M., and Morari M., 2004. *Multi-Parametric Toolbox (MPT)* (Hybrid Systems: Computation and Control), 2993, 448–462, Lecture Notes in Computer Science.

Leone T.G., Christenson E.J., and Stein R.A., 1996. Comparison of variable camshaft timing strategies at part load, *SAE World Congress*, SAE-960584, Detroit, MI.

Levine W.S. (Ed.), 1996. *The Control Handbook*, Boca Raton, FL: CRC Press.

Magner S., Jankovic M., and Dosdall J., 2007. Method to estimate variable valve performance degradation, *U.S. Patent* 7,171,929.

MATLAB User's Guide, 1998. Natick, MA: The MathWorks, Inc.

Montgomery D.C., 2001. *Design and Analysis of Experiments*, 5th ed., New York: John Wiley & Sons.

Nakagawa S., Katogi K., and Oosuga M., 2002. A new air–fuel ratio feed back control for ULEV/SULEV standard, *SAE World Congress*, SAE-2002–01–0194, Detroit, MI.

Niculescu S-I. and Annaswamy A.M., 2003. An adaptive Smith-controller for time-delay systems with relative degree $n \geq 2$, *Systems and Control Letters*, 49, 347–358.

Pavkovic D., Deur J., and Kolmanovsky I.V., 2009. Adaptive Kalman filter-based load torque compensator for improved SI engine Idle Speed Control, *IEEE Transactions on Control Systems Technology*, 17(1), 98–110.

Peyton-Jones J.C., Makki I., and Muske K.R., 2006. Catalyst diagnostics using adaptive control system parameters, *SAE World Congress*, SAE-2006–01–1070. Detroit, MI.

Popovic D., Jankovic M., Magner S., and Teel A., 2006. Extremum seeking methods for optimization of variable cam timing engine operation, *IEEE Transactions on Control Systems Technology*, 14, 398–407.

Powers, W., 1993. Customers and Control, *IEEE Control Systems Magazine*, 13(1), 10–14.

Sepulchre R., Jankovic M., and Kokotovic P.V., 1997. *Constructive Nonlinear Control*, London: Springer-Verlag.

Spall J.C., 1999. Stochastic optimization, stochastic approximation and simulated annealing, *Encyclopedia of Electrical Engineering*, 20, 529–542, John Wiley & Sons.

Stotsky A., Egardt B., and Eriksson S., 2000, Variable structure control of engine idle speed with estimation of unmeasured disturbances, *Journal of Dynamic Systems, Measurement and Control*, 122(4), 599–603.

Teel A., 2000. Lyapunov methods in nonsmooth optimization, Part II: Persistently exciting finite differences, *Proceedings of 39th IEEE CDC*, Sydney, Australia.

Wright M., 1995. Direct search methods: Once scorned, now respectable, in *Numerical analysis*, D.F. Griffiths and G.A. Watson (Eds.), Longman, UK: Addison Wesley.

Yildiz Y., Annaswamy A., Yanakiev D., and Kolmanovsky I.V., 2007. Adaptive idle speed control for internal combustion engines, *Proceedings of American Control Conference*, pp. 3700–3705, New York, NY.

3

Vehicle Controls

Davor Hrovat
Ford Motor Company

Hongtei E. Tseng
Ford Motor Company

Jianbo Lu
Ford Motor Company

Josko Deur
University of Zagreb

Francis Assadian
Cranfield University

Francesco Borrelli
University of California, Berkeley

Paolo Falcone
Chalmers University of Technology

3.1 Introduction

Vehicle control systems typically include chassis components that influence vehicle dynamics in three directions: longitudinal, lateral, and vertical. These three degrees of freedoms are controlled by chassis actuators such as brakes, steering, and suspensions, respectively. Traditionally, they were all mechanically controlled. For example, steering was actuated by the driver turning a steering wheel, which then caused the hydraulics in a power steering unit to amplify the driver-imposed torque and to create a desired wheel/tire rotation at the point of contact with the road.

During the past couple of decades, the above mechanical actuations have gradually been augmented by electrical and electronics/mechatronics actuations. This created opportunities for applications of computer controls and associated software. The early computer controls applications started with relatively slow (low bandwidth) load leveling suspensions and antilock braking systems (ABS). Later these were augmented by active and semiactive suspensions and four-wheel steer vehicle controls. On the other hand, there was development in traction controls for improved performance and stable operations on various road surfaces. This was further extended toward full vehicle stability control where brake intervention on one side of the vehicle was introduced in order to improve yaw stability and controllability. Additional

enhancement was made possible by roll stability control (RSC) to further improve stability in the roll direction and mitigate possible degraded performance.

This chapter addresses various aspects of vehicle control systems, starting from modeling of vehicle dynamics and associated tire characteristics, to active suspension and vehicle stability controls, concluding with active steering control and related future advanced control applications.

3.2 Tire Modeling for Vehicle Dynamics

Vehicle motion is predominantly determined by the tire–road forces. Therefore, a proper tire model is one of the key elements of a complete vehicle dynamics model needed for various simulation and control design studies. A static tire model is usually sufficient for typical vehicle handling and control simulations. However, a dynamic model reflecting the 3D tire structural compliance and the tire–road friction contact dynamics may be needed for more precise simulations.

3.2.1 Static Tire Model

Static tire models can be divided into two main groups: empirical models and physical models. Empirical models include a set of formulae that fit the experimentally recorded tire static curves, while physical models are based on a physical representation of the tire–road friction contact.

Any static model* describes the longitudinal and lateral tire forces, F_x and F_y, as functions of the longitudinal slip s and the tire slip angle α (see Figure 3.1), where the longitudinal slip may be defined as

$$s = \frac{v\cos\alpha - r\omega}{v\cos\alpha}, \tag{3.1}$$

where v is the tire center speed, ω is the tire rotational speed, and r is the effective tire radius.

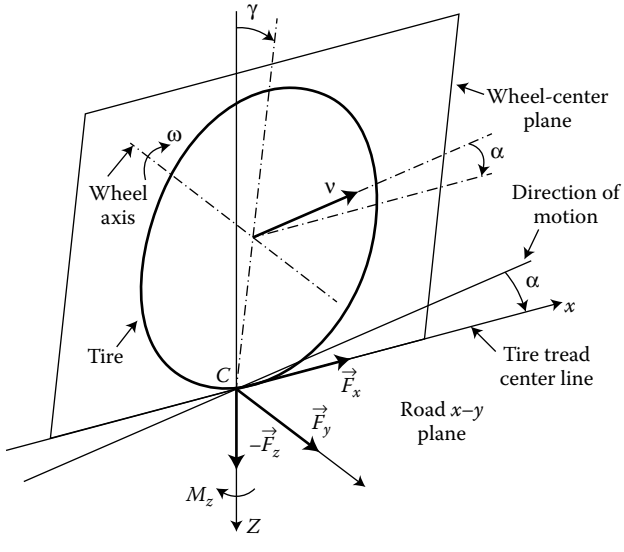

FIGURE 3.1 Coordinate system of tire.

* Modeling of the tire self aligning torque M_z is not presented herein. More details about the M_z-modeling approaches, which are similar to $F_{x,y}$-modeling, can be found in Bakker et al. (1987), Pacejka (2002), Pacejka and Sharp (1991), Deur et al. (2004).

The so-called "magic" formula model (Bakker et al., 1987) is probably the most widely used static tire model. A special trigonometric formula is first used to describe static curves $F_x(s)$ and $F_y(\alpha)$ for pure longitudinal motion and pure cornering, respectively. These basic curves are then combined in a semiphysical (Bakker et al., 1987) or again "magic" formula-based empirical way (Pacejka, 2002) to obtain the final static curves for combined longitudinal and lateral motion. For the sake of simplicity of presentation, a basic "magic" formula model is given herein:

$$\begin{aligned}
\sigma_x &= \frac{s}{(1-s)}, \\
\sigma_y &= \frac{\tan\alpha}{(1-s)}, \\
\sigma &= \sqrt{\sigma_x^2 + \sigma_y^2}, \\
F(\sigma) &= D\sin\left(C\arctan(B\sigma)\right), \\
F_x &= (\sigma_x/\sigma)F(\sigma), \\
F_y &= (\sigma_y/\sigma)F(\sigma).
\end{aligned} \tag{3.2}$$

This model includes only a single, simplified basic curve $F(\sigma)$ defined in the combined slip variable σ. The model parameters B, C, and D should be made dependent on different tire quantities such as tire normal load F_z, friction coefficient μ, and tire velocity v.

Physical models are typically based on the brush representation of tire–road friction contact (see Figure 3.2). When compared to the traditional brush models (Pacejka and Sharp, 1991; Pacejka, 2002), the recently proposed LuGre model has the advantages of a compact mathematical structure, accurate friction description, and ease of parameterization. The combined-slip model is represented by the following simple formula (Deur et al., 2004):

$$F_{x,y} = \frac{v_{rx,y}}{|v_r|}g(v_r)\left[1 - \frac{Z_{x,y}}{L}\left(1 - e^{-L/Z_{x,y}}\right)\right], \tag{3.3}$$

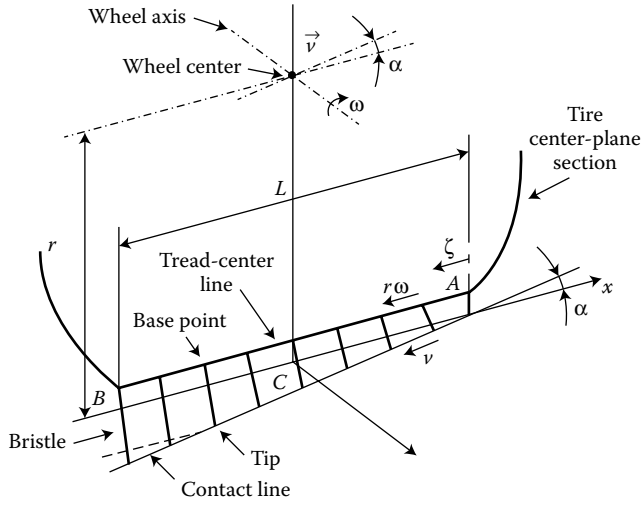

FIGURE 3.2 Brush model of tire.

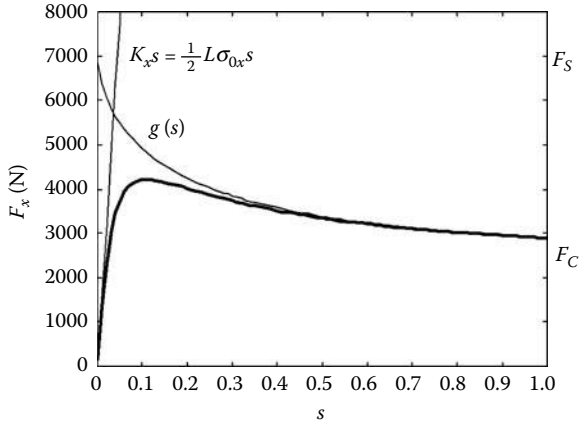

FIGURE 3.3 Construction of LuGre model static curve ($\alpha = 0$).

where L is the contact patch length (Figure 3.2), $v_{r(x,y)}$ are the slip speeds (cf. Figure 3.2), $g(v_r)$ is the tire–road friction potential function, and $Z_{x,y}$ is a length constant:

$$
\begin{aligned}
v_{rx} &= r\omega - v\cos\alpha, \\
v_{ry} &= v\sin\alpha, \\
v_r &= \sqrt{v_{rx}^2 + v_{ry}^2}, \\
g(v_r) &= F_C + (F_S - F_C)e^{-|v_r/v_s|^\delta}, \\
Z_{x,y} &= \left|\frac{r\omega}{v_r}\right| \frac{g(v_r)}{\sigma_{0x,y}}.
\end{aligned}
$$

As illustrated in Figure 3.3, the bristle horizontal stiffness parameter $\sigma_{0x,y}$ defines the zero-slip tire static curve gradient ("stiffness") and the sliding friction function $g(v_r)$ determines the high-slip (sliding) force values.

Figure 3.4 shows the comparative combined-slip static curves of a full "magic" formula model (Bakker, 1987) and the LuGre model given by Equation 3.3. If plotted in the coordinate system $F_y(F_x)$, the curves in Figure 3.4 would take on the realistic form of (friction) ellipse, while the basic "magic" formula model can only predict the friction circle. Figure 3.4 illustrates that the compact physical model (3.3) can provide an accurate prediction of more complex full "magic" formula model curves. The accuracy can be further improved by empirically based refinements (Deur et al., 2004). Figure 3.5 shows the comparative pure-braking curves for different tire normal loads F_z (Deur et al., 2004). Again, both models give very similar curves. Figure 3.6 illustrates the LuGre model static curve for different road conditions (Deur et al., 2004). Under a majority of road conditions, the tire behaves in the way that the tire curve stiffness K_x (cf. Figure 3.3) decreases with the decrease of tire–road friction coefficient μ, although there are some exemptions such as wet asphalt and dry ice conditions (Deur et al., 2005). In that regard, the overall tire force (Equation 3.3) should be scaled with the friction coefficient μ, rather than the friction potential function $g(v_r)$ only (Figure 3.6).

Figures 3.4 through 3.6 illustrate that the maximum vehicle longitudinal acceleration (driving) or deceleration (braking) with a reasonable large lateral (cornering) force can be achieved by keeping the longitudinal slip s at approximately 10%, which is the task of traction control systems (TCSs) or ABS, respectively. An optimized TCS or ABS will aim for different levels of desired slip depending on a driving condition, so that, for example, a smaller desired slip will be commanded when driving in a turn on a slippery road (Hrovat et al., 2000; Borrelli et al., 2006; Deur, 2009). On the other hand, under the

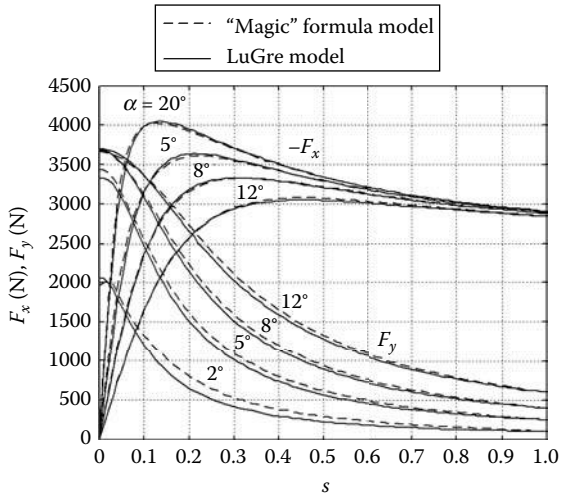

FIGURE 3.4 Combined-slip static curves obtained from "magic" formula and LuGre models (braking).

conditions of low friction coefficient, low normal load, and/or large longitudinal force, the lateral force F_y can easily be saturated, thus affecting the vehicle cornering stability. This can be overcome by a proper vehicle dynamics control (VDC) or electronic stability control (ESC) action.

3.2.2 Dynamic Tire Model

Figure 3.7 shows the structure of a dynamic tire model with respect to wheel torque variations. This is a two-mass elastic system comprising the wheel rim inertia I_a, the tire belt inertia I_b, the torsional sidewall compliance represented by the stiffness and damping coefficients k_θ and b_θ, and the tire friction dynamics. The main advantage of this model when compared to the static model is that it can predict a tire force lag behavior and provide computationally efficient simulations in a wide velocity range (Deur et al., 2004). The model in Figure 3.7 can be extended with the 3D tire dynamics including the longitudinal, lateral, and vertical compliance of sidewalls (Maurice, 2000; Pacejka, 2002).

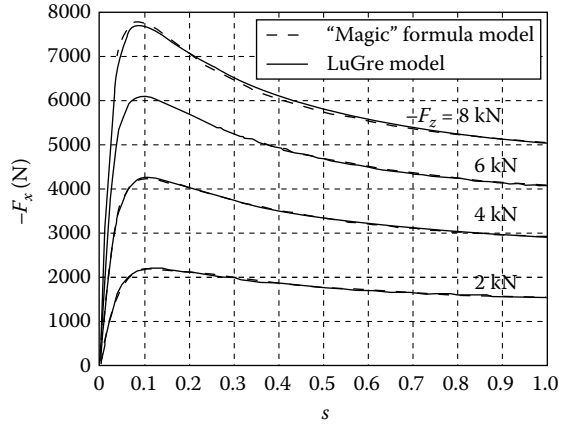

FIGURE 3.5 Pure-braking static curves for different tire normal loads.

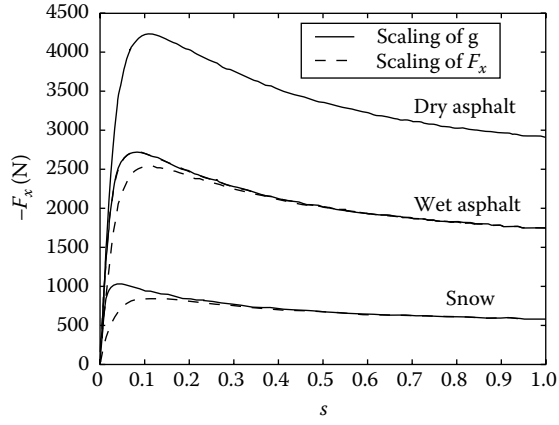

FIGURE 3.6 LuGre model pure-braking static curves for different road conditions.

The nonlinear tire friction dynamics may be modeled by extending the "magic" formula static model with a lumped-parameter tire tread (bristle) dynamics model (Bernard, 1995; Zegelaar, 1998; Maurice, 2000; Pacejka, 2002). The lumped parameters are "tuned" based on an analysis of distributed-parameter brush model behavior. Another, direct and potentially more accurate, approach is to develop a compact distributed-parameter model and transform it analytically to a lumped-parameter model. Such an approach is used in the LuGre model (Deur et al., 2004), and it results in the following final lumped-parameter combined-slip model:

$$\begin{aligned} \frac{d\tilde{z}_{x,y}}{dt} &= v_{rx,y} - \left[\frac{\sigma_{0x,y}\,|v_r|}{g(v_r)} + \frac{\kappa_{x,y}}{L} r\,|\omega|\right]\tilde{z}_{x,y}, \\ F_{x,y}(t) &= \sigma_{0x,y}\tilde{z}_{x,y}(t) + \sigma_{1(x,y)}\frac{d\tilde{z}_{x,y}(t)}{dt}, \end{aligned} \tag{3.4}$$

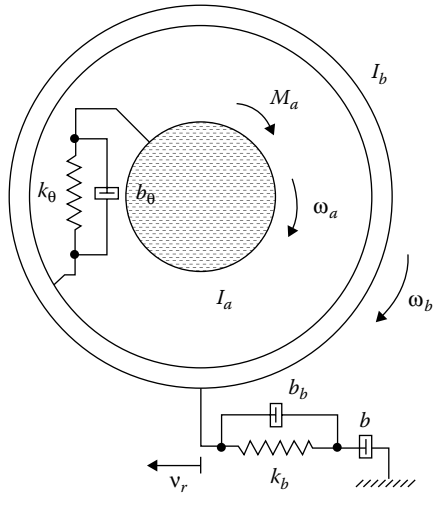

FIGURE 3.7 Principal model of tire dynamics with respect to wheel torque variations.

where $\tilde{z}_{x,y}$ is the average horizontal bristle deflection in the longitudinal and lateral direction, σ_1 is the bristle horizontal damping coefficient, and $\kappa_{x,y}$ is a characteristic lumped parameter given by

$$\kappa_{x,y} = \frac{1 - e^{-L/Z_{x,y}}}{1 - \frac{Z_{x,y}}{L}\left(1 - e^{-L/Z_{x,y}}\right)}.$$

The steady-state solution of the model 3.4 corresponds to the static LuGre model equations 3.3.

When applied to the longitudinal (in-plane) tire dynamics in Figure 3.7, only the longitudinal (x) equations are present in Equation 3.4, that is, $v_r = v_{rx}$ and $v_{ry} = 0$. The model linearization (Deur et al., 2009) reveals the structure of tire–road friction contact shown in Figure 3.7 and gives the following transfer function between the slip speed $v_r = r\omega_b$ and the longitudinal tire force F_x (p = Laplace variable):

$$G_x(p) = \frac{F_x(p)}{v_r(p)} = b\frac{b_b p + k_b}{(b + b_b)p + k_b}, \tag{3.5}$$

where the linearized model parameters b, k_b, and b_b in Equation 3.5 and Figure 3.7 are found to be related to the main features of the tire static curve in Figure 3.3 (with the slip s defined as $s = v_r/v$):

$$b = \frac{1}{v}\frac{dF_x}{ds}, \quad k_b \approx \sigma_0 \frac{dF_x/ds}{F_x/s}, \quad b_b \approx \sigma_1 \frac{dF_x/ds}{F_x/s}. \tag{3.6}$$

In addition to the well-known fact that the tire friction damping coefficient b is proportional to the tire static curve gradient dF_x/ds (Hrovat et al., 2000), the results 3.6 reveal that the equivalent stiffness and damping coefficients k_b and b_b depend on the tire operating point as well. More precisely, they depend on the gradient-to-secant ratio of the tire static curve. Note that the lag time constant $(b + b_b)/k_b$ is inversely proportional to the velocity v, that is, the lag effect is more emphasized at lower tire/vehicle velocities.

The overall transfer function of the model in Figure 3.7 is of the fourth order. It contains two vibration modes at approximately 40 and 90 Hz, for which approximate analytical expressions are given in Deur et al. (2009). The 40 Hz mode damping is related to the inverse of friction damping coefficient b. That is, the higher the slip s, the lower the damping coefficient b (see Equation 3.6 and Figure 3.3), and the higher the 40 Hz mode damping (note that the damper b is connected in series in Figure 3.7).

3.3 Vehicle Suspensions Control

There are three types of suspensions used on modern vehicles. The first is a conventional or so-called "passive" suspension type, which typically consists of springs and dampers at each corner of the vehicle with associated passive roll bars that are used for suppression of excessive vehicle roll during cornering or similar maneuvers. The second type of suspension is the so-called "active" suspension, which utilizes active power sources such as pumps and compressors in order to generate a desired suspension force or displacement. The third type, or so-called "semiactive" suspension, is essentially a controllable damper where the damping parameters can be changed to achieve desired suspension performance.

Automotive suspensions achieve several important functions of the vehicle. First, they act as a filter to reduce the road-induced vibration as well as excessive vehicle motion due to sudden steering action and possible road inclination. This way, the suspensions improve the vehicle ride attribute and occupants' ride comfort. Second, the suspensions also facilitate vehicle road following in all three dimensions (tracking both road grade and super-elevation of the road) which results in improved handling and overall maneuverability in emergency situations. With the increase of active and semiactive suspension applications, there will be new functions and unique capabilities emerging from these advanced suspension types and their integration with other vehicle control systems.

In this section, we will focus on using linear quadratic (LQ) optimal control methodology to investigate the potential benefits of active suspensions with respect to ride comfort and vehicle handling. Indeed,

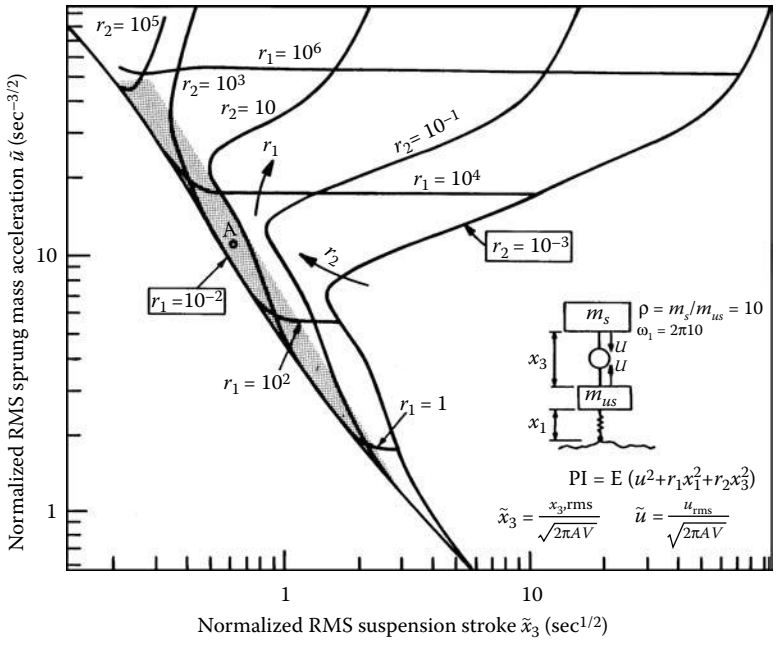

FIGURE 3.8 Quarter car model and sprung mass acceleration versus suspension stroke diagram.

as shown in [1], LQ methodology is well suited for this task since some of the key metrics for ride comfort are well represented by a quadratic term in vertical acceleration of the passenger compartment. By appending an additional quadratic term that reflects the design or packaging constraints, which limit the available space between suspension mounting points ("rattlespace"), and another quadratic term that penalizes excessive tire deflections in order to secure good handling, we will formulate the following LQ optimization problem.

Given a linear quarter car model shown in Figure 3.8, we can write the following state-space equations describing vehicle vertical dynamics,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ m_{us}x_2 \\ x_3 \\ m_s x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k_{us} & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} U + \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} w$$

or

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \rho \\ 0 \\ -1 \end{bmatrix} u + \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} w,$$

where x_1 is the primary suspension deflection (i.e., tire deflection), x_2 is the unsprung mass (wheel/tire/axle) velocity, x_3 is the secondary suspension deflection (rattlespace), x_4 is the sprung mass (main vehicle body) velocity, w is the ground input velocity due to road roughness, U is the active suspension force, and $u(= U/m_s)$ is the corresponding sprung mass acceleration (i.e., the normalized active suspension force).

In this context, the key vehicle parameters are k_{us}, m_{us}, and m_s, which represent tire stiffness, vehicle unsprung mass, and sprung mass, respectively. Furthermore, $\rho = m_s/m_{us}$ is the sprung mass versus

unsprung mass ratio; $\omega_1 = sqrt(k_{us}/m_{us})$ is the natural frequency of primary suspension, which is sometimes called wheel hop natural frequency; it is typically between 8 and 12 Hz for most automotive vehicles.

The road disturbance, w, represents road roughness velocity which can be approximated as a white-noise process with a power spectrum density of W, equal to the product of road roughness factor A and vehicle velocity V (Hrovat, 1997).

For the above dynamic system describing vehicle vertical motion, we can minimize the following performance index, expressed as expected value $E[\cdot]$, since road roughness is a stochastic variable.

$$J = E[r_1 x_1^2 + r_2 x_3^2 + u^2],$$

where r_1 and r_2 are weighting parameters penalizing excessive tire deflection (x_1) and suspension rattlespace (x_3), respectively. By minimizing x_1 excursions, we are providing for good contact with the road, which, in turn, leads to improved vehicle handling. On the other hand, by minimizing the rattlespace, we are avoiding excessive motion between the vehicle sprung and unsprung mass, which can lead to suspension bottoming and unpleasant and potentially structurally damaging shocks. The third component of the performance index amounts to minimizing mean square sprung mass acceleration, which is related to improved ride comfort. Thus, by varying r_1 and r_2, one can obtain the best possible combinations between ride comfort and vehicle-handling subject to vehicle design constraint reflected through available rattlespace.

The global maps of all possible combination between ride and handling, and design constraint, are illustrated in Figures 3.8 and 3.9. Figure 3.8 shows the optimal possible RMS sprung mass acceleration versus RMS suspension stroke where both quantities have been normalized with respect to the road roughness parameter W.

Similarly, Figure 3.9 shows the optimal relation between normalized RMS sprung mass acceleration and normalized RMS tire deflection. The shaded area in both figures indicates the most useful operating points applicable for most driving/road conditions. From these figures, it can be seen that by varying

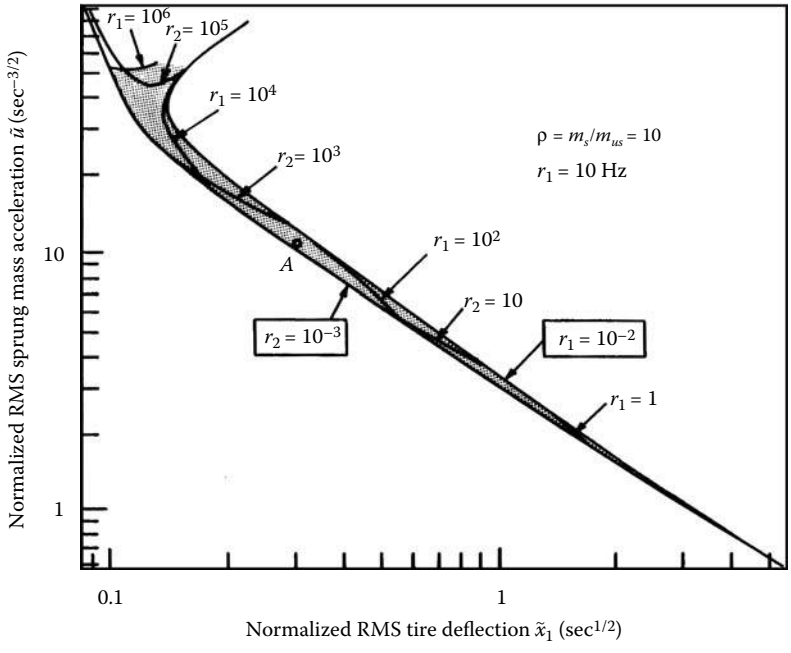

FIGURE 3.9 Sprung mass acceleration versus tire deflection diagram.

tuning parameters r_1 and r_2, we can adapt to different road conditions and driving style. For example, by reducing the tire deflection weight r_1, we are reducing the sprung mass acceleration and creating smoother ride, while at the same time, we are increasing tire deflection, which corresponds to handling deterioration. This particular compromise may be acceptable for traveling on a long straightaway with few curves. On the other hand, if we are traveling on a winding mountainous road, we may want to choose a different set of r_1 and r_2 weights that would severely penalize excessive tire deflections so that we will have maximum tire grip and road holding in this situation. This adaptive capability of active suspensions is one of their key advantages with respect to conventional passive suspensions (Hrovat, 1997) so that in any given situation, with an active suspension set, we can have the optimal combination of ride and handling parameters subject to vehicle design and road constraints.

Additional factors that should be taken into consideration during suspension design include response to various load conditions either due to the occupants' weight or additional inertia forces during corning/braking, or external forces such as wind gust. To this end, usage of load leveling or so-called fast load leveling based on LQI design with an additional integrator that secures zero steady-state error in vehicle posture can be very effective (Hrovat, 1997). It should be pointed out that the LQ approach also leads to the optimal suspension structures. This additional benefit is very important from a design and overall conceptual standpoint.

For example, applications of LQ methodology and related state space representation lead to the concept of the so-called "sky-hook" damper. Unlike a conventional damper or shock absorber, which are connected in between vehicle sprung and unsprung masses, the sky-hook damper is attached between the sprung mass and an "inertial" sky-hook reference point. This facilitates much more effective damping of the sprung mass vibration without simultaneously producing ground or road-roughness-induced vibrations, as is the case with conventional dampers. More specifically, the sky-hook damping action then makes possible a significant increase in sprung mass vibration mode damping ratio, which can be close to a value of 0.7, as opposed to the value of 0.2–0.3 that is typical of conventional suspensions. In other words, the LQ optimal system's closed-loop poles are better positioned than the corresponding passive suspension counterparts.

Other examples of structural improvement with active suspensions based on LQ optimal approach can be found in Hrovat (1997). This includes the cases where the "cheap" optimal controls have been used to arrive at the best possible quarter car suspension configuration. Further improvement in active suspension performance is possible by introducing a preview of the road profile ahead of the vehicle (Hrovat, 1997). The combination of all of these factors along with additional sensors, state estimators, consideration of nonlinear effects, and actuator design, and global positioning system (GPS) information can lead to substantial improvement in vehicle ride, handling, and overall active safety, along with many exciting new functionalities.

3.4 Electronic Stability Control

3.4.1 Introduction

ESC has gained substantial popularity and recognition since the late 1990s. In recent years, the wider proliferation of ESC across the vehicle fleet has allowed evaluation of its effectiveness in real-world crashes in various countries. According to a 2004 NHTSA (National Highway Traffic Safety Administration) report (Dang, 2007) and recent literature review (Ferguson, 2007), ESC is highly effective in reducing single-vehicle crashes in cars and sport utility vehicles (SUVs). Fatal single-vehicle crashes involving cars are reduced by 30–50%, fatal single-vehicle crashes involving SUVs are reduced by 50–70%, and fatal tripped rollover crashes are reduced by 70–90% (Dang, 2007; Ferguson, 2007).

The Society of Automotive Engineers (SAE) defines ESC as a system that has all of the following attributes (Vehicle Dynamics Standards Committee, 2004): ESC augments vehicle directional stability by applying and adjusting the vehicle brakes individually to induce correcting yaw torques to the vehicle;

ESC is a computer-controlled system that uses a closed-loop algorithm to limit understeer and oversteer of the vehicle when appropriate; ESC is able to estimate vehicle yaw rate and sideslip, and monitor driver steering input; ESC is expected to be operational over the full speed range of the vehicle.

ESC systems help drivers to maintain good control and lateral stability regardless of road conditions through a wide variety of maneuvers. Beyond yaw and lateral control, brake controls in ESC systems have recently been pursued to mitigate untripped rollovers that occur when driving on road. For example, Palkovics (1999) describes an enhanced system over yaw stability control (YSC) systems for commercial trucks. A rollover control function (RCF) is discussed as an enhancement to ESC (Lu et al., 2007a). Ford Motor Company has developed Roll Stability Control^{TM} (RSC) by adding a roll rate sensor together with additional sensing and control algorithms to ESC system (Lu et al., 2007b).

The control design tasks encountered for the development of ESC and its various enhancements (e.g., RSC) include, but are not limited to, driver intent recognition, vehicle lateral state estimation, vehicle roll estimation, and the corresponding yaw and roll stability control. In the following, our focus is on these tasks engaged in the development of a typical ESC and RSC system.

The following is organized sequentially to include driver intent recognition, vehicle lateral state estimation, vehicle roll estimation, enhanced roll angle estimation, yaw and roll stability control.

3.4.2 Driver Intent Recognition

Since ESC has the ability to affect a vehicle's attitude and motion, a function normally reserved for the driver, it needs to interpret the driver intent in order to provide proper directional control as a driver aid within the physical limitation of the moving vehicle. The angular position of the steering wheel relative to straight driving condition provides this essential information for ESC systems and is measured through a steering wheel angle sensor.

Steering wheel angle sensors can be categorized into absolute position sensors and relative position sensors. Absolute position sensors measure the absolute difference between current steering position and a fixed reference using hardware indexing. Relative position sensors measure only the steering travel relative to its power-on position and rely on software to find out the power-on position. Both types of sensors have to learn the true center, the steering wheel position of a straight-driving vehicle, due to possible hardware variations and wheel alignment changes.

Based on a single track model (see Figure 3.10), a nominal target yaw rate ω_{z-tgt} can be calculated as a function of the road wheel steering angle δ and the vehicle longitudinal speed v_x:

$$\omega_{z-tgt} = \frac{v_x}{(b_f + b_r) + k_{us} v_x^2 / g} \delta, \tag{3.7}$$

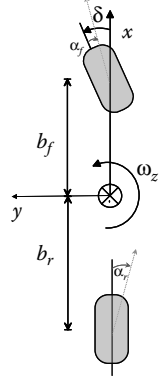

FIGURE 3.10 Bicycle model.

where $(b_f + b_r)$ represents the vehicle wheelbase, in which b_f represents the front axle to vehicle center-of-gravity (CG) and b_r represents the rear axle to CG, g is the gravity constant, and k_{us} is the steering characteristic of the passive vehicle. This nominal target yaw rate represents the typical vehicle behavior on a high friction road surface that drivers are usually accustomed to.

Depending on the road surface friction and the corresponding physical limit, this nominal value of driver desired/target yaw rate, ω_{z-tgt}, may not be achievable. Unfortunately, no sensor is available to directly measure the limit of adhesion between the tire and the road. ESC control strategies from different vendors vary on how to best modify the target yaw rate to an achievable target that ensures a stable vehicle maneuver without diminishing the driver's steering command within the road adhesion limit.

One approach (van Zanten, 2000) is to modify and limit the target yaw rate through the utilized level of road–tire friction reflected from current lateral acceleration measurement, a_y. Define a yaw rate target limit as

$$\omega_{z-\lim} = \frac{a_y}{v_x}. \tag{3.8}$$

This provides a pragmatic limit to the yaw rate that is achievable at the current tire/road condition without generating excessive sideslip. Based on a filtered version of the difference between ω_{z-tgt} and $\omega_{z-\lim}$, a delta value, $\Delta\omega_{z-tgt}$, can be calculated and subtracted from the nominal target yaw rate to generate a modified target yaw rate appropriate for ESC (Tseng et al., 1999). The modification is given by

$$\omega_{z-tgt,\,\text{modified}} = \left(\left|\omega_{z-tgt}\right| - \left|\Delta\omega_{z-tgt}\right|\right) \cdot \text{sign}[\omega_{z-tgt}], \tag{3.9}$$

where $\Delta\omega_{z-tgt}$ is derived from the following:
if $|\omega_{z-tgt}| \leq |\omega_{z-\lim}|$,

$$\Delta\omega_{z-tgt} = 0;$$

else

$$\frac{d}{dt}\Delta\omega_{z-tgt} = \frac{1}{\tau}(|\omega_{z-tgt}| - |\omega_{z-\lim}|) - \Delta\omega_{z-tgt},$$

where τ is a design parameter.

With the above process, the excessive target yaw rate is low-pass filtered and subtracted from the nominal target yaw rate. Thus, the modified target yaw rate, $\omega_{z-tgt,\,\text{modified}}$, allows a smooth transition to the achievable yaw rate under various road surfaces while maintaining responsiveness to the driver's steering wheel command.

One may also try to estimate the available friction and feed the information directly into the yaw rate target generation (Fennel and Ding, 2000). An available road friction estimation can be based on the difference between the linear yaw rate and linear lateral acceleration and these measured values (Kim et al., 2003), or it can be based on an extended vehicle state estimation (Lakehal-ayat et al., 2006).

3.4.3 Vehicle Lateral State Estimation

Vehicle lateral velocity information, or the corresponding vehicle/tire sideslip angles, can indicate the dynamic state of a vehicle related to its lateral stability and can be used as a control variable by ESC and RSC systems.

In the literature, a large number of model-based observers for lateral velocity combine a vehicle model with tire models that either have an effective/linear cornering stiffness or have nonlinear characteristics. The effective cornering stiffness is defined as the linear gain between tire sideslip angle and tire force. With this approach, the corresponding vehicle system model can be approximated as a linear model where tire forces are represented as the product of the tire slip angle and an effective cornering stiffness

(Ungoren et al., 2004; Sierra et al., 2006), and a variety of modern linear observer theories can be applied. Another approach with the nonlinear tire models offers an opportunity to provide a better correlation with the actual tire forces under a nominal road surface/condition. However, the specific, and usually more complex, tire function may not be valid under other road surface/conditions, and additional switching logic may be required (Fukuda, 1998; Tseng et al., 1999; Hac and Simpson, 2000; van Zanten, 2000; Nishio et al., 2001).

The approach with effective cornering stiffness and linear model can utilize a wide variety of modern control theories for analytical proof of estimation convergence (Kaminaga and Naito, 1998; Liu and Peng, 1998). However, the effective cornering stiffness, as defined, would need to be adapted to reflect tire nonlinearity and road surface changes. The performance of this approach, which requires parameter adaptation, thus depends on the rate of convergence in its parameter adaptations.

An example of the effective cornering stiffness model is given by

$$\begin{bmatrix} \dot{v}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} -\dfrac{c_f + c_r}{M_t v_x} & -v_x + \dfrac{-b_f c_f + b_r c_r}{M_t v_x} \\ \dfrac{-b_f c_f + b_r c_r}{I_z v_x} & \dfrac{-b_f^2 c_f - b_r^2 c_r}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} \dfrac{c_f}{M_t} \\ \dfrac{b_f c_f}{I_z} \end{bmatrix} \delta, \tag{3.10}$$

where c_f and c_r are the unknown front and rear tire effective cornering stiffnesses, respectively, M_t is the vehicle mass, I_z is the vehicle inertia with respect to yaw axis, ω_z, is the yaw rate, and v_y denotes the lateral velocity of the CG of the vehicle.

A state-space approach that estimates both parameters and state variables concurrently has been presented in Liu and Peng (1998) where the steady-state analytical convergence is guaranteed for both the states and parameters. A further experimental study shows that the speed of convergence may need further improvement for practical applications (Ungoren et al., 2004).

Another approach utilizes a Lyapunov observer for state convergence and uses a sliding mode observer (SMO) to expedite the parameter convergence (Tseng, 2002). This SMO approach is summarized below.

Consider a linear varying plant model as

$$\dot{x} = Ax + Bu, \quad y = Cx, \tag{3.11}$$

where $x = \begin{bmatrix} v_y \\ \omega_z \end{bmatrix}$, $u = \delta$,

$$A = \begin{bmatrix} -\dfrac{c_f + c_r}{M_t v_x} & -v_x + \dfrac{-b_f c_f + b_r c_r}{M_t v_x} \\ \dfrac{-b_f c_f + b_r c_r}{I_z v_x} & \dfrac{-b_f^2 c_f - b_r^2 c_r}{I_z v_x} \end{bmatrix}, \quad B = \begin{bmatrix} \dfrac{c_f}{M_t} \\ \dfrac{b_f c_f}{I_z} \end{bmatrix},$$

$$C = [0 \quad 1].$$

The combined observer structure is then derived with a Lyapunov function $V = \tilde{x} P \tilde{x} + \tilde{\theta}^T \Gamma \tilde{\theta}$,

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u + L(y - C\hat{x}) \\ &= W(\hat{x}, u)\hat{\theta} + L(y - C\hat{x}), \end{aligned} \tag{3.12}$$

$$\dot{\hat{\theta}} = \Gamma^{-1} W^T(\hat{x}, u) P \operatorname{sign}(\tilde{x}'), \tag{3.13}$$

where $\theta = [c_f \quad c_r]^T$, L is the observer gain design parameter, and P and Γ are the Lyapunov function design parameters.

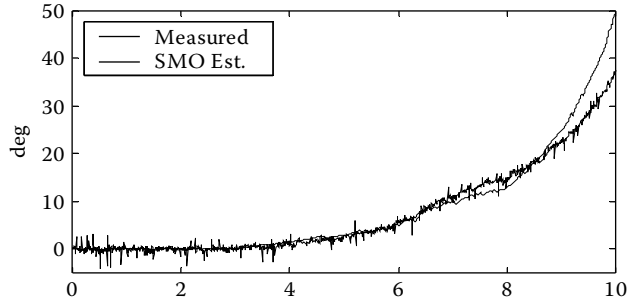

FIGURE 3.11 Sideslip angle at the rear axle of a slow drift maneuver on snow.

Note that $\tilde{x}' = \begin{bmatrix} \tilde{v}'_y \\ \tilde{\omega}_z \end{bmatrix} = \begin{bmatrix} v'_y - \widehat{v}_y \\ \omega_z - \widehat{\omega}_z \end{bmatrix}$, and v'_y, which was called the "virtual measurement" in Tseng (2002), is derived from

$$\dot{\tilde{v}}'_y = -\lambda \tilde{v}'_y + (\dot{v}_y - \dot{\hat{v}}_y), \tag{3.14}$$

where $\dot{v}_y$ is a measurement ($\dot{v}_y = a_y - v_x \cdot \omega_z$), $\tilde{v}'_y$ is the virtual lateral velocity measurement error, and λ is a design parameter.

The sideslip angle at the rear axle can be computed based on the aforementioned estimation of v_y:

$$\beta_{ESC-ra} = \tan^{-1} \frac{v_y - b_r \omega_z}{v_x}.$$

The experimental results of this SMO for several maneuvers that are generally considered difficult to estimate are discussed in Tseng (2002) and examined below. In Figures 3.11 through 3.14, the estimated β_{ESC-ra} and the measured one are compared. Note that the rear tire sideslip angle β_{ESC-ra} is a more transparent indicator of vehicle limit handling than lateral velocity itself or the sideslip angle at the CG of the vehicle.

Figure 3.11 shows β_{ESC-ra} estimate performance during a slow drift maneuver on snow. Figures 3.12 and 3.13 show its performance during various narrow- and wide-lane change maneuvers on snow. Both tracked the instrumentation measurement well in magnitude and phase.

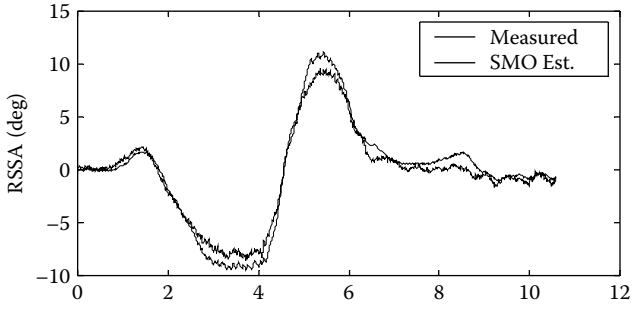

FIGURE 3.12 Sideslip angle at the rear axle of wide lane change maneuvers on snow.

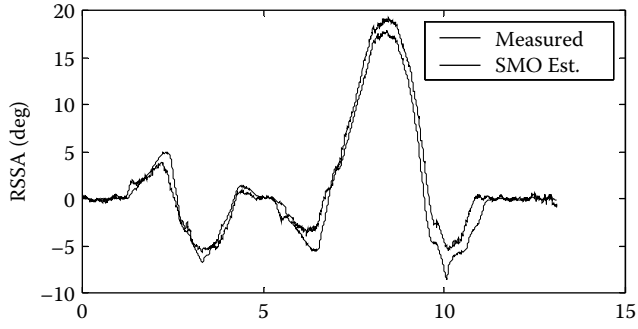

FIGURE 3.13 Sideslip angle at the rear axle of various lane changes on snow.

Note that during maneuvers on banked roads, the time derivative of lateral velocity is subject to the vehicle roll angle, φ_x, in addition to measured lateral accelerometer, a_y, yaw rate, ω_z, and vehicle speed, v_x. That is, the measured lateral velocity derivative contains vehicle roll disturbances,

$$\dot{v}_y = a_y - v_x \omega_z - g \sin \varphi_x. \tag{3.15}$$

Since the vehicle roll disturbances are inevitable and sometimes substantial, it is very desirable if vehicle roll angle can be correctly estimated.

3.4.4 Roll Angle Estimation with ESC Sensor Set

In the literature, Fukada (Fukada, 1998, 1999) first addressed vehicle roll angle estimate and its influence on vehicle lateral state estimate. He derived and compared the lateral tire force from the lateral acceleration measurement and from a tire model to obtain a roll angle estimate. In this approach, vehicle sideslip angle and vehicle roll angle estimates affect each other, and the observer stability is difficult to prove. Nishio et al. (2001) used a low-pass filtered version of the difference between lateral acceleration measurement and the product of yaw rate and vehicle longitudinal speed as a roll angle estimate but did not address the issue of a sometimes significant measurement disturbance. In particular, the difference described in his approach contains more than vehicle roll angle information during dynamic maneuvers. A roll angle estimate that is independent of sideslip angle estimation yet limiting the measurement disturbance is first proposed in Tseng (2000, 2001). This proposed calculation of road bank estimate is later simplified (Tseng and Xu, 2003; Xu and Tseng, 2007), which will be summarized below.

As described in the previous section, to know the correct value of lateral velocity derivative from sensor measurements, one would need to know the vehicle roll information; see Equation 3.15.

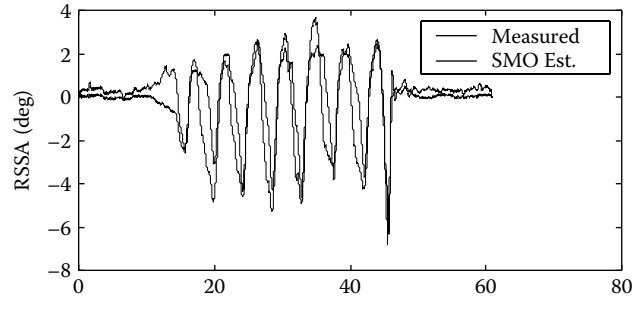

FIGURE 3.14 Sideslip angle at the rear axle estimate during banked slalom maneuvers.

Similarly, based on the same kinematic equation, to know the vehicle roll angle one would have to know lateral velocity derivative information, as

$$\varphi_x = \sin^{-1}[(a_y - v_x\omega_z)/g - \dot{v}_y/g]. \tag{3.16}$$

A raw vehicle roll angle estimate is first calculated:

$$\hat{\varphi}_{x,raw} = \sin^{-1}[(a_y - v_x\omega_z)/g], \tag{3.17}$$

where $\dot{v}_y$, the component that cannot be directly measured, is ignored. To account for this unknown dynamic disturbance and its influences on the raw vehicle roll angle estimate, a pragmatic approach is to modulate this raw estimate to attain a better estimate. As described in Tseng (2001), a refined vehicle roll can be achieved by the introduction of a dynamic factor, denoted as DFC below, to reflect the magnitude of the unknown vehicle dynamic disturbance $\dot{v}_y$.

$$DFC = \frac{2v_x^2}{[(b_f + b_r) + k_{us}v_x^2]g}\left[k_{us}a_y + \frac{\omega_z}{v_x}(b_f + b_r) - \delta\right], \tag{3.18}$$

$$\hat{\varphi}_{x,refined} = \sin^{-1}\left(\sin\hat{\varphi}_{x,raw} \cdot \max[0, \quad 1 - |DFC|]\right). \tag{3.19}$$

Note that DFC is zero during steady-state cornering if the bicycle model and its nominal understeer coefficient are accurate (Tseng, 2001). While the actual understeer coefficient of a vehicle may deviate from its nominal value, the effect of these variations on DFC can be minimized by choosing the nominal understeer coefficient based on the high friction surface behavior of the vehicle. With this choice of nominal k_{us}, the deviation of the product $k_{us}a_y$ from its true value is limited since a_y would be limited on low friction road surfaces, which in turn allows a value that is in practice robust to different road surfaces.

Therefore, with the modulation (Equation 3.19), the refined roll angle estimate can fully reflect steady-state vehicle roll angle while becoming progressively more conservative (i.e., the estimated value gets closer to zero) as the change rate of lateral velocity and the corresponding DFC increase.

Figure 3.14 demonstrates, through experimental data, that by feeding the refined roll angle estimate suggested here to the SMO described in the previous section, the lateral velocity estimate can be robust to road bank disturbances. It shows that the β_{ESC-ra} estimate is close to a GPS measurement during slalom maneuvers on a high friction banked road where both vehicle roll and road bank angle are dynamically changing.

It should be pointed out that the modulation logic suggests a vehicle usually does not experience excessive lateral velocity change when it is driven on roads with large bank angles, and vice versa. As such, this approach refines the vehicle roll angle estimate and can eliminate most of the lateral vehicle dynamic disturbance, for most but not for all situations. A roll rate sensor, together with the information from the ESC system, can help provide more robust vehicle roll angle estimation. This is discussed in the next section.

3.4.5 Enhanced Roll Angle Estimation

Vehicle roll angles of interest include both global and relative roll angles. The roll angle of a vehicle body with respect to sea level is called a *global roll angle*, φ_x, which is complementary to the angle between the earth gravity vector (perpendicular to sea level with a magnitude of $g = 9.81$ m/s^2) and the lateral direction of the vehicle body. This connection makes φ_x useful in compensating the lateral acceleration measurement. The earth gravity component contained in the lateral acceleration measurement can be removed such that the derivative of the lateral velocity of a vehicle can be extracted. As a result, the lateral velocity (thus sideslip angle) can be obtained from the lateral acceleration measurement. The lateral velocity of a vehicle can also be directly measured by using, for example, optical sensors. Optical sensors cost much more than accelerometers and have poor measurement performance for certain road conditions (e.g., snow surfaces) which make them impractical for mass production implementation.

Another angle of interests is the *relative roll angle*, φ_{xbm}, measuring the vehicle body's roll angle with respect to the average road surface. Generally speaking, the roll stability of a vehicle can be directly inferred from φ_{xbm} but not necessarily inferred from φ_x. Only when the road surface is the same as sea level does φ_{xbm} coincide with φ_x. Relative roll angle, φ_{xbm}, can also be directly determined, for example, using multiple laser sensors mounted on the vehicle body that measure the distances between the sensor mounting locations and the average road surface. However, the laser sensors are cost prohibitive for mass production implementation.

Practical approaches for determining roll angles and the other vehicle states are through estimation algorithms using inputs from inertial sensors such as accelerometers and angular rates. The most widely used inertial sensor configuration is the so-called *ESC sensor set* including a longitudinal accelerometer, a lateral accelerometer, a yaw rate, a steering wheel angle, four wheel speeds, and a master cylinder pressure sensor. The first three inertial sensors are usually packed into a cluster called an *ESC motion sensor cluster*. Another configuration adds a roll rate sensor to the ESC sensor set. Namely, a roll rate sensor is added to the ESC motion sensor cluster. Such an incremental sensor set is used in RSC (Brown and Rhode, 2001; Lu et al., 2007b), which is called the *RSC sensor set*. The 4 sensor element cluster incremented from the ESC motion sensor cluster is called the *RSC motion sensor cluster*.

The global roll angle, φ_x, can be related to a roll Euler angle of the vehicle body. Euler angles (Greenwood, 1998) of the vehicle body determine the angular positions of the vehicle body with respect to sea level and they can be mathematically characterized by various types. They are defined through various sequences of simple rotations around the body-fixed axes. Euler angles have the ability to uniquely capture the vehicle body's motions in the three-dimensional space and they can be related to the angular rate sensor measurements. Generally speaking, φ_x cannot be simply estimated through integrating the roll rate signal alone due to it being coupled with a pitch Euler angle φ_y and the angular rate measurements from the roll, pitch, and yaw angular rate sensors mounted on the vehicle body. More specifically, φ_x and φ_y obey the following kinematics:

$$\begin{aligned} \dot{\varphi}_x &= \omega_x + \omega_y \sin(\varphi_x)\tan(\varphi_y) + \omega_z \cos(\varphi_x)\tan(\varphi_y), \\ \dot{\varphi}_y &= \omega_y \cos(\varphi_x) - \omega_z \sin(\varphi_x), \end{aligned} \tag{3.20}$$

where ω_x, ω_y, and ω_z measure rotations along the body-fixed longitudinal, lateral, and vertical axes. If the sensor measurements are well calibrated and compensated for uncertainties, the direct integration of the coupled ordinary differential equations 3.20 will lead to a desired estimation of φ_x and φ_y.

Angles φ_x and φ_y can also be determined from accelerations through earth gravity components as in the following:

$$\begin{aligned} \varphi_y &= \sin^{-1}\left\{\frac{\dot{v}_x - a_x - \omega_z v_y}{g}\right\}, \\ \varphi_x &= \sin^{-1}\left\{\frac{a_y - \omega_z v_x - \dot{v}_y}{g\cos(\varphi_y)}\right\}, \end{aligned} \tag{3.21}$$

where a_x and a_y are the longitudinal and lateral accelerations of the vehicle body, v_x and v_y are the longitudinal and lateral velocities.

Not all the signals in Equation 3.20 or 3.21 are directly measured through sensors. In the ESC case, only three motion variables a_x, a_y, and ω_z are measured together with the longitudinal velocity v_x computed from the other sensor signals such as wheel speeds; while in the RSC case, an additional roll rate ω_x is used. Combining Equation 3.20 with Equation 3.21 leads to hybrid ordinary differential equations (HODEs) with regular ordinary differential equations (ODEs) (3.20) plus algebra constraints (3.21). Since we have four equations with four unknowns φ_x, φ_y, ω_y, and v_y, solving those unknowns from the HODEs might be a well-posed problem.

Before solving the HODEs, we need to answer if they have a unique solution when the only known signals are the sensor measurements and the computation v_x. Since the state estimation using the ESC

sensor set has been discussed in the previous section, our focus here is on vehicle state estimation using the RSC sensor set.

By choosing v_y as an unknown and manipulating Equations 3.20 and 3.21 through small-angle approximations, the HODEs are equivalent to the following single ordinary differential equation with a single unknown v_y

$$\ddot{v}_y - \omega_z^2 v_y = g(\omega_x + \omega_z \Psi_y - \dot{\Psi}_x), \tag{3.22}$$

where

$$\Psi_y = (\dot{v}_x - a_x)/g, \quad \Psi_x = (a_y - \omega_z v_x)/g.$$

There is a unique solution v_y for Equation 3.22 for a given initial condition for v_y. Hence, the HODEs must have a unique solution. This can be easily proved by plugging a solution v_y obtained from Equation 3.22 into the algebraic equations 3.21 to obtain the Euler angles φ_x and φ_y, and the pitch rate ω_y. That is, Equation 3.22 shows the existence of a unique solution for $\varphi_x, \varphi_y, \omega_y$, and v_y from the HODEs if ω_x, ω_z, a_x, a_y, and v_x are given.

In reality, the sensor element signals have various errors caused by

- Sensor biases (including slow time-varying components due to temperature variation).
- Sensor scaling factor error.
- Misalignment of sensor axes from harmonization and nonorthogonality.

Hence, sensor signals need to be compensated. For instance, the roll and yaw rate sensor bias might be found when a vehicle is parked; the accelerometer bias might be compensated based on certain driving conditions under certain assumptions. Not all the above sensor errors can be compensated without using extra information (e.g., a GPS receiver). The challenge here is how to compensate the sensor signals as much as possible such that the error in vehicle state estimation can be minimized.

Besides the errors inside the sensor cluster, sensor mounting can also introduce error. The sensor cluster has its own coordinate system (called a *sensor frame*) that does not necessarily align perfectly with the coordinate system of the vehicle body (called a *body frame*). Figure 3.15 shows the rear view of a vehicle driven on a banked road with sensor frame S consisting of longitudinal, lateral, and vertical axis s_1, s_2, and s_3 (only s_2 and s_3 are depicted), while the vehicle body has body frame B with longitudinal, lateral, and vertical axis b_1, b_2, and b_3 (only b_2 and b_3 are depicted). A frame M attached to the road surface underneath the vehicle but traveling and yawing with the vehicle is called a *moving road frame* with lateral and vertical axes m_2 and m_3 as shown in Figure 3.15. Let us denote φ_{xbw} as the relative roll angle between body frame B and the axle of either the front or the rear wheels and φ_{xbm} as the relative roll angle between body frame B and moving road frame M.

In order to differentiate the signals defined on sensor frame S from the general signals used previously, a subscript "s" is added to each variable. That is, the roll and yaw rate sensor measurements are now denoted as ω_{xs} and ω_{zs}; a_{xs} and a_{ys} denote the longitudinal and lateral acceleration measurements of the origin of the RSC motion sensor cluster but defined on sensor frame S; v_{xs} and v_{ys} denote the velocities of the origin of the RSC motion sensor cluster defined on sensor frame S.

The total misalignments between the sensor units and body frame B are denoted as Δ_x, Δ_y, and Δ_z for the roll, pitch, and yaw misalignments. Under certain driving conditions, Δ_x, Δ_y, and Δ_z can be estimated from the raw sensor signals. Let σ_{xs}, σ_{ys}, and σ_{zs} be the three components of the sensor units (either angular rates or accelerations) and σ_{xb}, σ_{yb}, and σ_{zb} be their projections on to body frame B, we then have the following transformation:

$$\begin{aligned}
\sigma_{xb} &= c_z c_y \sigma_{xs} - (s_z c_x - c_z s_y s_x)\sigma_{ys} + (s_z s_x + c_z s_y c_x)\sigma_{zs}, \\
\sigma_{yb} &= s_z c_y \sigma_{xs} + (c_z c_x + s_z s_y s_x)\sigma_{ys} + (c_z c_x + s_z s_y c_x)\sigma_{zs}, \\
\sigma_{zb} &= -s_z \sigma_{xs} + c_y s_x \sigma_{ys} + c_y c_x \sigma_{zs},
\end{aligned} \tag{3.23}$$

where $s_x = \sin\Delta_x$, $c_x = \cos\Delta_x$, $s_y = \sin\Delta_y$, $c_y = \cos\Delta_y$, $s_z = \sin\Delta_z$, and $c_z = \cos\Delta_z$.

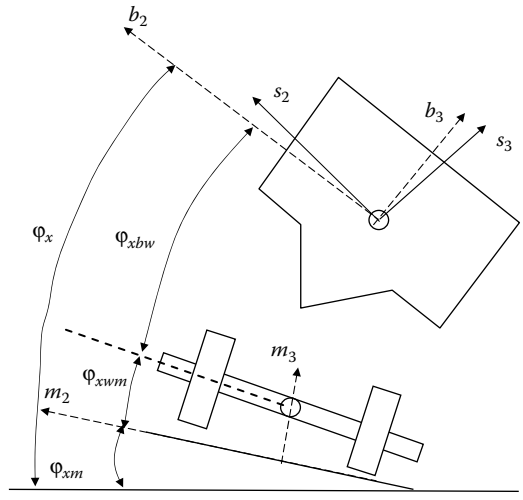

FIGURE 3.15 The roll angle definitions for a vehicle driven on a banked road (rear view).

Let us first consider computing φ_{xbw} which is also called the *chassis roll angle* whose main contributor is the suspension deflections. Let F_{yf} and F_{yr} be the resultant forces along the lateral direction of the RSC motion sensor cluster but applied to the vehicle body through the front and the rear roll center, respectively. Let h_f and h_r be the vertical distance from the vehicle body CG to the front and the rear roll center, respectively. Let l_{s2cg} be the longitudinal distance between the origin of the RSC motion sensor cluster and vehicle body CG. Using Newton's law in sensor frame S leads to the following equations of motions:

$$\begin{aligned} M_t(a_{ys} + l_{s2cg}\dot{\omega}_{zs}) &= F_{yf} + F_{yr}, \\ I_z\dot{\omega}_{zs} &= F_{yf}b_f - F_{yr}b_r, \\ I_x\dot{\omega}_{xs} &= F_{yf}h_f + F_{yr}h_r - K_{roll}\varphi_{xbw} - D_{roll}\varphi_{xbw}, \end{aligned} \tag{3.24}$$

where I_x is the moment of inertia of the vehicle body with respect to its longitudinal body axis; K_{roll} and D_{roll} are the equivalent roll stiffness and damping rate of the suspension. Based on Equation 3.24 and using the Laplace transformation, φ_{xbw} can be solved as in the following:

$$\varphi_{xbw} = T_1(s)a_{ycgs} + T_2(s)\omega_{sx} + T_3(s)\omega_{zs}, \tag{3.25}$$

where $T_1(s)$, $T_2(s)$ and $T_3(s)$ are three transfer functions which can be easily obtained from Equation 3.24, and

$$a_{ycgs} = a_{ys} + l_{s2cg}\dot{\omega}_{zs} \tag{3.26}$$

is the lateral acceleration at the vehicle body CG but projected along the lateral direction of sensor frame S.

Thus, calculated φ_{xbw} is based on a linear model with a fixed body roll axis. It is likely to deviate from the true value if the vehicle has wheel lift or is operated at the nonlinear operation region of its suspension. It might also be sensitive to the variation of vehicle loading, CG height, or roll moment of inertia. However, if there is no wheel-lift, φ_{xbw} coincides with φ_{xbm} for the given vehicle parameters. Hence a small magnitude of φ_{xbw} is sufficient to infer a roll-stable situation.

A roll-instable condition or a rollover is defined as a crash when referring to any vehicle with rotation above 90° around its longitudinal axis. Or equivalently, φ_{xbm} is sufficiently large. Another way of directly measuring a rollover event is the roll angle between the axle of the front or the rear wheels and frame M, which is called a wheel departure angle denoted as φ_{xwm} in Figure 3.15.

Through the aforementioned φ_x and φ_{xbw}, the wheel departure angle φ_{xwm} is not necessarily observable since it cannot be separated from the road bank angle φ_{xm} shown in Figure 3.15 due to the following relationship:

$$\varphi_x - \varphi_{xbw} = \varphi_{xwm} + \varphi_{xm}. \tag{3.27}$$

One way to determine φ_{xwm} so as to infer a roll-instable condition is to identify when the roll-instable condition starts to happen. Such a qualitative determination together with Equation 3.27 might be used to jointly infer φ_{xwm} with the assumption that the road bank variation during a rollover event is negligible in comparison with the wheel departure angle φ_{xwm} itself. Namely, the road bank φ_{xm} is assumed to be almost constant, by clipping its value to the one right before the rollover condition, throughout the rollover event. Therefore, the estimation of φ_{xwm} would need a characterization of a potential rollover event through independent sources other than roll angle computed from the inertial sensor measurements.

The occurrence of an on-road rollover can be indirectly inferred from the driver's steering input, the vehicle's cornering acceleration, and the wheel motion status. The inference of a rollover event without directly using the roll angle information, instead using dynamics capturing the conditions of wheel lifting from the ground, is called a wheel lift detection (WLD). This focuses on the qualitative nature of determining a large rolling event and it will be discussed in more detail in the sequel. A state estimation integrating the quantitative roll angles from inertia sensors with the qualitative roll sensing from the WLD algorithms is used here. Such an approach is different from many existing ones in rollover detection, where either the robust quantitative roll sensing is missing (e.g., without using a roll rate sensor) or the qualitative roll sensing alone is manipulated for rollover mitigation purposes. The advantage of the integrated approach is that it provides a roll-instability detection that is robust to sensor uncertainties, road variation, and driving condition variations. At the same time, the quantitative portion of roll sensing serves as the feedback control variable for generating a smooth but effective control command to prevent rollovers.

More specifically, WLD determines if there is an inside wheel lifted from the ground through an active wheel lift detection (AWLD) algorithm and a passive wheel lift detection (PWLD) algorithm. The integrated wheel lift detection (IWLD) algorithm arbitrates and coordinates AWLD and PWLD to generate the final indication of the wheel lift condition of the vehicle. The wheel lift status for each wheel is set to one of five discrete states assuming values of 2, 4, 8, 16, and 32 (stored in different bits on a one-byte variable) that qualitatively characterize the wheel as being *absolutely grounded, possibly grounded, no indication, possibly lifted, and absolutely lifted*, respectively.

Furthermore, AWLD determines a wheel lift condition by checking the wheel rotation in response to a test brake pressure. It requests a small amount of brake pressure to be sent to an inside wheel when the driver shows an aggressive steering input together with a sufficiently large cornering acceleration and checks the rotational response of that wheel. If a longitudinal slip ratio larger than a threshold is seen, the wheel is deemed to be likely lifted from the ground (the cornering acceleration condition screens out the low friction surface condition). Considering the wheel rotation pattern and the other vehicle states, the wheel lift is characterized as being possibly or absolutely lifted based on the detection confidence.

Due to the reactive nature to the test brake pressure, a wheel lift conclusion from AWLD algorithm might suffer some time delay. PWLD determines a wheel's wheel lift condition by checking the vehicle and wheel states without actively requesting sending a test brake pressures to that wheel. Namely, it passively monitors the wheel motion pattern together with vehicle states to determine if wheel lift is occurring.

In order to capitalize on the benefit of AWLD during steady-state driving conditions and the benefit of PWLD during dynamic maneuvers, IWLD is used. Figure 3.16 shows a conceptual description of the integration. Figure 3.17 uses a vehicle test data during a J-turn maneuver with a detuned RSC controller to illustrate various wheel lift status determined in real-time. The brake pressure due to AWLD request and the wheel speed response are also shown in Figure 3.17. More details on WLD can be found in Lu et al. (2006).

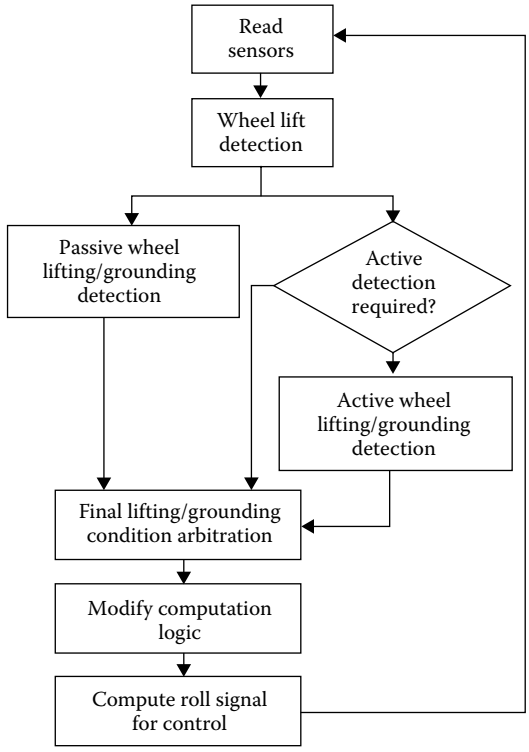

FIGURE 3.16 The integration between AWLD and PWLD.

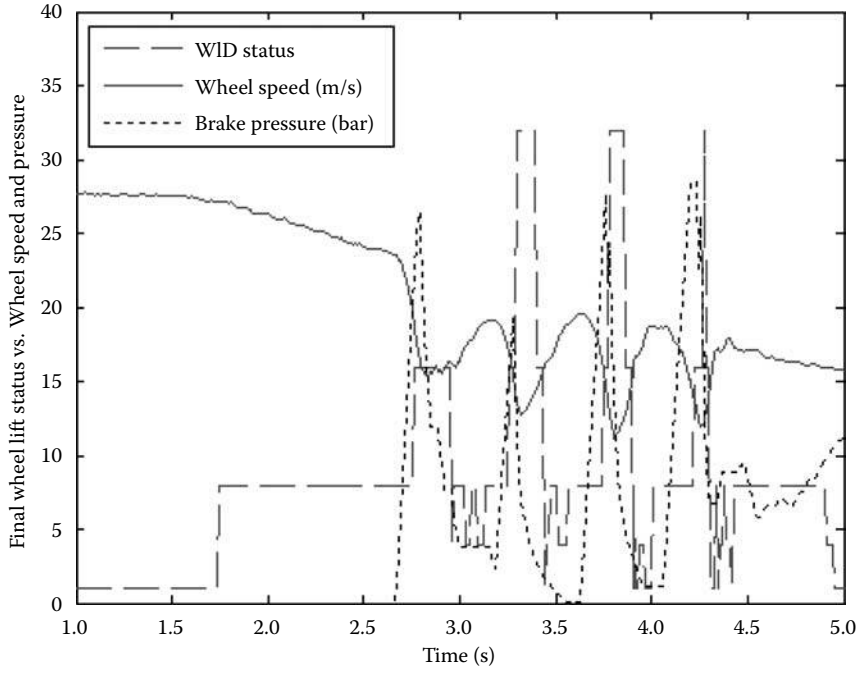

FIGURE 3.17 The WLD flag for an inside wheel during a J-turn maneuver (with detuned control).

3.4.6 Yaw Stability Control

Through individual wheel brake control, a yaw torque can be introduced to achieve lateral stability of a vehicle during turning maneuvers on slippery roads. By monitoring the vehicle yaw rate and sideslip angle, a wheel brake induced yaw torque control can effectively correct for oversteering and understeering (Pillutti et al., 1995; Hrovat and Tran, 1996).

One approach is to regulate the vehicle yaw rate to the desired yaw rate discussed in Section 3.4.2. Another approach is to trade off yaw rate regulation by limiting the vehicle sideslip angle. The advantages and limitations of each approach have been discussed in the literature (Hac, 1998). Both approaches can be combined for more predictable, progressive, and nonintrusive vehicle lateral behaviors. The details of these practical combination philosophies can be found in Tseng et al. (1999) and van Zanten (2000). For example, the target lateral response can be designed to be responsive and progressive so that it not only ensures the stability of the vehicle but also preserves the "fun-to-drive" quality. Other actuation considerations such as system transparency, actuation efficiency, and smoothness are also discussed in Tseng et al. (1999). A YSC flow diagram is illustrated in Figure 3.18.

3.4.7 Roll Stability Control

The goal of RSC is to prevent an untripped rollover from happening. That is, RSC makes a vehicle understeer some more in order to avoid an untripped rollover. The maneuvers that lead to potential rollovers can be determined by monitoring the roll trending of the vehicle during large cornering acceleration maneuvers.

The method documented in Lu et al. (2007b) is used as an example to discuss the control strategy designed for RSC. The control strategy needs not only to determine the control commands but also to (1) overcome the time delays in the brake hydraulics; (2) provide effective brake torques to counteract

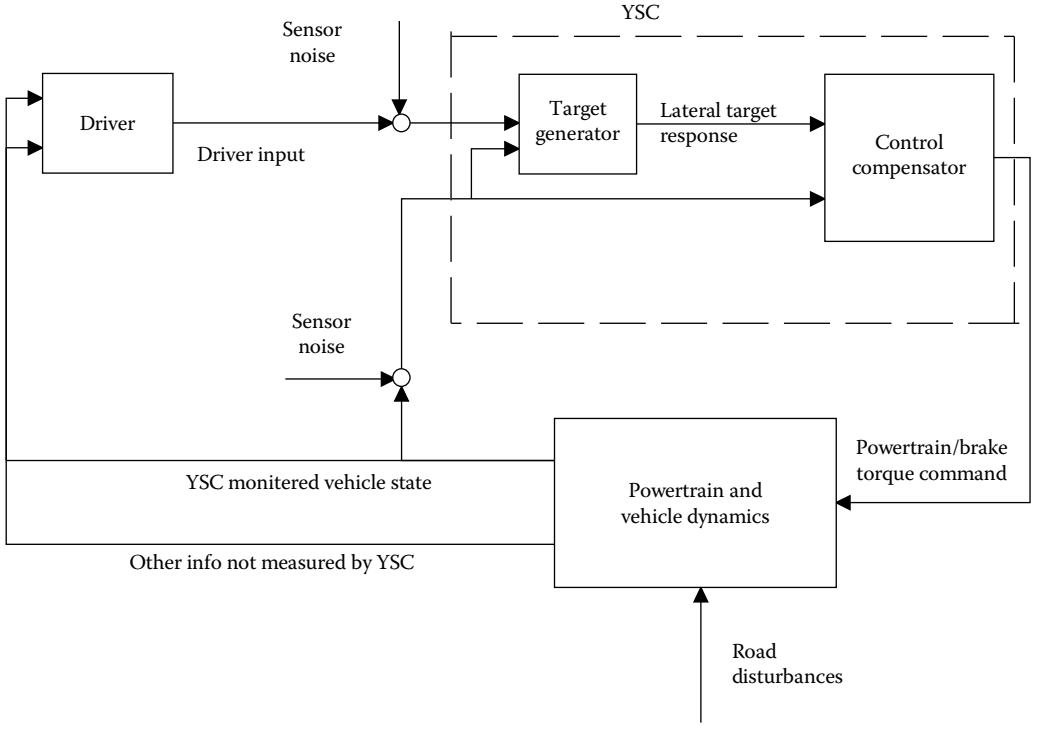

FIGURE 3.18 YSC and feedback.

the vehicle roll motion for all vehicle configurations, road conditions, and driver inputs; and (3) not generate unnecessary activations that could be intrusive, annoying, and limiting the responsiveness of the vehicle.

RSC in Lu et al. (2007b) intends to achieve those requirements. It contains a *Transition Controller* that performs the RSC for the transitional portions of dynamic maneuvers and a *Quasi-Steady-State Feedback Controller* that performs the RSC for less dynamic portion of dynamic maneuvers.

3.4.7.1 Transition Controller

The transition controller includes a feedforward control law plus three feedback control laws. Generally speaking, in order to counteract rollover, a large and rapid brake pressure buildup is requested on the front outside wheel during a large and rapid roll trending. It is likely to exceed the maximum brake pressure buildup rate. In this case, significant time delays in building up brake pressure to match the request can occur due to the limitations in the hydraulic capabilities. If a brake pressure buildup is requested after the roll instability is underway, there may not be sufficient time to control or mitigate the roll-instable event. To deal with the brake pressure buildup delay, a feedforward control is first used in the transition controller to precharge the brake hydraulics; see the brake pressure profile in "x" line in Figure 3.19. Such a feedforward control utilizes prediction based on the driver's steering input and the vehicle states to prefill the brake caliper prior to roll instability. This prefill is only designed to minimize pressure buildup delay and it requests relatively small amount of pressures to overcome inertia in the brake pump and to reduce caliper knockback. In addition to caliper prefill, pressure buildup prediction and actuator delay compensation have also been introduced. Limitations in brake hydraulics are compensated for by projecting forward when a predetermined pressure level is likely to be requested, based on φ_{xbw}, ω_x, $\dot{\omega}_x$, and the estimated caliper pressure. Brake pressure is built up such that the desired peak pressure can be achieved when it is needed to reduce the effects of pressure buildup rate limitations.

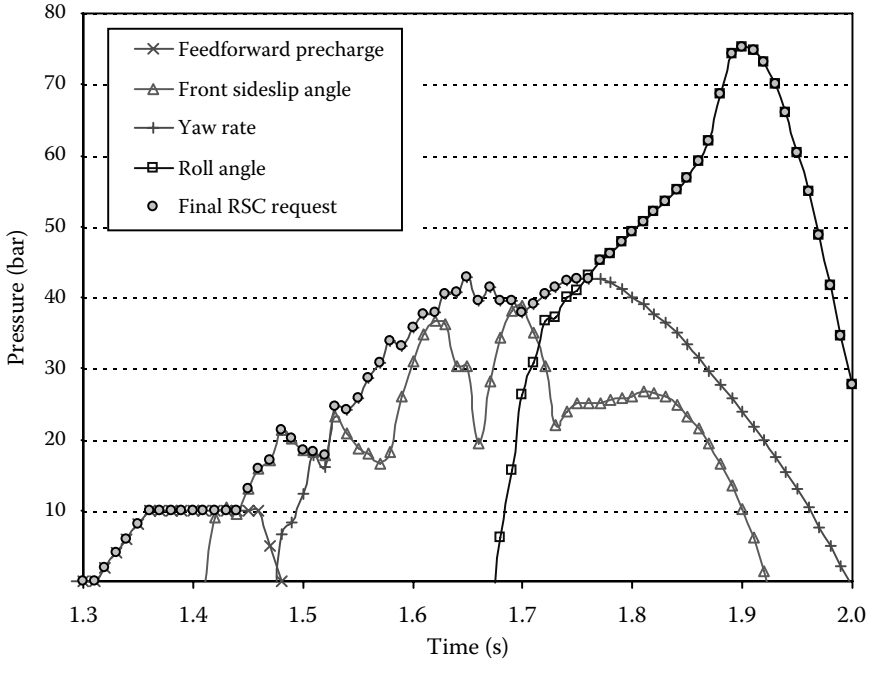

FIGURE 3.19 Pressure profile of the transition control during a fishhook maneuver.

The other control strategy in the transition controller is a feedback control consisting of three control laws. The first feedback control law uses the model-based linear sideslip angle β_{falin} at the front axle which is defined as

$$\beta_{falin} = \frac{F_{yf}}{c_f}, \tag{3.28}$$

where F_{yf} is the front cornering force that can be computed from Equation 3.24 as in the following:

$$F_{yf} = \frac{I_z \dot{\omega}_{zs} + b_r M_t a_{ycgs}}{b_f + b_r}. \tag{3.29}$$

Since β_{falin} is proportional to the actual front cornering force F_{yf}, it builds up its own dominance prior to the development of yaw rate and roll angle. The brake pressure profile based on such a β_{falin} feedback is shown in "Δ" line in Figure 3.19.

When F_{yf} develops before yaw rate peaks out, yaw rate is likely to dominate the vehicle responses. A yaw-rate-based proportion and differentiation (PD) feedback control is introduced in order to provide adequate yaw damping to the vehicle. Such a yaw-rate-based feedback control is different from the one designed for ESC in a sense that the target yaw rate here is zero regardless of the driver's steering input. It achieves the following benefits for the vehicle: (1) minimizing the occurrence of excessive yaw rate overshoot in limit maneuvers; (2) reducing the occurrence of excessive sideslip angle and lateral forces that exceed the steady-state cornering capacity of the vehicle; and (3) increasing the roll stability margin especially during aggressive maneuvers. Such a control is designed to provide as much yaw damping as possible without inhibiting the responsiveness of the vehicle during normal dynamic driving. The brake pressure profile based on such a yaw rate feedback is shown in "+" line in Figure 3.19.

When yaw rate is approaching its peak, vehicle roll angles start to build up. Hence a roll-angle-based feedback control can generate effective command to directly counteract the vehicle's roll motion (while the prior feedback controls are indirect control schemes). A PD feedback control law based on φ_{xbw} with the event adaptive control gains and deadbands is introduced. In order to be robust to various driving conditions, φ_{xbw} used here is compensated for the vehicle loading condition that is determined in real time through a conditional least-squares parameter identification algorithm. For a sufficiently aggressive transitional maneuver, the roll momentum can result in a lifting of the CG of the vehicle at the end of the transition maneuver. It is an objective of this φ_{xbw} based PD feedback control to generate effective roll damping before the occurrence of a wheel lift by rounding off the front cornering force, as it peaks out in the final phase of the transition maneuver; see the brake pressure profile in "□" line in Figure 3.19. Note that if the brake pressure request is issued when φ_{xbw} is already built up, it would be too late to control the vehicle's roll motion due to the limitation in the brake hydraulics. The leading indicator of β_{falin} and yaw rate generate feedback control commands to prepare the brake hydraulics for conducting direct roll control which increases the effectiveness of mitigating potential rollovers. When φ_{xbw} increases to certain level, both β_{falin} and yaw rate start to decrease due to the vehicle's energy transferring from side sliding and yawing to rolling, which indicates that feedback based solely on yaw rate and β_{falin} cannot provide sustainable control variables longer enough than the roll angle can. The envelope of all the aforementioned brake pressure commands is used as the final brake pressure command; see the "◯" line in Figure 3.19.

In such a control structure, combining feedbacks with feedforward, the phasing would be such that a particular control is dominant as the transitional maneuver progresses (see Figure 3.19) in a fishhook maneuver, which approximates a panic driver's steering effort in road edge recovery. This control design supports smooth interventions and reduces the potential for exciting pitch dynamics of the vehicle.

Because the Transition Controller is designed to lead the control used in the quasi-steady-state feedback controller (Section 3.4.7.2) for a given maneuver, the quasi-steady-state feedback control can then be initiated when the brake pressure is already reaching a significant level. Hence quasi-steady-state feedback controller requires less magnitude of feedback signals to achieve the critical brake pressure level to stabilize the vehicle.

3.4.7.2 Quasi-Steady-State Control

During a quasi-steady-state dynamical driving condition (usually in the nonlinear vehicle dynamic region but with less dynamical content), a vehicle could experience a slow buildup with extended wheel lift or sideslip angle. For example, during a J-turn maneuver for a vehicle with a high CG (e.g., vehicle with roof loading), the vehicle could have one- or two-wheel lift (roll instability) before the sideslip angle at the vehicle's rear axle is building up (lateral instability). In this case, the rate change of the roll rate, yaw rate, and the driver's steering wheel angle are all at small level such that the aforementioned Transition Controller is no longer effective. On the other hand, for the same maneuver if the vehicle has a low CG, the vehicle might experience slow sideslip buildup (lateral instability) before one- or two-wheel lift (roll instability) occurs. A similar event could occur in a decreasing radius turn such as those on some freeway on- or off-ramps.

The quasi-steady-state conditions cannot be effectively captured by the computations used in ESC due to sensing limitation in ESC sensor set. Under these driving conditions, the ability to detect and accurately estimate the slow buildup of vehicle roll angle and the rear sideslip angle becomes very critical to generate appropriately timed stabilizing torque. Using the RSC sensor set, the proper computation of φ_{xwm} or φ_{xbm} and the rear sideslip angle β_{RSC-ra} (nonlinear in comparison with the linear sideslip angle β_{falin}) are possible. Hence RSC can provide the incremental ability to control the vehicle in the quasi-steady-state region in addition to the highly dynamic rolling and yawing conditions. The relative roll angle between the vehicle body and the moving road frame, φ_{xbm}, and β_{RSC-ra} are the main feedback control variables used in those driving conditions.

For vehicles with a high CG and driven with a rather steady-state steering input, the wheel lift could build up at relatively low lateral accelerations (i.e., before a large rear sideslip angle is built up), thus leading to the buildup of the wheel departure angle. Since the Transition Controller described earlier does not address this scenario, φ_{xbm} provides a unique characterization of such quasi-steady-state conditions, an effective roll-angle-based feedback control command is possible. A PID feedback structure based on the relative roll angle between the body and the road (including wheel departure angle) φ_{xbm} is used in Lu et al. (2007b). The PID controller deadbands and gains are established in a way that an appropriately progressive brake pressure level is requested during the increase of wheel departure angle (roll instability), while allowing for vehicle to behave well in limit handling maneuvers without unnecessary brake interventions whenever the wheel departure angle is significantly small (roll stable condition).

For cases where a vehicle is operating with a low CG and is being driven in a steady-state maneuver near the vehicles' handling limit, the vehicle may experience an abrupt wheel lift if the vehicle's sideslip angle at the rear axle builds up above certain threshold, that is, the rear sideslip angle can slowly build up before a roll instability develops. In those cases, the roll angle φ_{xbm} based feedback control will be nonexistent; yet the buildup of rear sideslip angle can occur at a slow rate. If such a condition is left undetected and uncontrolled, the slowly growing rear sideslip angle β_{RSC-ra} can eventually lead to sudden roll instability. Hence, in this case, the calculated rear sideslip angle β_{RSC-ra} provides the ability to detect the slow sideslip buildup and a PD feedback control using β_{RSC-ra} as the control variable is devised to design control law to counteract such a diverging sideslipping tendency of the vehicle.

During a J-turn maneuver for a vehicle with nominal load, RSC sideslip angle control requests brake pressure on the outside front wheel that extends beyond the ESC pressure request. The details can be found in Lu et al. (2007b). Such a control leads to reduced vehicle sideslip angle that further reduces the cornering force so as to mitigate a potential rollover.

3.4.7.3 Control Integration Inside RSC

The control strategies previously discussed include the feedforward control within the Transition Controller to prepare the brake hydraulics to compensate delays in pressure buildup, the feedback controls within the Transition Controller to mitigate rollover occurring during dynamical conditions such as fish-hooks and double-lane changes, and the feedback controls in quasi-steady-state feedback controller to

mitigate rollovers occurring during nondynamical conditions such as J-turn and decreasing radius turns. In order to achieve a coordinated or combined control strategy, integration among the aforementioned control strategies need to be conducted. Figure 3.20 provides a schematic overview of an example of such integration.

3.4.7.4 RSC Interaction with the Other Functions

The ESC system gives a driver the full ability to control the vehicle, but with intervention when needed to control the vehicle follow the driver's intent. One of the biggest differentiators between ESC and RSC is that the brake control in RSC is no longer solely in response to driver intent. It is possible that the RSC system may cause the vehicle to reduce the lateral force at the outside tire patches, which could lead to the activation of the ESC system to request understeer reduction control during the RSC activation, that is, the RSC function is counteracted by the ESC understeer reduction control. For this reason, it is important to integrate the RSC and ESC functions.

On the other hand, if during an RSC activation ESC oversteer control is also activated, the arbitrated brake pressure should pick the maximum between the ESC oversteer control command and the RSC control command together with a large slip target control function. Note that RSC must also be integrated with the ABS function. While ABS aims to maintain a certain slip target to optimize stopping distance and steerability during an ABS braking, RSC will likely request an alternate slip target, so as to modulate cornering forces and subsequently reduce the resulting roll moment of the vehicle. Since the AWLD is checking if a potentially lifted inside wheel will develop longitudinal slip ratio as a result from a small brake pressure build requested for that wheel, it can enter ABS event. Therefore, the AWLD used in RSC will also need to interact with the ABS function.

The RSC system resides in the brake electronic control unit (ECU) where the ABS, TCS, and ESC functions typically reside, such that the integration between RSC and the existing brake control functions can be easily implemented. A block diagram for such an integration is shown in Figure 3.21, where the lower block depicts the brake ECU which is divided into two parts: the lower portion contains the existing functions, their priority, and arbitration logic together with the other modules such as the sensor fail-safe and interface logic; the upper portion includes the RSC function and its priority and arbitration logic.

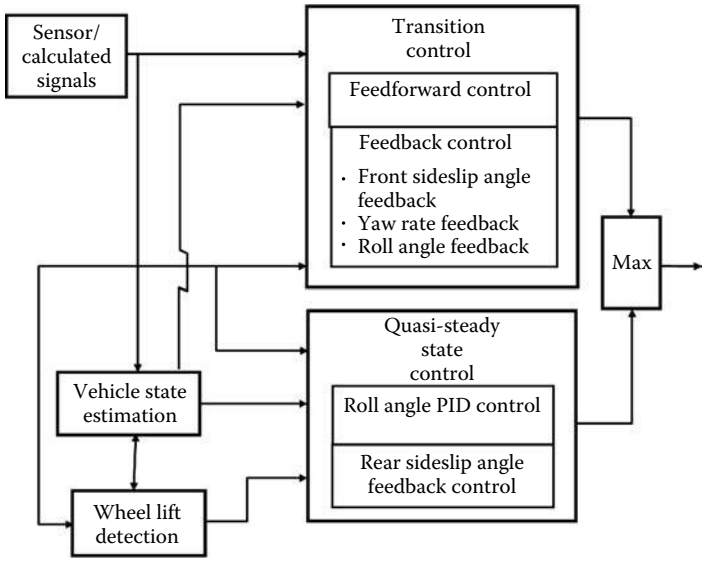

FIGURE 3.20 RSC algorithm integration.

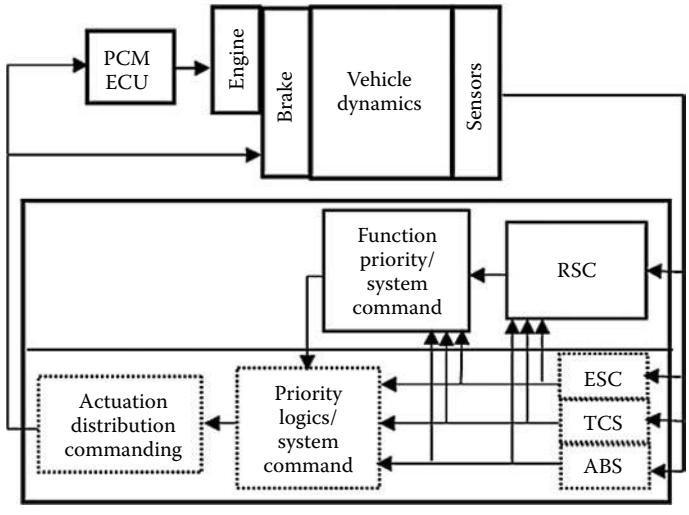

FIGURE 3.21 Function partition in a brake control ECU.

3.4.8 Summary

ESC systems provide stability enhancement and handling predictability of a vehicle. Not unlike ABS and TC systems, the system requires a driver's command to assist the driver in achieving the desired maneuver safely under a variety of driving conditions. For the system to be valuable to the driver, it is important for the active system to recognize the driver's intention, to know the current vehicle status, and to assist the driver in a nonintrusive and cost-effective manner. This section addressed these practical concerns. With future sensors/actuators price reduction, possible addition of sensing technologies, and addition of control authorities in steering and suspensions, ESC can be expected to provide additional safety features and further improve driver/vehicle interaction.

3.5 Electronic Differential Control

3.5.1 Introduction

Vehicle handing control systems are becoming commonplace in the automotive industry. The majority of such systems currently in production are ESC systems that use brake interventions at individual wheels to develop yaw moments that protect vehicle stability at times when it would otherwise be compromised. These systems are highly effective from a vehicle dynamics and stability perspective and are now standard on premium vehicles. However, for the next generation of vehicle handling control systems, the focus of the premium vehicle manufacturers is not just to improve stability but also to increase driver enjoyment whilst driving both below and at the limits of adhesion. Brake based stability systems tend to be somewhat intrusive, causing decelerations and loss of vehicle speed.

Alternative actuation systems are therefore being considered that may provide increased stability, without the intrusiveness of a brake-based system. Active limited slip differentials (ALSDs) that allow electronically controlled transfer of torque between the driven wheels are one such alternative. Controlled torque transfer across an axle allows a yaw moment to be generated that can be used to increase vehicle stability and because wheel torque is reapportion, rather than reduced, this increase in stability can be achieved in a less intrusive manner than would be possible with a brake-based control system.

In order to deliver stability benefits on an actual vehicle, a practical ALSD control algorithm is required. Key challenges with respect to the development of such a controller are the one-directional semiactive nature of the actuator to be elaborated below and their relatively slow dynamic response (in comparison to a brake system). One of the alternatives for designing such a controller is the utilization of H-Infinity (Hinf) controller, a modern control synthesis that offers the capability to formulate an optimization problem in frequency domain. This section describes the development of a practical feedback controller for an ALSD. The controller is applied to an actual vehicle and the stability control performance of the system is assessed.

3.5.2 Active Differentials

Traditional open differentials distribute the same amount of torque to the left and right wheels, while allowing them to rotate at different speeds. Active differentials utilize clutches to provide a controlled left/right (or front/rear) torque distribution to the wheels, thus resulting in enhancing the traction control and YSC performances in a smooth, well-controlled manner (Sawase and Sano, 1999).

The ALSD (Figure 3.22a) utilizes a single clutch that connects the differential rotating case with one of the output shafts. Since the case speed is equal to $\omega_c = (\omega_1 + \omega_2)/2$ (Hrovat et al., 2000) and the clutch always transfers the torque from its faster to slower shaft, the ALSD can provide the torque transfer to slower wheel only. The direction of torque transfer, and therefore the direction of the yaw moment applied is determined by the wheel speed difference across the axle. In this sense an ALSD is therefore only a semiactive device (Hrovat, 1997). While this is effective for traction control (the slower wheel has better traction), it is sometimes insufficient for high-performance YSCs since only understeer torque can be generated (torque transfer to inner/slower wheel).

Torque vectoring differentials (TVDs) can transfer the torque to both slower and faster wheel, thus providing full active yaw control functionality (understeer and oversteer (Gillespie, 1992) can be generated). This can be achieved by extending the ALSD hardware by an additional gearing and an additional clutch, as shown in Figure 3.22b (Sawase and Sano, 1999). The spur gear set $z_1 - z_4 - z_5 - z_2$ speeds up the input shaft of the clutch F_2 (the gear ratio $h_2 = z_1 z_5/(z_4 z_2) > 1$), thus allowing the clutch F_2 to transfer the torque to the right wheel even if it rotates faster than the left wheel. Similarly, the gear set $z_1 - z_4 - z_6 - z_3$ slows down the input shaft of the clutch $F_1 (h_1 = z_1 z_6/(z_4 z_3) < 1)$, so that the torque can be taken from the right wheel and brought to the left wheel even if the left wheel rotates faster than the right wheel. For the particular values of gear ratios $h_1 = 0.875$ and $h_2 = 1.125$, the torque can be transferred to the

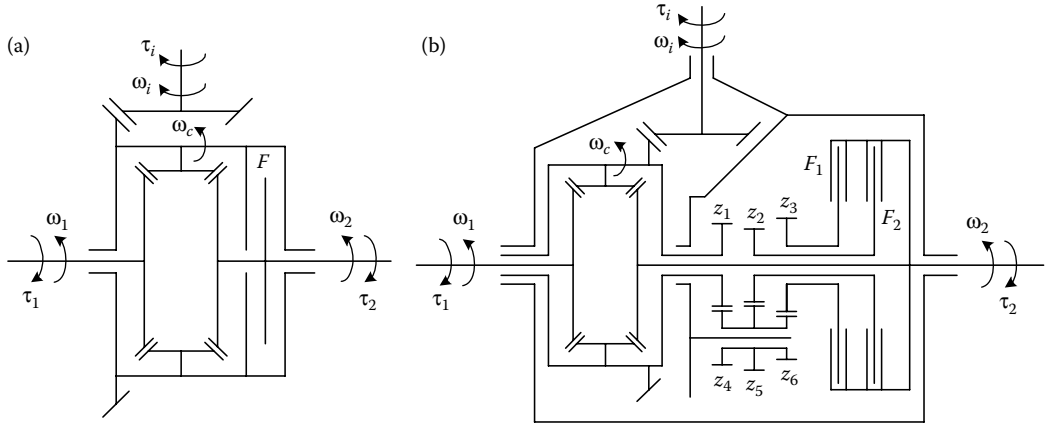

FIGURE 3.22 Kinematic schemes of ALSD (a) and TVD (b).

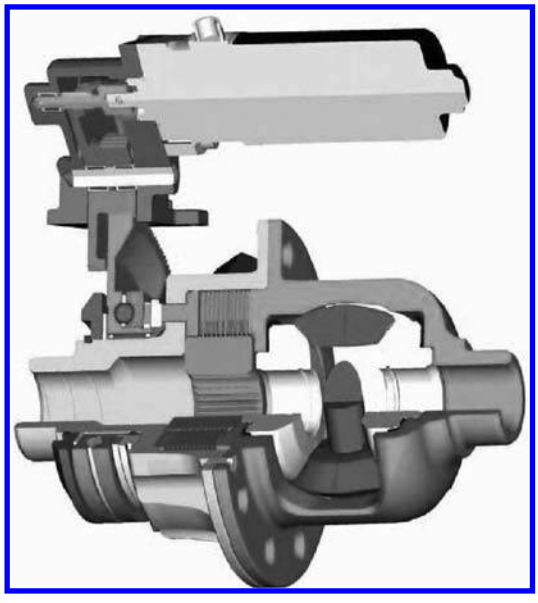

FIGURE 3.23 Schematic representation of the ALSD fitted to the prototype vehicle.

faster wheel if its speed is not larger than the slower wheel speed by more than 28.6% (Sawase and Sano, 1999; Deur et al., 2008a). This is an ample reserve for a variety of VDC scenarios (Assadian and Hancock, 2005). Some other characteristic kinematic structures of TVDs are described and analyzed in Deur et al. (2008a) and Sawase et al. (2006).

In addition to the disadvantage of constrained torque transfer direction, the ALSD can suffer from an inaccurate and slow torque transfer response (Deur, 2008b). This is because the engaged clutch can easily become locked for mild turns (the slip speed $\omega_f = (\omega_1 - \omega_2)/2$ drops to zero), and the locked clutch is not controllable. On the other hand, due to the use of additional gearing the TVDs' clutches have an increased slip speed and can hardly be locked. This gives a favorable controllability, but the power losses are larger than for the ALSD.

The vehicle used for this investigation is a rear wheel drive saloon. A schematic representation of the ALSD employed is shown in Figure 3.23. It features a wet friction clutch pack that transfers torque between the two drive shafts. The clamping force on the clutch pack is controlled by an electric motor driven actuation system that acts through a ball-and-ramp mechanism. The response of the differential can be characterized by a pure delay followed by a first-order lag (Figure 3.24).

3.5.3 Control Design

The structure of the ALSD stability controller is shown in Figure 3.25. As can be observed from the figure, the controller is reference model based. The reference model is a relatively simple vehicle model that generates a target yaw rate based on vehicle speed, driver steering input, and estimated road surface friction coefficient. The error between this target yaw rate and the vehicle's actual yaw rate is then fed through a feedback controller in order to generate a required torque transfer for the ALSD. Since the ALSD can only generate an understeer developing yaw moment (when off throttle), this torque transfer is only applied when the calculated yaw error indicates oversteer. It is the design of the feedback controller that will be the focus of this section.

In order to analyze the potential impact controlled differentials can have on vehicle performance, it is first necessary to develop a control algorithm. It is clear from Section 3.5.2 that a controlled differential is

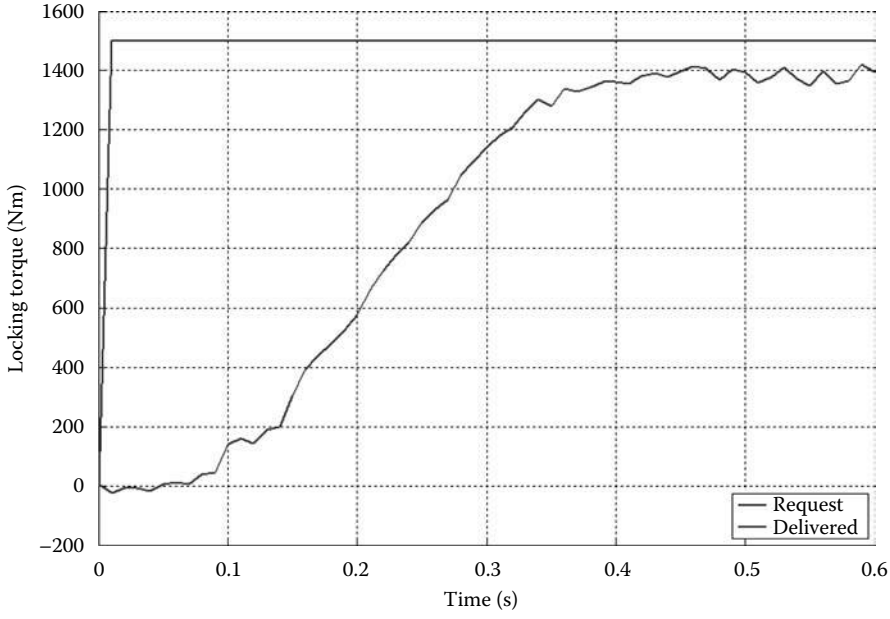

FIGURE 3.24 Actuator time response.

capable of influencing performance in the two relatively broad areas, yaw moment/stability and traction (Assadian and Hancock, 2005). These two areas are initially considered separately. This section describes the development of the YSC algorithm.

The plant model used to design the controller is a two-degree-of-freedom linear bicycle model. The degrees of freedom are yaw and sideslip velocity. In state-space form, this model may be written as

$$\begin{bmatrix} \dot{V} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Y_V/M & Y_r/M - U \\ N_V/I_{zz} & N_r/I_{zz} \end{bmatrix} \begin{bmatrix} V \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 1/I_{zz} \end{bmatrix} [T_{yaw}] + \begin{bmatrix} Y_\delta/M \\ N_\delta/I_{zz} \end{bmatrix} [\delta]$$

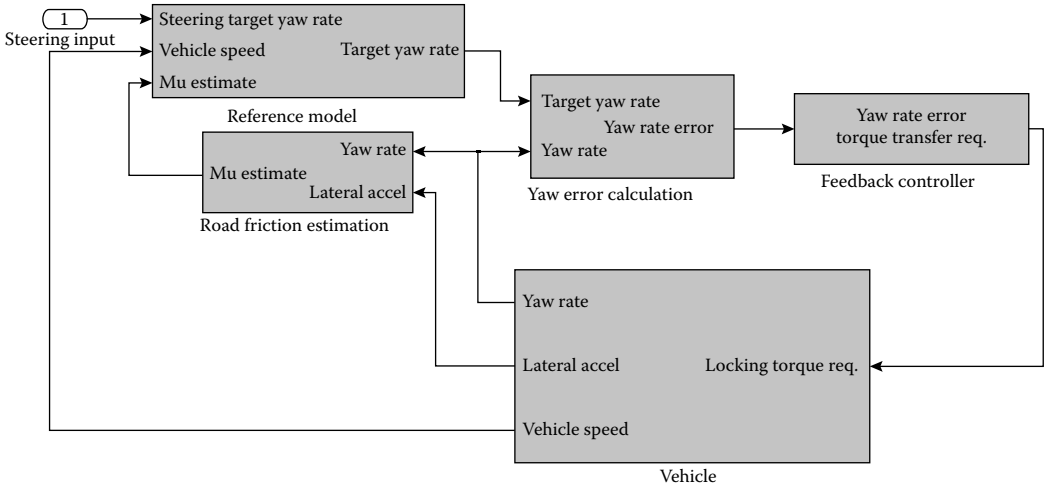

FIGURE 3.25 Controller structure.

or

$$\dot{X} = AX + Bu + Gw,$$

where

$$Y_V = \frac{-C_{\alpha f} - C_{\alpha r}}{U}, \quad Y_r = \frac{-bC_{\alpha f} + cC_{\alpha r}}{U}, \quad Y_\delta = C_{\alpha f},$$
$$N_V = \frac{-bC_{\alpha f} + cC_{\alpha r}}{U}, \quad N_r = \frac{-b^2C_{\alpha f} - c^2C_{\alpha r}}{U}, \quad N_\delta = bC_{\alpha f},$$

where M is the vehicle mass, I_{zz} is the mass moment of inertia around z-axis, $C_{\alpha f}$ is the front tire cornering stiffness, $C_{\alpha r}$ is the rear tire cornering stiffness, U is the longitudinal vehicle velocity, b is the distance from CG to the front axle, and c is the distance from CG to the rear axle.

It is important to note that the YSC has access to both yaw rate and yaw acceleration. Hence, the final controller is a multiinput single output (MISO) controller. The usual method of accessing yaw acceleration is to expand the state space with yaw acceleration by differentiating the state equations. However, an easier approach is to augment the output matrix C to obtain the yaw acceleration as follows:

$$\begin{bmatrix} r \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ N_V/I_{zz} & N_r/I_{zz} \end{bmatrix} \begin{bmatrix} V \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 1/I_{zz} \end{bmatrix} [T_{yaw}] + \begin{bmatrix} 0 \\ N_\delta/I_{zz} \end{bmatrix} [\delta]$$

or

$$y = CX + D_2u + D_1w.$$

The body yaw torque, T_{yaw} is then mapped statically to torque transfer (locking torque request), ΔT, as shown in Figure 3.25.

The previous model contains the nominal plant, P_0. The Hinf approach (Mayne, 1996) requires augmentation of the nominal plant by shaping filters, as illustrated in Figure 3.26, for derivation of the final controller. The filters that are used for the nominal plant augmentation are given as follows,

$$W_1 = \begin{bmatrix} W_{11} = \dfrac{K\omega_n^2}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \\ W_{12} = K \end{bmatrix},$$

where, K is a constant, ω_n is the natural frequency, and ς is the damping ratio.

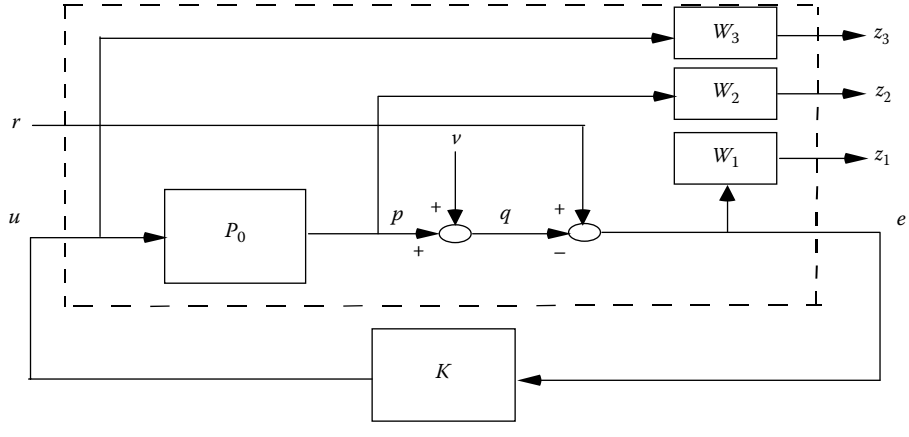

FIGURE 3.26 Nominal augmented closed-loop system.

This filter matrix weights the sensitivity transfer function (disturbance input to controlled output). The W_{11} filter weights the sensitivity transfer function to the yaw rate output, while the W_{12} weights the sensitivity transfer function to the yaw acceleration output. It is also worth mentioning that the closed-loop sensitivity transfer function follows the inverse of these filters. For example, the sensitivity transfer function to the yaw rate output, based on the shaping of the inverse of W_{11} filter, is small at low frequency and increases at high frequency. To simplify matters, the weighting for the yaw acceleration is used as a constant. Due to the limitation on the actuator bandwidth, the benefit of the yaw acceleration feedback is equivalently limited. As the actuator bandwidth is increased, the yaw acceleration feedback benefits become more apparent.

The shaping filter matrix, W_2, for the closed-loop transfer function (complementary transfer function), is given as follows:

$$W_2 = \begin{bmatrix} W_{21} = \dfrac{K_1 s}{s + \dfrac{K_1}{T_{\max}}} \\ W_{22} = W_{21} \end{bmatrix}.$$

This filter matrix is used as a weighting on the complementary transfer function so that that closed-loop stability is guaranteed in the presence of the pure time delay of $T_{\max}$, due to the actuator dynamics.

In order to design the weighting filter W_2 with parameter K_1 and $T_{\max} = 0.1$s that will ensure the robustness in the presence of a pure time delay less than 0.1 s, we know from small gain theory (Helton, 1999) that we need

$$\left| \frac{\hat{P}(jw)}{P(jw)} - 1 \right| \leq W_2$$

where $\hat{P}$ is the actual plant with the pure delay uncertainty and P is the nominal plant. Therefore, we have $|e^{-Tjw} - 1| \leq W_2$. Figure 3.27 illustrates that the designed weighting filter satisfies this condition by showing the bode plots of both sides of the inequality sign.

The final weighting filter is used to limit the actuator effort and is given by

$$W_3 = \frac{1}{\rho},$$

where ρ is a constant and is equivalent to the maximum body yaw torque generated by the actuator.

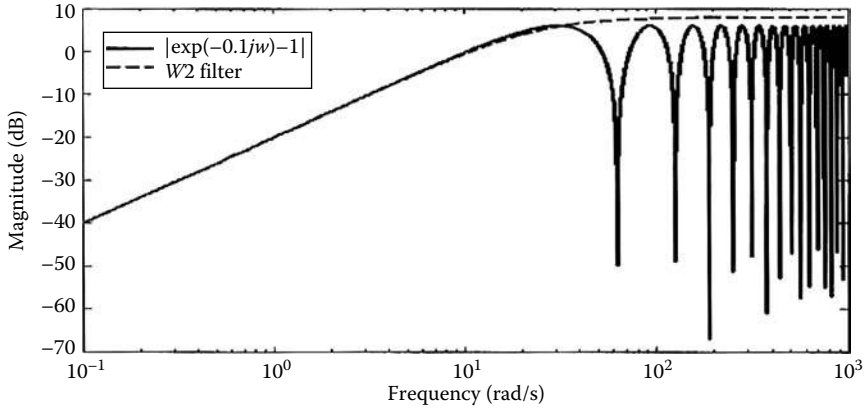

FIGURE 3.27 Bode plot of $|e^{-0.1jw} - 1|$ (solid) and W_2 (dashed).

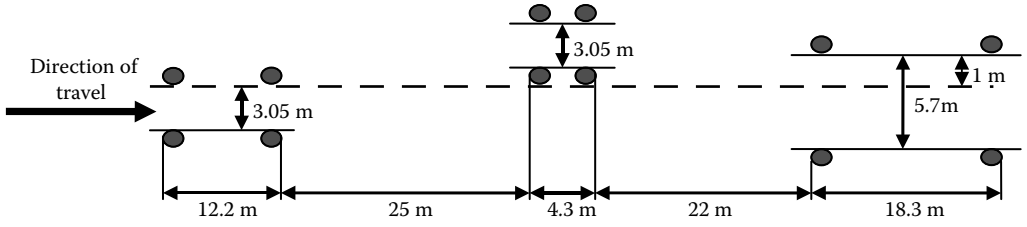

FIGURE 3.28 Double-lane change course.

In addition to the above feedback controller, there is also a feedforward counterpart, derived by inverting the transfer function from the body yaw torque, T_{yaw}, to the yaw rate, r. Then, the steady-state gain of this transfer function is used as a proportional gain for the feed-forward controller.

3.5.4 On-Vehicle Results

The control performance of the system is analyzed in simulation and tested on-vehicle using several different maneuvers. The on-vehicle results of double-lane change maneuver are presented in this section. Here the vehicle is driven through the course shown in Figure 3.28. The maneuver is carried out off throttle (the throttle is released at the entry gate) with the maximum initial speed that still allows the driver to steer through the cones without losing control of the vehicle. Due to the closed-loop nature of this test, large numbers of both passive and controlled runs are required to ensure that the results are meaningful. In addition, passive and controlled runs are continuously interchanged to ensure comparable levels of tire wear, tire temperature, track condition, and driver familiarity in each configuration. Also any runs where the initial speed significantly deviates from the target are discarded.

The steering wheel angle time histories for passive runs carried out with a target initial speed of 125 kph are shown in Figure 3.29. As can be observed, there is significant run to run variation and, in the majority of cases, the driver is applying counter steer at some point in the maneuver.

This double-lane change requires three distinct steering inputs, one to the left to steer toward the second gate, one to the right to steer through the second gate and toward the exit gate and one back to the left to return to straight ahead. Hence, if there is a fourth steering input (as there is in many of the runs), the

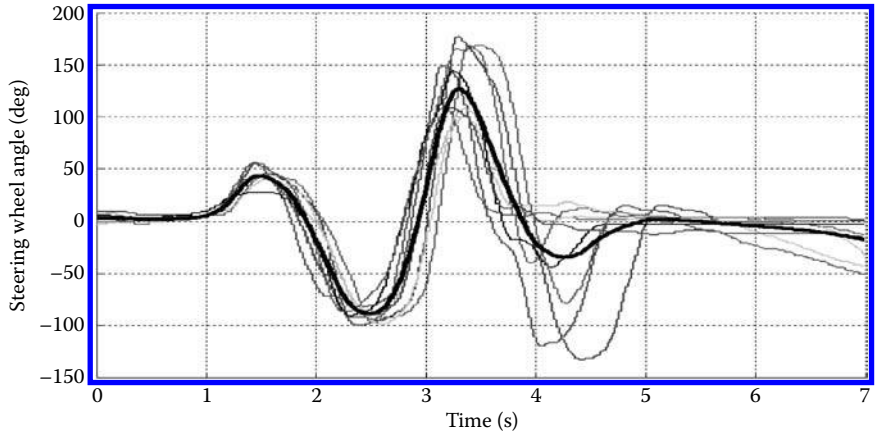

FIGURE 3.29 Steering wheel angle time histories for nine double-lane changes carried out with a target initial speed of 125 kph (average actual initial speed = 127.2 kph). Passive vehicle. Average trace is shown in black.

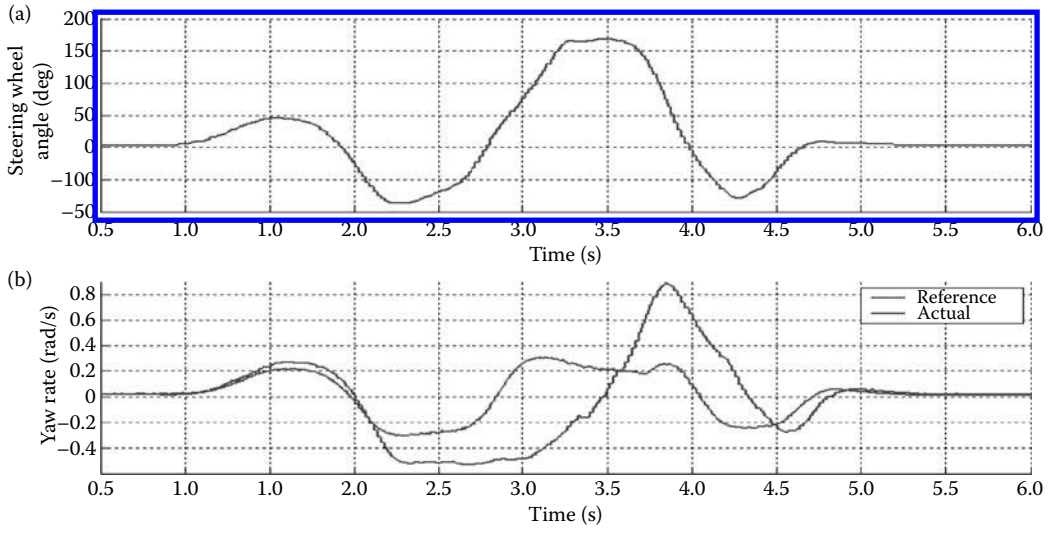

FIGURE 3.30 Example of vehicle and driver behavior during a passive double-lane change carried out with a target initial speed of 125 kph.

vehicle is oversteering as it comes through the exit gate and the driver is applying counter steer. However, the vehicle can also begin to oversteer as it exits the second gate and hence the third steering input can also include counter steer. This point can be illustrated further by considering a specific example (Figure 3.30). Here, it is clear from the comparison of the reference and actual yaw rates (Figure 3.30b) that the vehicle is oversteering after the second steering input as the two yaw rates have become significantly out of phase and the third steering input (Figure 3.30a) is therefore initially at least counter steer, since the steering input and yaw rate are of opposite sign at around 3 s.

The steering wheel angle time histories for eight runs with the controller active are shown in Figure 3.31. As it can be observed, there is significantly less run to run variation than in the passive case, which in itself is an indication that the vehicle is more predictable/controllable and thus easier to drive. This increased consistency is illustrated by the dramatic reduction in the variance of the peak values of the third and

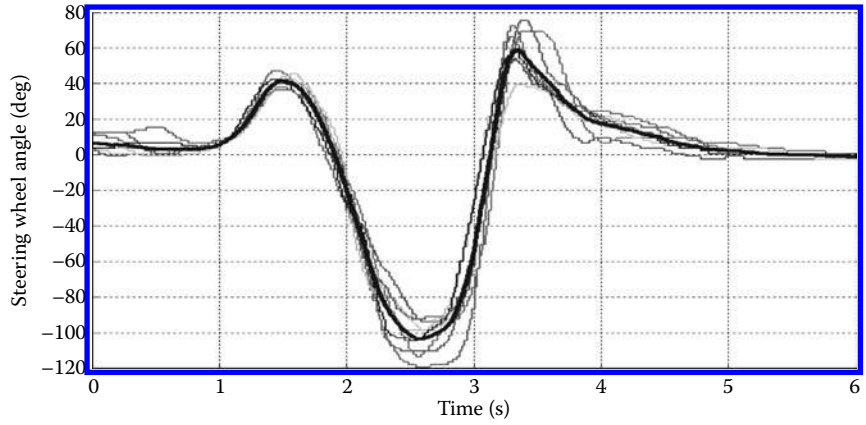

FIGURE 3.31 Steering wheel angle time histories for eight double-lane changes carried out with a target initial speed of 125 kph (average actual initial speed=126.4 kph). Active vehicle. Average trace is shown in black.

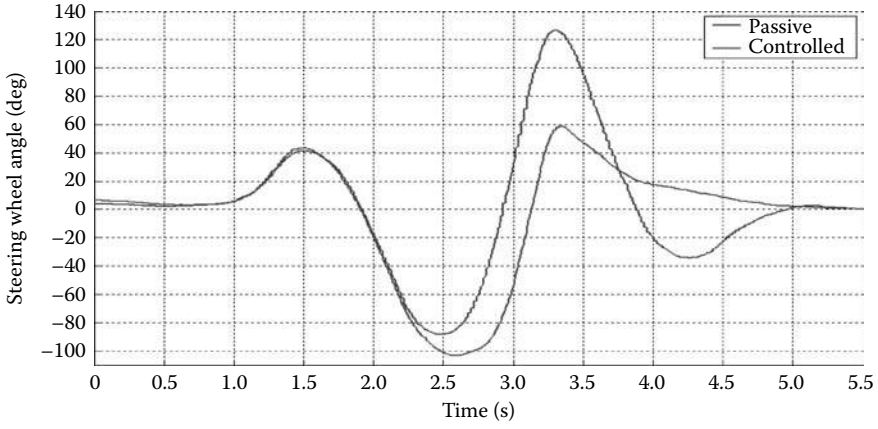

FIGURE 3.32 Comparison of average steering wheel angle time histories for passive and controlled cases during a double-lane changes carried out with a target initial speed of 125 kph.

fourth peak steering inputs of the controlled runs relative to the passive runs. Note that, in the controlled case, counter steer on the exit of the maneuver is largely eliminated (see below), so there is no fourth input.

The impact of the ALSD on driver workload is shown clearly in Figure 3.32 where the average steering inputs from the passive and active cases are compared. Here it can be observed that all counter steer on the exit of the maneuver has been eliminated and the magnitude of the third steering input has also been reduced by over 50%. In both cases, this indicates significantly greater vehicle controllability and stability. Note that the validity of this conclusion is supported by the fact that there is less than 0.5% difference between the average actual entry speed for the two cases (127.2 kph for passive and 126.4 kph for controlled.

Typical passive and actively controlled runs are compared in Figure 3.33. Once again, it is clear from Figure 3.33a that driver workload is dramatically reduced. This is reflected in the controlled vehicle's yaw rate, which is now largely in phase with the reference yaw rate throughout the maneuver (Figure 3.33b).

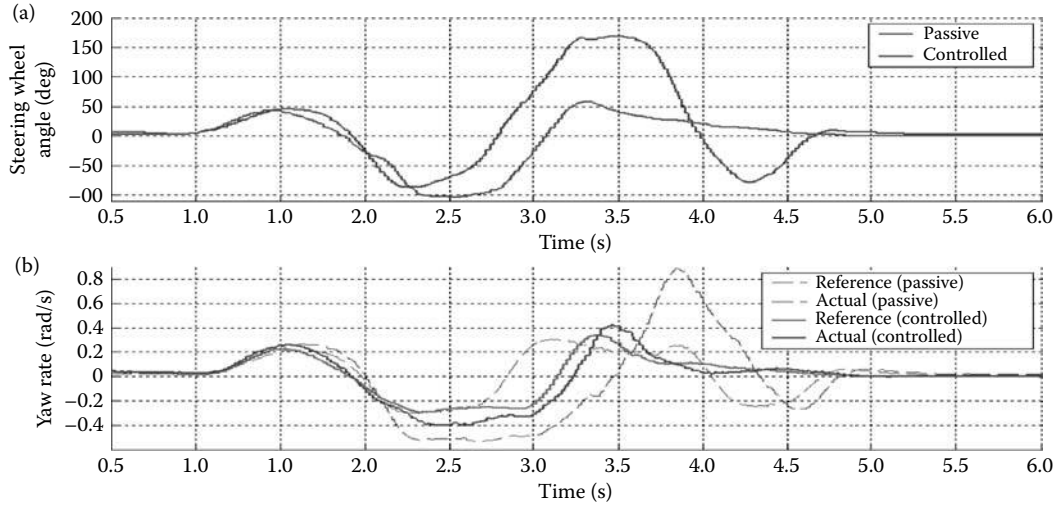

FIGURE 3.33 Comparison of passive and actively controlled example of driver, vehicle, and controller behavior during a double-lane change carried out with a target initial speed of 125 kph.

This result confirms the observation made in the simulation environment that, even in an extreme maneuver such as this, the ALSD is still capable of forcing the vehicle to track the reference yaw rate, even if it does not do so precisely. It achieves this through the application of torque transfer from the point when the steering input begins to return to center after going through the entry gate. This is therefore a confirmation of the effectiveness of an ALSD as a stability control device.

3.5.5 Summary

In this section, we presented several different types of active differentials and their potential use for increasing vehicle controllability and stability. Due to its hardware simplicity and lower cost, the ALSD is selected for further illustration of the corresponding control system development. A MISO robust Hinf controller is designed for the active control of the vehicle yaw. The effectiveness of the ALSD along with the Hinf controller is demonstrated on-vehicle through a double-lane change maneuver at relative high speeds. It is worth mentioning that although the Hinf controller is a linear controller, it is implemented on a highly nonlinear actuator with a pure time delay of 100 ms and rise time of 400 ms. Furthermore, even though it is a semiactive device, which is capable of only transferring torque from a faster to a slower wheel, the ALSD has sufficient impact on the understeer gradient (yaw authority) to stabilize the vehicle under highly dynamic maneuvers.

It is important to note that the ALSD is not designed to replace brake-based stability controllers as they do not have as much vehicle yaw authority as the brake-based controllers. The synchronization of the ALSD and brake-based stability controller is under further investigation, and although the results are not shown here, the two actuators seem to complement each other very well. The ALSD is tuned to intervene much earlier than the brake-based controller, so that, when the brake-based controller becomes active, it utilizes much lower brake pressures; hence, this results in a significant decrease in brake actuator intrusiveness. In addition, when the two actuators are working together, they have more yaw authority and increased impact on the understeer gradient.

3.6 Active Steering Control and Beyond

3.6.1 Introduction

Recent advances in drivetrain and vehicle design has increased the number of possible interventions to influence the vehicle dynamic behavior. In particular, equipments such as Active Front Steering (AFS), four wheels steering, active differentials, and active or semiactive suspensions can be integrated in existing or new active safety systems in order to improve the safety, the comfort, and the agility of the vehicle.

This section focuses on the use of AFS systems to further enhance lateral and yaw vehicle stability (Ackermann, 1990; Ackermann and Sienel, 1993; Ackermann et al., 1999) in autonomous path-following scenarios. We assume that with the increased inclusion of onboard cameras, radars, and infrared sensors, augmented by GPS signals and associated digital maps, vehicles will be able to identify obstacles on the road such as an animal, a rock, or fallen tree/branch, and assist the driver by following the best possible path, in terms of avoiding the obstacle and at the same time keeping the vehicle on the road at a safe distance from incoming traffic. Autonomous obstacle-avoidance evasive maneuver using a steering robot has previously been investigated on nominal high friction surfaces (Tseng et al., 2005). This section summarizes the main results of Borrelli et al. (2005); Falcone et al. (2006a,b, 2007a,b, 2008a,b) where we investigated model predictive control (MPC) approaches to a class of path-following scenarios via AFS and combined AFS and individual braking problems on surfaces with lower friction.

The main concept of MPC is to use a *model* of the plant to *predict* the future evolution of the system (Mayne and Michalska, 1993; Mayne et al., 2000). At each sampling time, starting at the current state of the vehicle, an open-loop optimal control problem is solved over a finite time horizon. The open-loop optimal control problem minimizes the deviations of the predicted outputs from their references over a sequence of future steering angles and braking torques, subject to operating constraints. The resulting optimal

command signal is applied to the process only during the following sampling interval. At the next time step, a new optimal control problem based on new measurements of the state is solved over a shifted horizon.

We will focus on AFS control and its integration with active differential braking by means of MPC. Autonomous driving tests at high speed on icy/snowy roads will be shown through experiments. We point out that the proposed MPC methodologies can be applied to simpler problem classes where information on global position is not required (such as standard yaw rate control design by using differential braking and AFS).

We remark that problem size and systems nonlinearities greatly affect the complexity of the MPC problem. In particular, a large number of actuators and the vehicle strong nonlinearities significantly increase the problem complexity and prevent the real-time implementation of MPC algorithms. Experiments are described in this section highlighting the benefits and the limitations of implementing real-time MPC schemes.

This section is structured as follows: in Section 3.6.2 the vehicle models used next for control design are presented. In Sections 3.6.3 and 3.6.5, we formulate an autonomous path-following task as an MPC problem where the control inputs are the front steering and the combined steering and braking at the four wheels, respectively. Experimental results are presented and discussed. Section 3.6.7 summarizes the section with concluding remarks.

3.6.2 Control-Oriented Vehicle Models

In this section we present vehicle models used next for control design. The models are well known in literature and thoroughly presented in Gillespie (1992) and reported next for the sake of completeness.

3.6.2.1 The Two-Track Vehicle Model

The nomenclature used in the following refers to the model depicted in Figure 3.34 (Falcone et al., 2009). Moreover, two subscript symbols are used throughout the rest of the paper to denote variables related

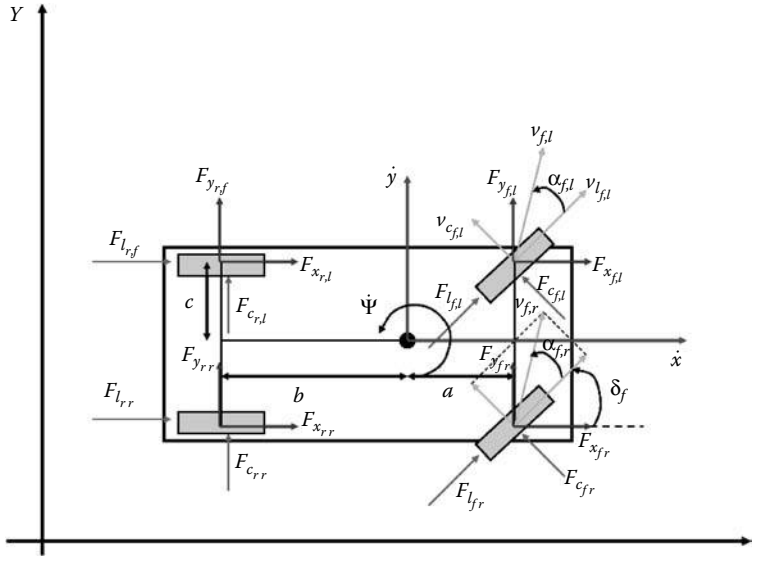

FIGURE 3.34 F_l, F_c are the longitudinal (or "tractive") and lateral (or "cornering") tire forces, respectively, F_x, F_y are the tire forces in car body frame, a, b and c are derived from the car geometry, v_l, v_c are the longitudinal and lateral wheel velocities, $\dot{x}$ and $\dot{y}$ are the longitudinal and lateral vehicle speeds, respectively, α is the tire slip angle, δ_f is the front steering angle, and ψ is the heading angle. The lower scripts $(\cdot)_{f,l}$, $(\cdot)_{f,r}$, $(\cdot)_{r,l}$, and $(\cdot)_{r,r}$ particularize a variable at the front left, front right, rear left, and rear right wheel, respectively.

to the four wheels. In particular, the first subscript, $\star \in \{f, r\}$, denotes the front and rear axles, while the second, $\bullet \in \{l, r\}$, denotes the left and right sides of the vehicle. For example, the variable $(\cdot)_{f,l}$ is referred to the front left wheel.

The longitudinal, lateral, and yaw dynamics of the vehicle are described through the following set of differential equations with respect to the $x-y$ coordinate system fixed with the vehicle:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + \left(F_{l_{f,l}} + F_{l_{f\,r}}\right)\sin\delta_f + \left(F_{c_{f,l}} + F_{c_{f\,r}}\right)\cos\delta_f + F_{c_{r,l}} + F_{c_{r\,r}}, \tag{3.30a}$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + \left(F_{l_{f,l}} + F_{l_{f\,r}}\right)\cos\delta_f + \left(F_{c_{f,l}} + F_{c_{f\,r}}\right)\sin\delta_f + F_{l_{r,l}} + F_{l_{r\,r}}, \tag{3.30b}$$

$$\begin{aligned} I\ddot{\psi} = {} & a\left[\left(F_{l_{f,l}} + F_{l_{f\,r}}\right)\sin\delta_f + \left(F_{c_{f,l}} + F_{c_{f\,r}}\right)\cos\delta_f\right] - b\left(F_{c_{r,l}} + F_{c_{r\,r}}\right) \\ & + c\left[\left(-F_{l_{f,l}} + F_{l_{f,r}}\right)\cos\delta_f + \left(F_{c_{f,l}} - F_{c_{f,r}}\right)\sin\delta_f - F_{l_{r,l}} + F_{l_{r\,r}}\right]. \end{aligned} \tag{3.30c}$$

The vehicle's equations of motion in the absolute inertial frame XY are

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi, \tag{3.31a}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi. \tag{3.31b}$$

The cornering $F_{c_{\star,\bullet}}$ and longitudinal $F_{l_{\star,\bullet}}$ tire forces in Equation 3.30 are given by

$$F_{c_{\star,\bullet}} = f_c(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu_{\star,\bullet}, F_{z_{\star,\bullet}}), \tag{3.32a}$$

$$F_{l_{\star,\bullet}} = f_l(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu_{\star,\bullet}, F_{z_{\star,\bullet}}), \tag{3.32b}$$

where $\alpha_{\star,\bullet}$ are the tire slip angles, $s_{\star,\bullet}$ are the slip ratios, $\mu_{\star,\bullet}$ are the road friction coefficients, and $F_{z_{\star,\bullet}}$ are the tires normal forces. In the following, we assume constant normal tire load, that is, $F_{z_{\star,\bullet}} = \text{constant}$. As shown in Figure 3.34, the slip angle $\alpha_{\star,\bullet}$ in Equation 3.32 represents the angle between the wheel velocity vector $v_{\star,\bullet}$ and the direction of the wheel itself, and can be compactly expressed as

$$\alpha_{\star,\bullet} = \arctan\frac{v_{c_{\star,\bullet}}}{v_{l_{\star,\bullet}}}. \tag{3.33}$$

Remark 1

The tire slip ratios $s_{\star,\bullet}$ at the four wheels are nonlinear functions of the wheel angular speeds. The latter can be computed as the solution of a nonlinear differential equations system, whose right-hand side is function of the braking torques. The equations of $s_{\star,\bullet}$ are not relevant for the rest of the chapter and are omitted here. The interested reader shall refer Falcone et al. (2009) for further details.

The tire longitudinal and cornering forces (Equation 3.32) are described by a Pacejka model (Bakker et al., 1987). Further details can be found in Bakker et al. (1987), Borrelli et al. (2005), and Falcone et al. (2007a). Using Equations 3.30 through 3.33 and the additional wheel dynamics mentioned in Remark 1, the nonlinear vehicle dynamics can be described by the following compact differential equation, assuming a certain road friction coefficient $\mu = \left[\mu_{f,l}, \mu_{f,r}, \mu_{r,l}, \mu_{r,r}\right]$ vector:

$$\dot{\xi}(t) = f^{4w}_{\mu(t)}(\xi(t), u(t)), \tag{3.34}$$

where $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X, \omega_{f,l}, \omega_{f,r}, \omega_{r,l}, \omega_{r,r}]$, $u = [\delta_f, T_{b_{f,l}}, T_{b_{f\,r}}, T_{b_{r,l}}, T_{b_{r\,r}}]$, $T_{b_{\star,\bullet}}$ are the braking torques at the four wheels and $\omega_{\star,\bullet}$ are the wheels angular velocities.

3.6.2.2 The Simplified Two-Track Vehicle Model

The two-track vehicle model presented next (Falcone et al., 2006a) is based on the following set of simplifications.

Simplification 3.1:

Small-angle approximation is used, that is, $\cos\delta_f = 1$ *and* $\sin\delta_f = 0$.

Simplification 3.2:

Single wheel braking is considered on each side of the vehicle, that is, $F_{l_{f,\bullet}} F_{l_{r,\bullet}} = 0$.

Remark 2

By Simplifications 3.1 and 3.2, the effects on the longitudinal and yaw dynamics of the longitudinal tire forces $F_{l_{\star,\bullet}}$ can be described though the forces $F_{l_\bullet} = F_{l_{f,\bullet}} + F_{l_{r,\bullet}}$, with $\bullet \in \{l, r\}$.

By Simplifications 3.1 and 3.2, Equations 3.1 can be rewritten as follows:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{c_{f,l}} + F_{c_{f\,r}} + F_{c_{r,l}} + F_{c_{r\,r}}, \tag{3.35a}$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{l_l} + F_{l_r}, \tag{3.35b}$$

$$I\ddot{\psi} = a\left(F_{c_{f,l}} + F_{c_{f\,r}}\right) - b\left(F_{c_{r,l}} + F_{c_{r\,r}}\right) + c\left(-F_{l_l} + F_{l_r}\right), \tag{3.35c}$$

where F_{l_l} and F_{l_r} are the longitudinal forces induced by braking at the left and right sides, respectively, of the vehicle (see Remark 2). Using Equations 3.31 through 3.35, the nonlinear vehicle dynamics can be described by the following compact differential equation:

$$\dot{\xi}(t) = f^{4w,\text{simpl}}_{s(t),\mu(t)}(\xi(t), u(t)), \tag{3.36}$$

where $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X]$ and $u = [\delta_f, F_{l_l}, F_{l_r}]$, respectively and where $s(t) = \left[s_{f,l}, s_{f,r}, s_{r,l}, s_{r,r}\right](t)$ and $\mu(t) = \left[\mu_{f,l}, \mu_{f,r}, \mu_{r,l}, \mu_{r,r}\right](t)$ are the vectors of slip ratios and road friction coefficients, respectively, at the four wheels at time t.

3.6.2.3 Simplified Single-Track Model

Starting from the vehicle model presented in Section 3.6.2.1, we derive a further simplified single-track (or bicycle) model (Margolis and Asgari, 1991) by introducing the following.

Simplification 3.3:

At the front and rear axles, the left and right wheels are identical and lumped together in a single wheel.

Simplification 3.4:

No braking is applied at the four wheels, that is, $F_{l_{\star,\bullet}} = 0$.

By Simplifications 3.3 and 3.4 Equations 3.30 can be rewritten as follows:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{c_f}\cos\delta_f + 2F_{c_r}, \tag{3.37a}$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2F_{c_f}\sin\delta_f + 2F_{c_r}\sin\delta_f, \tag{3.37b}$$

$$I\ddot{\psi} = 2aF_{c_f}\cos\delta_f - 2bF_{c_r}, \tag{3.37c}$$

where Equations 3.32 are written for a single axle (i.e., the second symbol is dropped). By combining Equations 3.31 through 3.33 and Equation 3.37, the simplified bicycle model can be described by the following compact differential equation:

$$\dot{\xi}(t) = f^{2w}_{s(t),\mu(t)}(\xi(t), u(t)), \tag{3.38}$$

where the state and input vectors are $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X]$ and $u = \delta_f$, respectively and where $s(t) = \left[s_f, s_r\right](t)$ and $\mu(t) = \left[\mu_f, \mu_r\right](t)$ are the vectors of slip ratios and road friction coefficients, respectively, at the two axles at time t.

3.6.2.4 Simplified Single-Track Model with Braking Yaw Moment

Clearly, the simplified bicycle model presented in Section 3.6.2.3 does not model the effects of individual braking on the yaw dynamics. Next we introduce further simplifications in order to derive from model (Equation 3.34) a simple bicycle model accounting for the effect of individual wheel braking (Falcone et al., 2008b).

Simplification 3.5:

Braking application induces only yaw moment without longitudinal and/or lateral force changes.

Simplification 3.6:

In the lateral tire force calculation the tire slip ratio is assumed to be zero, that is, $F_c = f_c(\alpha, 0, \mu, F_z)$.

Simplification 3.7:

The steering and braking effects on vehicle speed are negligible.

Remark 3

Since braking is used for yaw stabilization only, minimum and single-sided usage of brakes is expected. The Simplifications 3.5 through 3.7 are therefore deemed reasonable.

By the Simplifications 3.3 and 3.5 through 3.7, we have

$$\dot{x} \simeq 0, \tag{3.39a}$$

$$2F_{y_f} \simeq 2F_{c_f}\big|_{s=0} \cos\delta_f, \tag{3.39b}$$

$$2F_{y_r} \simeq 2F_{c_r}\big|_{s=0}. \tag{3.39c}$$

According to Equations 3.39 and 3.30, the simplified bicycle model can be rewritten as follows:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2F_{y_f} + 2F_{y_r}, \tag{3.40a}$$

$$\ddot{x} = 0, \tag{3.40b}$$

$$I\dot{\psi} = 2aF_{y_f} - 2bF_{y_r} + M, \tag{3.40c}$$

where M is the braking yaw moment, in the four-wheel model (3.30), that is computed as follows:

$$M = c\left(-F_{x_{f,l}} + F_{x_{f\,r}} - F_{x_{r,l}} + F_{x_{r,r}}\right). \tag{3.41}$$

The forces in Equations 3.39 can be computed through the Equations 3.32 and 3.33, particularized for the front and rear axles.

Remark 4

We remark that the forces F_{y_f} and F_{y_r} in Equations 3.40 represent the lateral components of the cornering tire forces F_c generated by the contact of a single wheel with the ground.

The nonlinear vehicle dynamics described by Equations 3.31 through 3.33 and 3.39 and 3.40 can be rewritten in the following compact form:

$$\dot{\xi}(t) = f_{\mu(t)}^{2w,brk}(\xi(t), u(t)), \tag{3.42}$$

where $\mu(t) = [\mu_f(t), \mu_r(t)]$. The state and input vectors are $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X]$ and $u = [\delta_f, M]$, respectively.

3.6.3 Active Steering Controller Design

In this section we present two control design procedures for the proposed path-following problem via an AFS system. The controller design procedures follow a predictive model-based approach. In particular, in Section 3.6.3.1 we present an MPC algorithm based on the nonlinear bicycle model in Section 3.6.2.3 and requiring the solution of a nonlinear constrained optimization problem every time step. A lower complexity design approach is presented in Section 3.6.3.2, requiring the solution of a quadratic programming (QP) problem.

3.6.3.1 Nonlinear Model Predictive Controller (NMPC) Active Steering Controller

Desired references for the heading angle ψ and the lateral distance Y define a desired path over a finite time horizon. The nonlinear vehicle dynamics (Equation 3.38) and the Pacejka tire model are used to predict the vehicle behavior, and the front steering angle δ_f is chosen as control input.

In order to obtain a finite-dimensional optimal control problem we discretize the system dynamics (Equation 3.38) with a fixed sampling time T_s:

$$\xi(t+1) = f_{s(t)\,\mu(t)}^{2w,dt}(\xi(t), \Delta u(t)), \tag{3.43a}$$

$$u(t) = u(t-1) + \Delta u(t), \tag{3.43b}$$

where the Δu formulation is used, with $u(t) = \delta_f(t)$, $\Delta u(t) = \Delta\delta_f(t)$.

We define the following output map for yaw angle and lateral position states:

$$\eta(t) = h(\xi(t)) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \xi(t), \tag{3.44}$$

and consider the following cost function:

$$J(\xi(t), \Delta\mathcal{U}_t) = \sum_{i=1}^{H_p} \|\eta_{t+i,t} - \eta_{ref\ t+i,t}\|_Q^2 + \sum_{i=1}^{H_c-1} \|\Delta u_{t+i,t}\|_R^2, \tag{3.45}$$

where, $\eta = [\psi, Y]$ and η_{ref} denotes the corresponding reference signal. At each time step t, the following finite horizon optimal control problem is solved:

$$\min_{\Delta\mathcal{U}_t} J(\xi_t, \Delta\mathcal{U}_t)$$

subject to

$$\xi_{k+1,t} = f_{s_{k,t},\mu_{k,t}}^{2w,dt}(\xi_{k,t}, \Delta u_{k,t}), \tag{3.46a}$$

$$\eta_{k,t} = h(\xi_{k,t}), \quad k = t, \ldots, t + H_p, \tag{3.46b}$$

$$\delta_{f,min} \le u_{k,t} \le \delta_{f,max}, \tag{3.46c}$$

$$\Delta\delta_{f,min} \le \Delta u_{k,t} \le \Delta\delta_{f,max}, \tag{3.46d}$$

$$u_{k,t} = u_{k-1,t} + \Delta u_{k,t}, \quad k = t, \ldots, t + H_c - 1, \tag{3.46e}$$

$$\Delta u_{k,t} = 0, \quad k = t + H_c, \ldots, t + H_p, \tag{3.46f}$$

$$s_{k,t} = s_{t,t}, \quad k = t, \ldots, t + H_p, \tag{3.46g}$$

$$\mu_{k,t} = \mu_{t,t}, \quad k = t, \ldots, t + H_p, \tag{3.46h}$$

$$\xi_{t,t} = \xi(t), \tag{3.46i}$$

where $\Delta\mathcal{U}_t = \left[\Delta u_{t,t}, \ldots, \Delta u_{t+H_c-1,t}\right]$ is the optimization vector at time t, $\eta_{t+i,t}$ denotes the output vector predicted at time $t+i$ obtained by starting from the state $\xi_{t,t} = \xi(t)$ and applying to systems (Equations 3.43 and 3.44) the input sequence $\Delta u_{t,t}, \ldots, \Delta u_{t+i,t}$ and H_p and H_c denote the output prediction horizon and the control horizon, respectively. We use $H_p > H_c$ and the control signal is assumed constant for all $H_c \leq t \leq H_p$. We assume that slip and friction coefficient values are constant and are equal to the estimated values at time t over the prediction horizon (constraints (Equations 3.46g through 3.46h)).

In Equation 3.45, the first summand reflects the penalty on trajectory tracking error, while the second summand penalizes the steering changes between two subsequent sampling time instants. Q and R are weighting matrices of appropriate dimensions.

We denote the sequence of optimal input increments by $\Delta\mathcal{U}_t^* \triangleq [\Delta u_{t,t}^*, \ldots, \Delta u_{t+H_c-1,t}^*]'$ computed at time t by solving Equation 3.46 for the current observed states $\xi(t)$. Then, the first sample of $\Delta\mathcal{U}_t^*$ is used to compute the optimal control action and the resulting state feedback control law is

$$u(t, \xi(t)) = u(t-1) + \Delta u_{t,t}^*(t, \xi(t)). \tag{3.47}$$

At the next time step $t+1$, the optimization problem 3.46 is solved over a shifted horizon based on the new measurements of the state $\xi(t+1)$.

Remark 5

The problem (Equation 3.46) is a nonlinear, in general, nonconvex, constrained optimal control problem. Depending on the vehicle operating conditions, solving the problem (Equation 3.46) might require complex computational infrastructures. In Section 3.6.3.2, we present an alternative, lower complexity MPC problem formulation. Hereafter, the controller (Equations 3.46 and 3.47) will be referred to as NMPC.

3.6.3.2 Linear Time-Varying MPC (LTV MPC) Active Steering Controller

The computational complexity of the NMPC controller (Equations 3.46 and 3.47) is strongly dependent on the vehicle-operating conditions and its real-time implementation might be restricted to small operating regions (see Remark 5).

In this section we present a lower complexity MPC scheme, compared to the controller (Equations 3.46 and 3.47). We observe that nonlinearity and possibly nonconvexity of problem (Equation 3.46) stem from the nonlinear vehicle dynamics (Equation 3.46a). Moreover, cost function (Equation 3.45) is quadratic and constraints (Equation 3.46b through 3.46i) are linear. Hence, in order to formulate an MPC controller based on a convex optimization problem, we replace the nonlinear vehicle dynamics (Equation 3.46a) with linearized dynamics computed every time step, based on the current vehicle state and the previously applied control input. In particular, at time step t, let $\xi(t)$, $u(t-1)$ be the current state and the previous input of system (Equation 3.38), respectively. We consider the optimization problem obtained from Equation 3.46, by replacing the nonlinear discrete-time dynamics (Equation 3.46a) with

the approximated linear dynamics

$$\xi_{k+1,t} = \mathcal{A}_t \xi_{k,t} + \mathcal{B}_t u_{k,t} + d_{k,t}, \quad k = t, \ldots, t + H_p - 1, \tag{3.48}$$

where

$$\mathcal{A}_t = \left.\frac{\partial f_{s_t,\mu_t}^{2w,dt}}{\partial \xi}\right|_{\xi_t,u_t}, \qquad \mathcal{B}_t = \left.\frac{\partial f_{s_t\,\mu_t}^{2w,dt}}{\partial u}\right|_{\xi_t,u_t}, \tag{3.49a}$$

$$d_{k,t} = \xi_{k+1,t} - \mathcal{A}_t \xi_{k,t} - \mathcal{B}_t u_t. \tag{3.49b}$$

The resulting optimization problem can be recast as a QP (details can be found in Borrelli et al. (2005)), hence convex. Hereafter, we will refer to this lower complexity MPC formulation as LTV MPC.

We observe that, when evaluating the online computational burden of the proposed scheme, in addition to the time required to solve the optimization problems (Equations 3.46, 3.48, and 3.49), one needs to consider the resources spent in computing the linear models $(\mathcal{A}_t, \mathcal{B}_t)$ in Equation 3.48 and translating Equations 3.46, 3.48, and 3.49 into a standard QP problem. Nevertheless, for the proposed application, complexity of the LTV MPC scheme greatly reduces when compared to the NMPC controller. This will be shown for a specific scenario in Sections 3.6.4.1 and 3.6.4.2.

We point out that using the linearized dynamics (Equation 3.48) instead of Equation 3.46a leads to a loss of controller performance. Extensive simulations have showed that, as the vehicle speed increases, the LTV MPC controller is not able to limit the tire slip angles and force the vehicle to operate within a stable operating region. On the other hand, simulation results of the NMPC controller up to 17 m/s on snow, reported in Borrelli et al. (2005), demonstrate that, due to the knowledge of the tire characteristics, the NMPC controller implicitly limits the front tire slip angle (α_f) in the interval $[-3, +3]$deg. This range is approximately within the linear region of the tire characteristic for a snow covered road ($\mu = 0.3$), where maximum lateral tire forces are achieved. Extensive simulations have shown that this phenomenon can always be observed in extreme driving conditions and led us to the use of an additional constraint in the LTV MPC formulation. In particular, we have added the constraints

$$\begin{bmatrix} \alpha_{f_{k,t}} \\ \alpha_{r_{k,t}} \end{bmatrix} = \mathcal{C}_t \xi_{k,t} + \mathcal{D}_t u_{k,t} + e_{k,t}, \tag{3.50a}$$

$$\alpha_{f_{min}} - \varepsilon \leq \alpha_{f_{k,t}} \leq \alpha_{f_{max}} + \varepsilon, \tag{3.50b}$$

$$\alpha_{r_{min}} - \varepsilon \leq \alpha_{r_{k,t}} \leq \alpha_{r_{max}} + \varepsilon, \tag{3.50c}$$

$$k = t+1, \ldots, t + H_u$$

$$\varepsilon \geq 0, \tag{3.50d}$$

where the matrices $\mathcal{C}_t, \mathcal{D}_t$ and the vector e are computed by linearizing Equation 3.33, $H_u \leq H_p$ is the constraint horizon, ε is a slack variable, and $\alpha_{\star_{max}}, \alpha_{\star_{min}}$ are bounds on the tire slip angles, respectively.

In this lower complexity MPC formulation, in order to include a yaw rate reference, the output map is modified as follows:

$$\eta(t) = h(\xi(t)) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \xi(t). \tag{3.51}$$

3.6.4 Double-Lane Change with Active Steering

The NMPC and the LTV MPC steering controllers described in Section 3.6.3 have been implemented to perform a sequence of double-lane changes at different entry speeds. The desired path is shown in Figure 3.35. This test represents an obstacle avoidance emergency maneuver in which the vehicle is

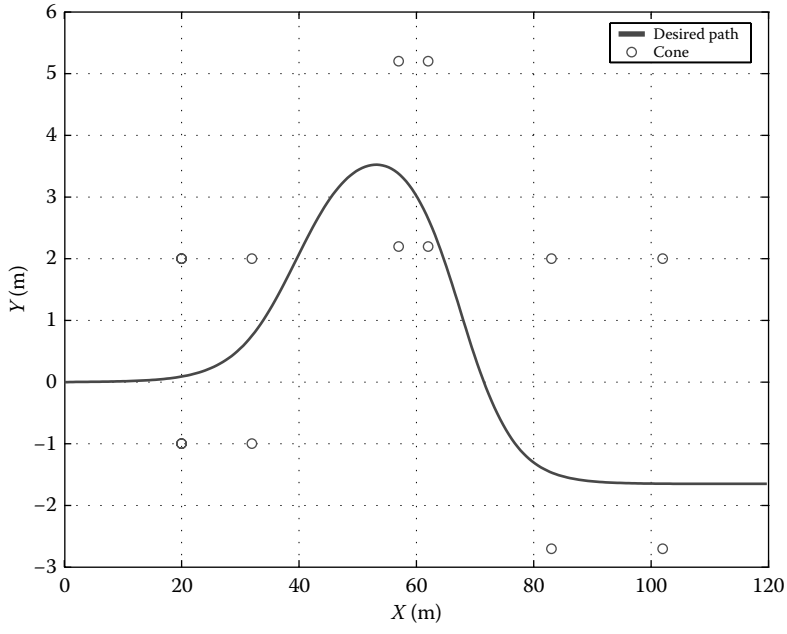

FIGURE 3.35 Reference path to be followed.

entering a double-lane change maneuver on snow or ice with a given initial forward speed. The control input is the front steering angle and the goal is to follow the trajectory as close as possible by minimizing the vehicle deviation from the target path. The experiment is repeated with increasing entry speeds until the vehicle loses control.

The same controller can be used to control the vehicle during different maneuvers in different scenarios. In Keviczky et al. (2006) the same MPC controller is used in order to evaluate the effect of an external side wind gust on the vehicle. The study allowed to estimate the maximum wind speed that an MPC-based active steering system is able to contain so that the vehicle remains stable.

3.6.4.1 Experimental Setup

The MPC controllers presented in Section 3.6.3 have been tested through simulations and experiments on slippery surfaces. The experiments have been performed at a test center equipped with icy and snowy handling tracks. The MPC controllers have been tested on a passenger car, with a mass of 2050 kg and an inertia of 3344 kg/m^2. The controllers were run in a dSPACE^{TM} Autobox system, equipped with a DS1005 processor board and a DS2210 I/O board, with a sample time of 50 ms.

We used an Oxford technical solution (OTS) RT3002 sensing system to measure the position and the orientation of the vehicle in the inertial frame and the vehicle velocities in the vehicle body frame. The OTS RT3002, is housed in a small package that contains a differential GPS receiver, inertial measurement unit (IMU), and a digital signal processor (DSP). It is equipped with a single antenna to receive GPS information. The IMU includes three accelerometers and three angular rate sensors. The DSP receives both the measurements from the IMU and the GPS, utilizes a Kalman filter for sensor fusion, and calculate the position, the orientation, and the other states of the vehicle such as longitudinal and lateral velocities.

The car was equipped with an AFS system which utilizes an electric drive motor to change the relation between the hand steering wheel and road wheel angles (RWAs). This is done independently of the steering wheel position, thus the front RWA is obtained by summing the driver hand wheel position and the actuator angular movement. Both the hand wheel position and the angular relation between hand and

road wheels are measured. The sensor, the dSPACE^{TM} Autobox, and the actuators communicate through a CAN bus.

The autonomous steering test is initiated by the driver with a button. When the button is pushed, the inertial frame in Figure 3.34 is initialized as follows: the origin is the current vehicle position, the X and Y axes are directed as the current longitudinal and lateral vehicle axes, respectively. Such an inertial frame becomes also the desired path coordinate system. Once the initialization procedure is concluded, the vehicle executes the double-lane change maneuver.

During the experiment, the hand wheel may deviate from its center position. This is caused by the difficulty the driver can have in holding the steering still, which was needed to facilitate autonomous behavior with that particular test vehicle. In our setup this is treated as a small bounded input disturbance. Further, noise may affect the yaw angle measurement due to the single antenna sensor setup.

3.6.4.2 Presentation and Discussion of Results

In the next section the two MPC controllers will be presented. These controllers have been derived from the MPC problem formulations presented in Sections 3.6.3.1 and 3.6.3.2 and will be referred to as Controller A an Controller B:

- *Controller A:* Nonlinear MPC (Equations 3.46 and 3.47 with the following parameters
  - $T = 0.05\,\text{S}$, $H_p = 7$; $H_c = 3$; $\delta_{f,min} = -10^\circ$, $\delta_{f,max} = 10^\circ$, $\Delta\delta_{f,min} = -1.5^\circ$, $\Delta\delta_{f,max} = 1.5^\circ$, $\mu = 0.3$,

$$Q = \begin{pmatrix} 500 & 0 \\ 0 & 75 \end{pmatrix}, \quad R = 150$$

- *Controller B:* LTV MPC described in Section 3.6.3.2 with the following parameters
  - $T = 0.05\text{S}$, $H_p = 25$; $H_c = 1$, $H_u = H_p$, $\delta_{f,min} = -10^\circ$, $\delta_{f,max} = 10^\circ$, $\Delta\delta_{f,min} = -0.85^\circ$, $\Delta\delta_{f,max} = 0.85^\circ$, $\mu = 0.3$,

$$\alpha_{f_{min}} = -2.2\,\text{deg}, \quad \alpha_{f_{max}} = 2.2\,\text{deg}, \quad \alpha_{r_{min}} = -\infty, \quad \alpha_{\text{r}_{max}} = \infty$$

  - Weighting matrices $Q = \begin{pmatrix} 200 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$, $R = 5 \times 10^4$, $\rho = 10^3$.

Next the results obtained with the two controllers will be described and experimental results will be presented for each of them. For both controllers, the actual road friction coefficient μ was set manually and constant for each experiment depending on the road conditions. This choice was driven by the desire to focus on the study of the controller closed-loop performance independently from the μ estimation and its associated error and dynamics.

3.6.4.2.1 Controller A

The controller (Equations 3.46 and 3.47) has been implemented as a C-coded S-function in which the commercial NPSOL software package (Gill et al., 1998) is used for solving the nonlinear programming problem (Equation 3.46). NPSOL is a set of Fortran subroutines for minimizing a smooth function subject to constraints, which may include simple bounds on the variables, linear constraints, and smooth nonlinear constraints. NPSOL uses a sequential quadratic programming (SQP) algorithm, in which each search direction is the solution of a QP subproblem.

Limited by the computational complexity of the nonlinear programming solver and the hardware used, we could perform experiments at low vehicle speeds only. In fact, as the entry speed increases, larger prediction and control horizons are required in order to stabilize the vehicle along the path. Larger prediction horizons involve more evaluations of the objective function, while larger control horizons imply a larger optimization problem (Equation 3.46). In Table 3.1, we report simulation results summarizing the

TABLE 3.1 Maximum Computation Time of the Controllers A and B Performing a Double-Lane Change Maneuver at Different Vehicle Speeds

$\dot{x}$ [m/s]	Controller A Computation time (s)	Controller B Computation time (s)
10	0.15 ($H_p = 7, H_c = 2$)	0.03 ($H_p = 7, H_c = 3$)
15	0.35 ($H_p = 10, H_c = 4$)	0.03 ($H_p = 10, H_c = 4$)
17	1.3 ($H_p = 10, H_c = 7$)	0.03 ($H_p = 15, H_c = 10$)

maximum computation time required by Controllers A and B to compute a solution of the underlying nonlinear and QP problems, respectively, when the maneuver described in Section 3.6.4 is performed at different vehicle speeds. The selected control and prediction horizons in Table 3.1 are the shortest allowing the stabilization of the vehicle at each speed. The results have been obtained in simulation with a 2.0 GHz Centrino-based laptop running MATLAB® 6.5.

In Figure 3.36, the experimental results for a maneuver at 7 m/s are presented. In the upper plot of Figure 3.36b the dashed line represents the steering action from the driver (i.e., the input disturbance) that, in this test, is negligible. The actual RWA is the summation of the RWA from the MPC controller and the steering action from the driver. In the lower plot of Figure 3.36b, the NPSOL output flag is reported. In our tests the flag assumed the values 0, 1, 4, and 6. The value 0 is returned when an optimal feasible solution is found. The value 1 is returned when the solver does not converge to a feasible solution. The value 4 indicates that the limit on the iteration number has been reached and a feasible but nonoptimal solution has been found. The value 6 indicates that the solution does not satisfy the optimality conditions (Gill et al., 1998). In experimental tests, the solver often reaches the selected iteration limit and returns a suboptimal solution. Yet, because of the low vehicle speed, the performance associated to the suboptimal solution is very good.

Experimental tests at 10 m/s have shown that Controller A is not able to stabilize the vehicle. Data analysis has shown that the controller fails because the nonlinear solver does not converge to a feasible solution. The simulation results presented in Borrelli et al. (2005) have shown that, the NMPC controller is able to perform the maneuver at 10 m/s or higher speed if the solver maximum iteration number is not constrained.

3.6.4.2.2 Controller B

We recall that Controller B is based on the LTV MPC scheme derived from the problem (Equation 3.46) by replacing the nonlinear dynamics (Equation 3.46a) with the linearized dynamics (Equation 3.48) and adding the constraints (Equation 3.50). Controller B with the parameters defined here has been implemented as a C-coded S-function in MATLAB, using the QP solver routine available in The MathWorks Inc. (2005). Such routine implements the Dantzig–Wolfe's algorithm, has a good performance, and its source C code is publicly available. Next, we do not report the solver output flag since the solver always converged to an optimal solution.

We remark that the computation burden of this LTV MPC controller is reduced significantly compared to Controller A, as demonstrated by the computation times reported in Table 3.1.

Controller B has been tested in experiments where the vehicle enters the double-lane change at speeds ranging from 10 to 21 m/s. For the sake of brevity, only experimental tests at 21 m/s (i.e., the maximum entry speed) are reported next. Results of tests at lower speeds are summarized in Table 3.2. In Figure 3.37, the experimental results at the entry speed of 21 m/s are presented, showing that the controller is able to stabilize the vehicle, even though with large tracking errors. Detailed discussion of the experiment can be found in Falcone et al. (2007a). We observe that, in spite of the remarkably high speed on low friction surface, the controller is able to stabilize the vehicle. This is due to the controller capability of forcing the front tire slip angle within a range corresponding to a stable operating region (see the lower plot in Figure 3.37b). We remark that the tire slip angles violate constraint (Equation 3.50) in a small amount.

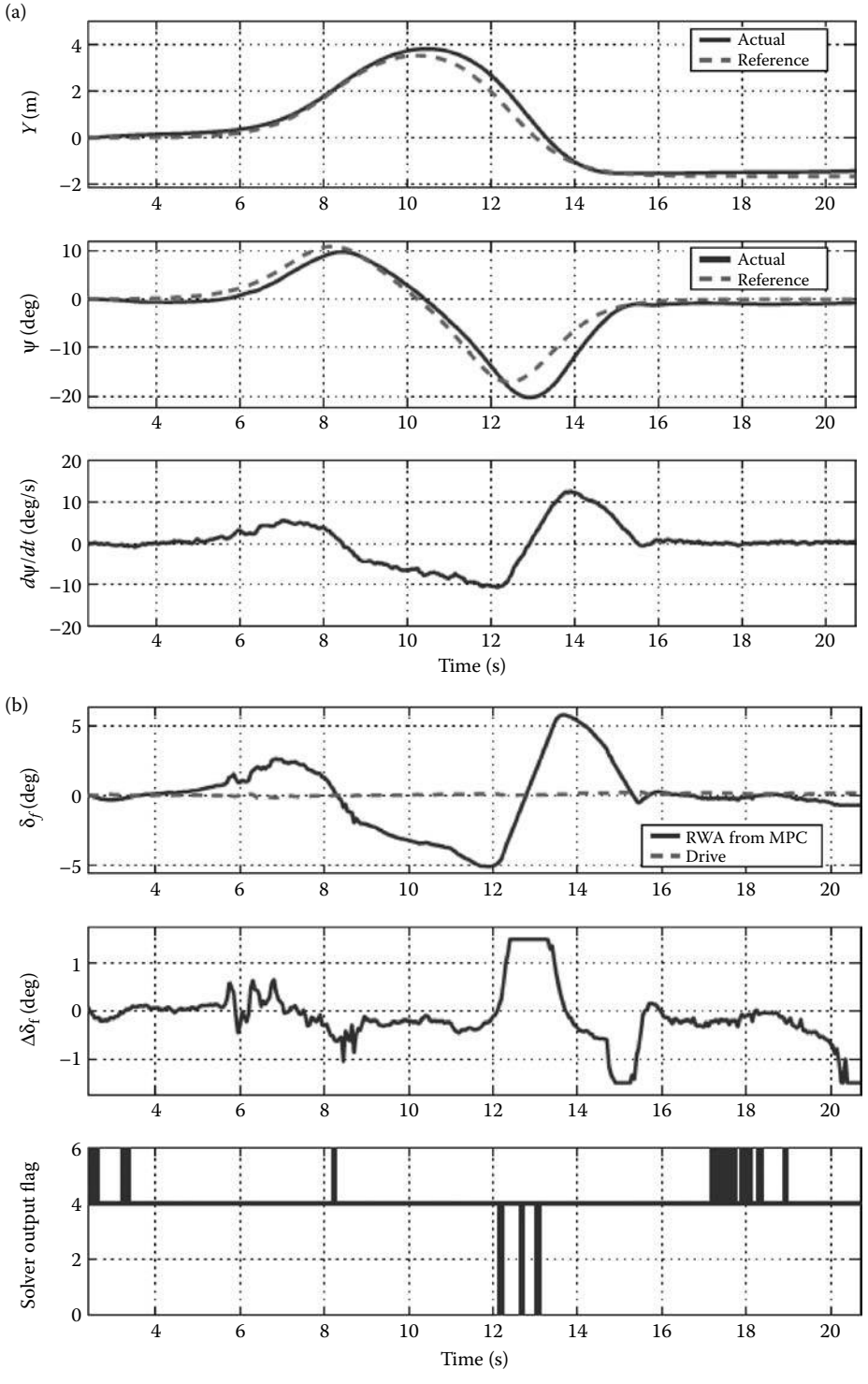

FIGURE 3.36 Experimental results at 7 m/s entry speed. Controller A described in Section 3.6.4.2.1

TABLE 3.2 Summary of Experimental Results, Controller B. RMS and Maximum Tracking Errors as Function of Vehicle Longitudinal Speed

$\dot{x}$ (m/s)	μ	ψ_{rms} (deg)	Y_{rms} (m)	ψ_{max} (deg)	Y_{max}(m)
10	0.2	9.52×10^{-1}	5.77×10^{-2}	13.12	3.28
21	0.2	1.037	7.66×10^{-2}	12.49	3.20

This is in agreement with the use of soft constraint and makes the system robust to driver's slight steering action.

3.6.5 Integrated Braking and Active Steering Control

Next, we extend the control design procedure presented in Section 3.6.3, in order to include individual braking at the four wheels. Three different low-complexity approaches are presented and validated in the path-following scenario described in Section 3.6.4. We provide a thorough and detailed discussion of experimental results obtained with the proposed control schemes and highlight the simplicity of the proposed scheme in orchestrating five different inputs with a minimal tuning effort. Moreover, we explain how complex countersteering maneuvers are naturally obtained as the consequence of an additional state and input constraint.

We adopt the control design methodology presented in the previous section. Discretize the vehicle model (Equation 3.34) with a sampling time T_s:

$$\xi(t+1) = f_{\mu(t)}^{4w,dt}(\xi(t), u(t)), \tag{3.52a}$$

$$u(t) = u(t-1) + \Delta u(t), \tag{3.52b}$$

where $u(t) = [\delta_f(t), T_{b_{f,l}}(t), T_{b_{f,r}}(t), T_{b_{r,l}}(t), T_{b_{r,r}}(t)]$, $\Delta u(t) = [\Delta\delta_f(t), \Delta T_{b_{f,l}}(t), \Delta T_{b_{f,r}}(t), \Delta T_{b_{r,l}}(t), \Delta T_{b_{r,r}}(t)]$. The output variables to be tracked are defined through the following output map:

$$\eta(t) = h(\xi(t)) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xi(t). \tag{3.53}$$

Moreover, we modify the cost function (Equation 3.45) as follows:

$$J(\xi(t), \Delta\mathcal{U}(t), \varepsilon) = \sum_{i=1}^{H_p} \|\eta(t+i) - \eta_{ref}(t+i)\|_Q^2 + \sum_{i=0}^{H_c-1} \|\Delta u(t+i)\|_R^2 + \sum_{i=0}^{H_c-1} \|u(t+i)\|_S^2 + \rho\varepsilon^2, \tag{3.54}$$

where, compared to Equation 3.45 (in Section 3.6.3.1), the third summand is added in order to penalize the braking torques.

We solve in receding horizon a nonlinear, in general nonconvex, constrained optimal control problem obtained from problem (Equation 3.46) by replacing the bicycle model (Equations 3.46a and 3.46b) with Equations 3.52 and 3.53 and minimizing the cost function (Equation 3.54).

The computational complexity of the NMPC controller obtained with the design procedure outlined in this section can be prohibitive for a real-time implementation. For this reason, alternative low-complexity approaches are presented in the next sections.

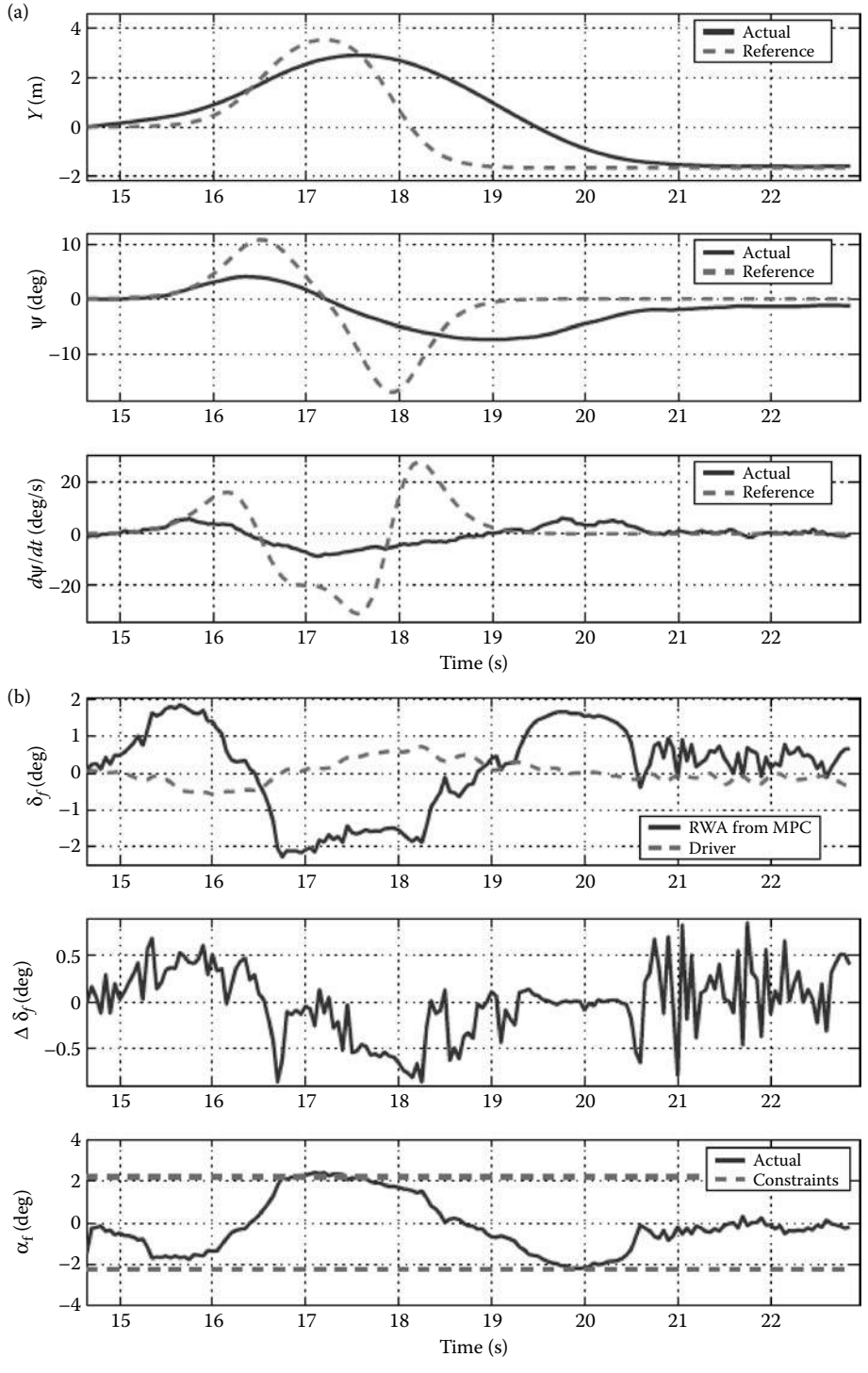

FIGURE 3.37 Experimental results at 21 m/s entry speed. Controller B.

3.6.5.1 LTV MPC Combined Active Steering and Braking Controller

Similarly to the LTV MPC steering controller presented in Section 3.6.3.2, an LTV MPC combined steering and braking controller can be designed by repeatedly linearizing the vehicle model (Equation 3.52) around the current state and the previous control input. That is, in Equation 3.49a, the nonlinear function (Equation 3.43) is replaced by Equation 3.52. In order to preserve vehicle stability, constraints (Equation 3.50) are added as well (Falcone et al., 2009).

3.6.5.2 Nonlinear MPC Combined Active Steering and Braking Controller Based on Low-Complexity Nonlinear Vehicle Models

Two different MPC combined active steering and braking controllers are presented next. In both approaches, a nonlinear constrained optimization problem is solved every time step. Complexity reduction, compared to the MPC controller based on the nonlinear vehicle model (Equation 3.52), is achieved by using simplified *nonlinear* vehicle models leading to smaller size optimization problems, yet including fundamental vehicle nonlinearities.

The first low complexity NMPC approach, hereafter referred to as *Two Actuators MPC*, is sketched in Figure 3.38a. An MPC problem is formulated by replacing the model (Equation 3.52) with a discrete-time version of model (Equation 3.42). Every time step, the NMPC controller computes a steering command δ_f and a desired braking yaw moment M. The braking logic in Figure 3.38a then calculates the braking torques at four individual wheels in order to generate the desired yaw moment M, based on the current vehicle state. An example of this braking logic, next referred to as *Two-Actuator Algorithm* is detailed in Falcone et al. (2008b). In particular, in order to induce the desired braking yaw moment with minimum

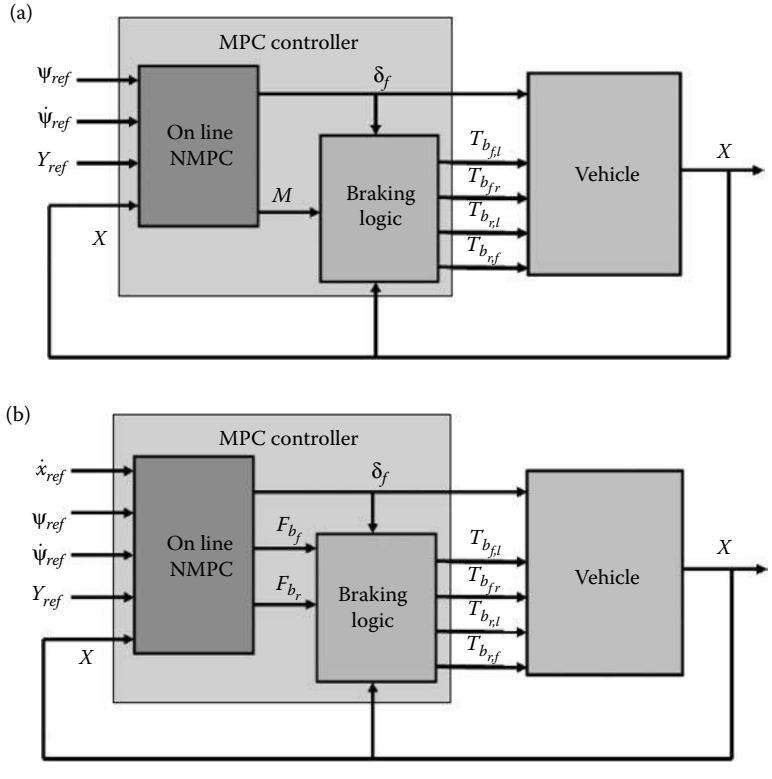

FIGURE 3.38 Two different approaches to the integrated VDC problem.

longitudinal dynamics effect, *Two-Actuator Algorithm* implements a single wheel braking logic. The *Two-Actuator Algorithm* is based on the following well-known results:

- Outside wheel braking induces understeer while inside wheel braking induces oversteer.
- Left/right brake distribution is more effective in steering the vehicle than front/rear distribution (Motoyama et al., 1992).
- Braking at the rear inside corner is most effective in inducing an oversteer yaw moment, and braking at the front outside corner is most effective to induce an understeer yaw moment (Tseng et al., 1999; Bahouth, 2005; Bedner et al., 2007).

The second MPC approach, next referred to as *Three-Actuator MPC*, is sketched in Figure 3.38b. This is based on the simplified nonlinear vehicle model (Equation 3.36). Every time step, the steering command and the braking forces at the left and right side of the vehicle are computed. As for the Two-Actuator MPC controller, a braking logic can be deployed to compute individual braking torques at each wheel, based on the vehicle current state. An example for this braking logic can be found in Falcone et al. (2006a) and is next referred to as *Three-Actuator Algorithm*.

3.6.6 Double-Lane Change with Active Steering and Differential Braking

The combined steering and braking MPC controllers described in Section 3.6.5 have been implemented to autonomously perform a double-lane change in the scenario described in Section 3.6.4.

Next we present the experimental results only for the LTV MPC combined braking and steering controller in Section 3.6.5.1. The Two- and Three-Actuator controllers, presented in Section 3.6.5.2, have been tested in experiments and the results show that they successfully stabilize the vehicle along the desired path up to 60 kph (Falcone et al., 2010).

The LTV MPC controller has been implemented with the following parameters:

- *Sampling time:* $T = 0.05$ s.
- *Horizons:* $H_p = 15, H_c = 1, H_u = 2$.
- *Bounds:*
  - $\delta_{f,min} = -10^\circ,\ \delta_{f,max} = 10^\circ,\ \Delta\delta_{f,min} = -0.85$ deg, $\Delta\delta_{f,max} = 0.85$ deg.
  - $T_{b_{\star,\bullet min}} = 0$ N m, $T_{b_{\star,\bullet max}} = 600$N m, $\Delta T_{b_{\star,\bullet} min} = -58.33$ N m, $\Delta T_{b_{\star,\bullet max}} = 58.33$ Nm.
  - $\alpha_{\star_{min}} = -2.5$ deg, $\alpha_{\star_{max}} = 2.5$ deg.
- *Friction coefficient:* $\mu = 0.3$.
- *Weighting matrices:*
  - $Q \in \mathcal{R}^{4\times4}$ with $Q_{11} = 1, Q_{22} = 10, Q_{33} = 1, Q_{44} = 30$, and $Q_{ij} = 0$ for $i \neq j$.
  - $R \in \mathcal{R}^{5\times5}$ with $R_{ij} = 10$ for $i = j$ and $R_{ij} = 0$ for $i \neq j$.
  - $S \in \mathcal{R}^{5\times5}$ with $S_{ij} = 10^{-1}$ for $i = j$ and $S_{ij} = 0$ for $i \neq j$.
  - $\rho = 10^5$.

The bounds $\Delta\delta_{f,\#}$ and $\Delta T_{b_{\star,\bullet}\#}$, with $\# = \{\min, \max\}$, are derived from the steering and braking rate, respectively, of the real actuators. The bounds $\alpha_{\star_\#}$ are selected in order to force the vehicle to operate in the linear region of the tire characteristic. The horizons H_p, H_c, and H_u are selected by trading off performance and computational complexity. In particular, the horizons H_p and H_c are the largest allowing the real-time execution of the controller on the rapid prototyping platform described in Falcone et al. (2009). The constraint horizon H_c and the weighting matrices Q, R, and S have been tuned through extensive simulations.

The road friction coefficient μ in Equation 3.52 was set manually and constant for each experiment depending on the road conditions.

Figures 3.39 and 3.40 show the experimental results when the vehicle enters the double-lane change with a longitudinal speed of 50 and 70 kph, respectively. In Figures 3.39b and 3.40b, the braking torques at the four wheels at 50 and 70 kph, respectively, are reported. In particular, the solid lines represent the

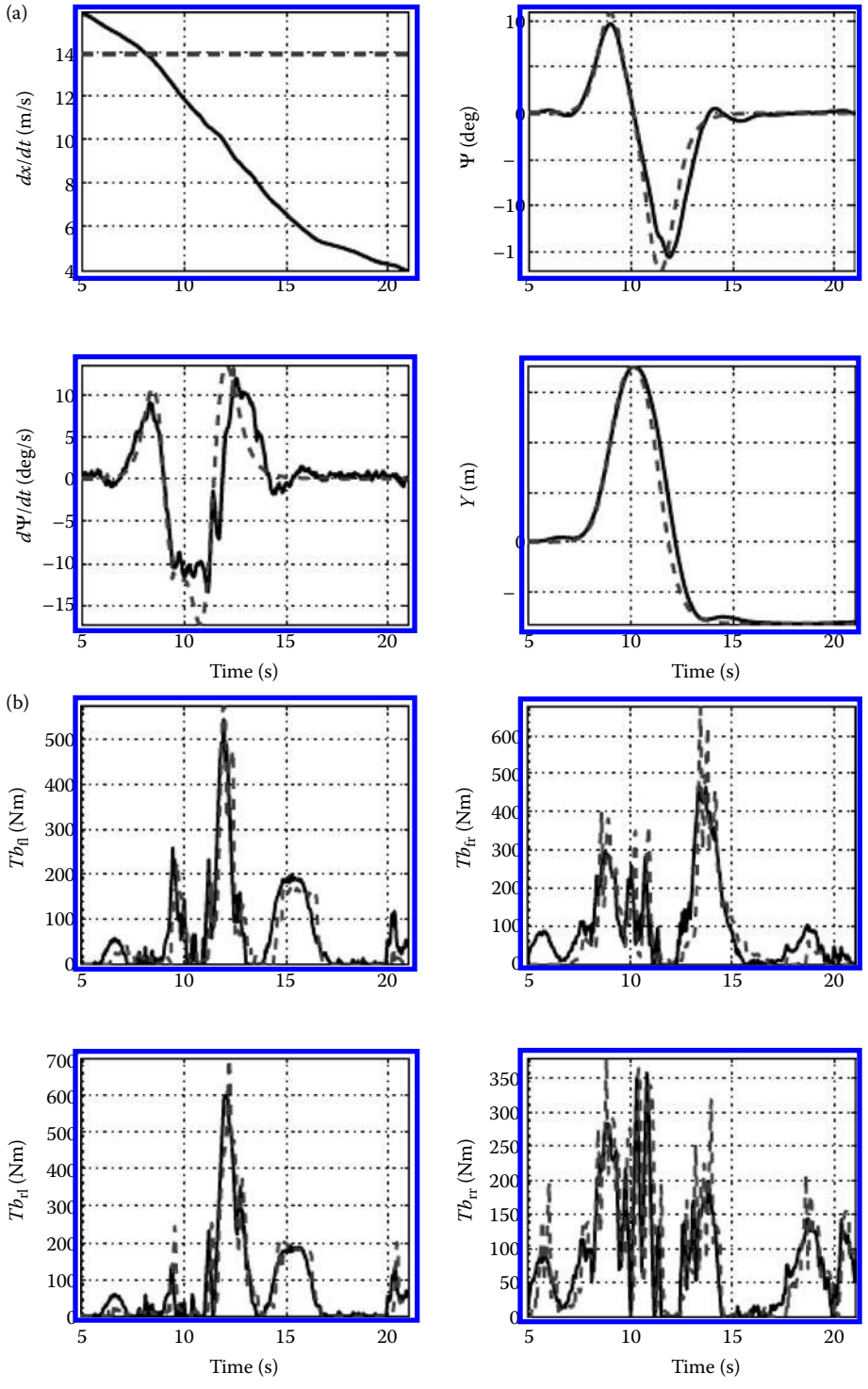

FIGURE 3.39 Experimental results at 14 m/s entry speed.

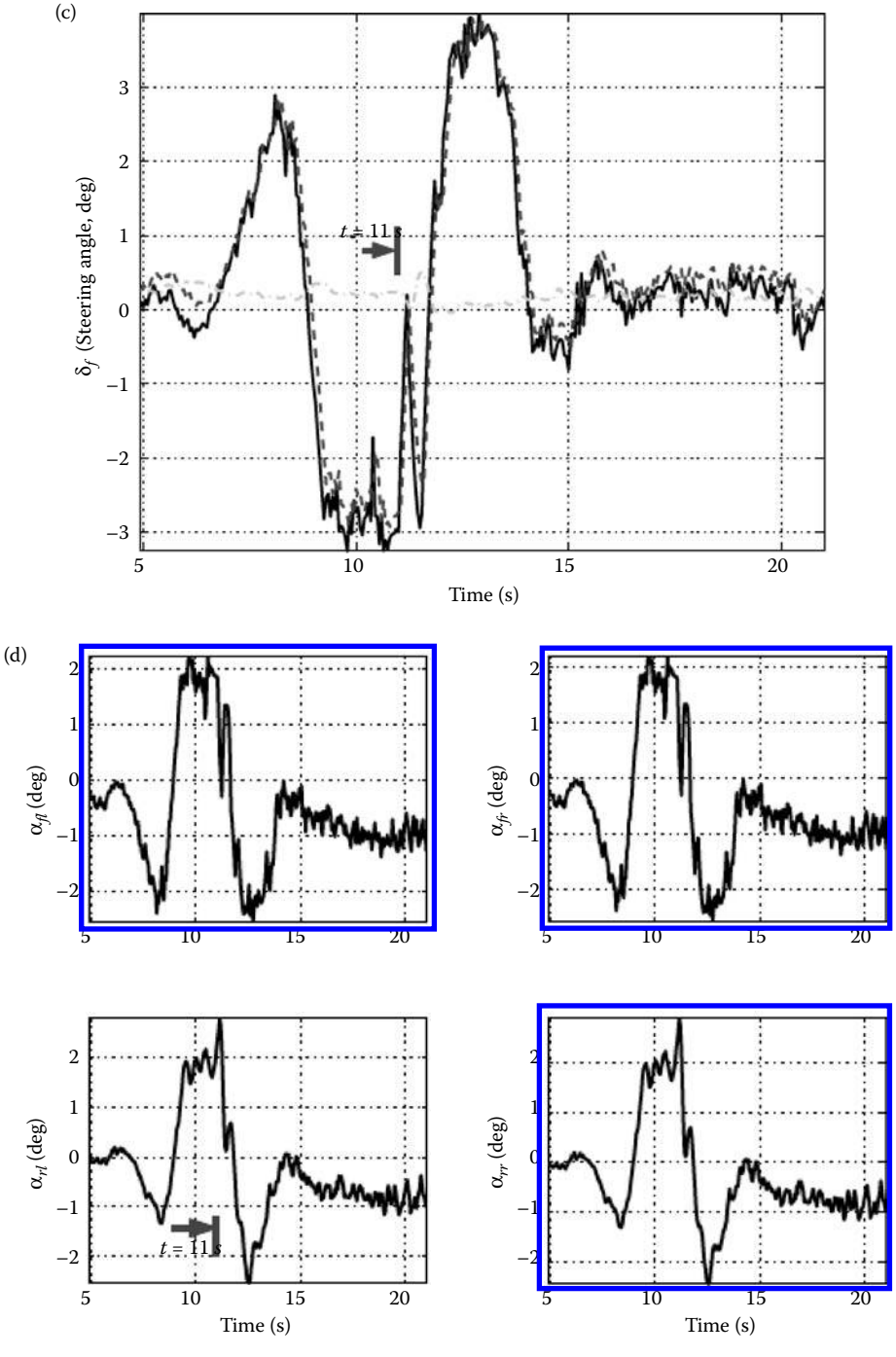

FIGURE 3.39 Continued.

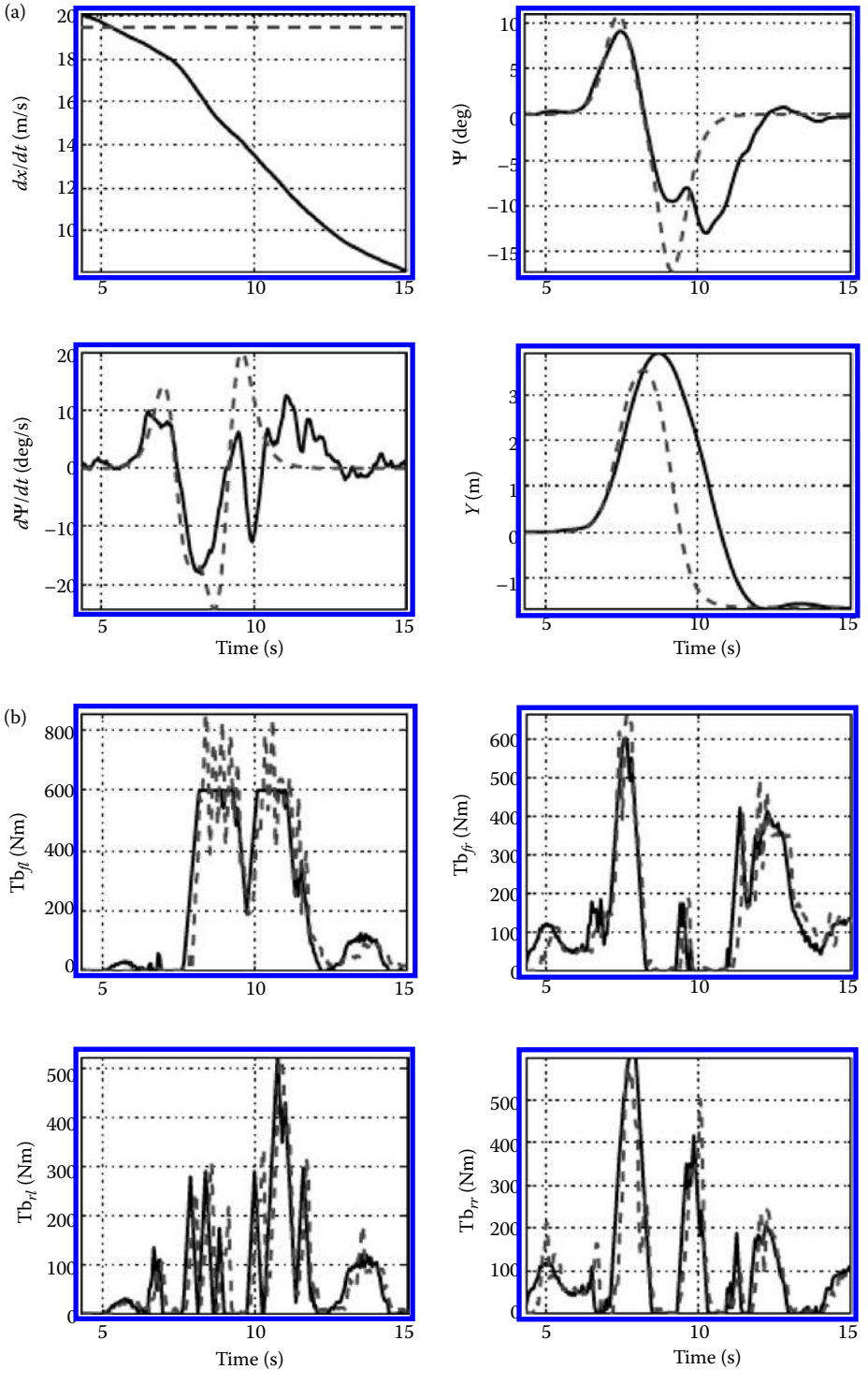

FIGURE 3.40 Experimental results at 19 m/s entry speed.

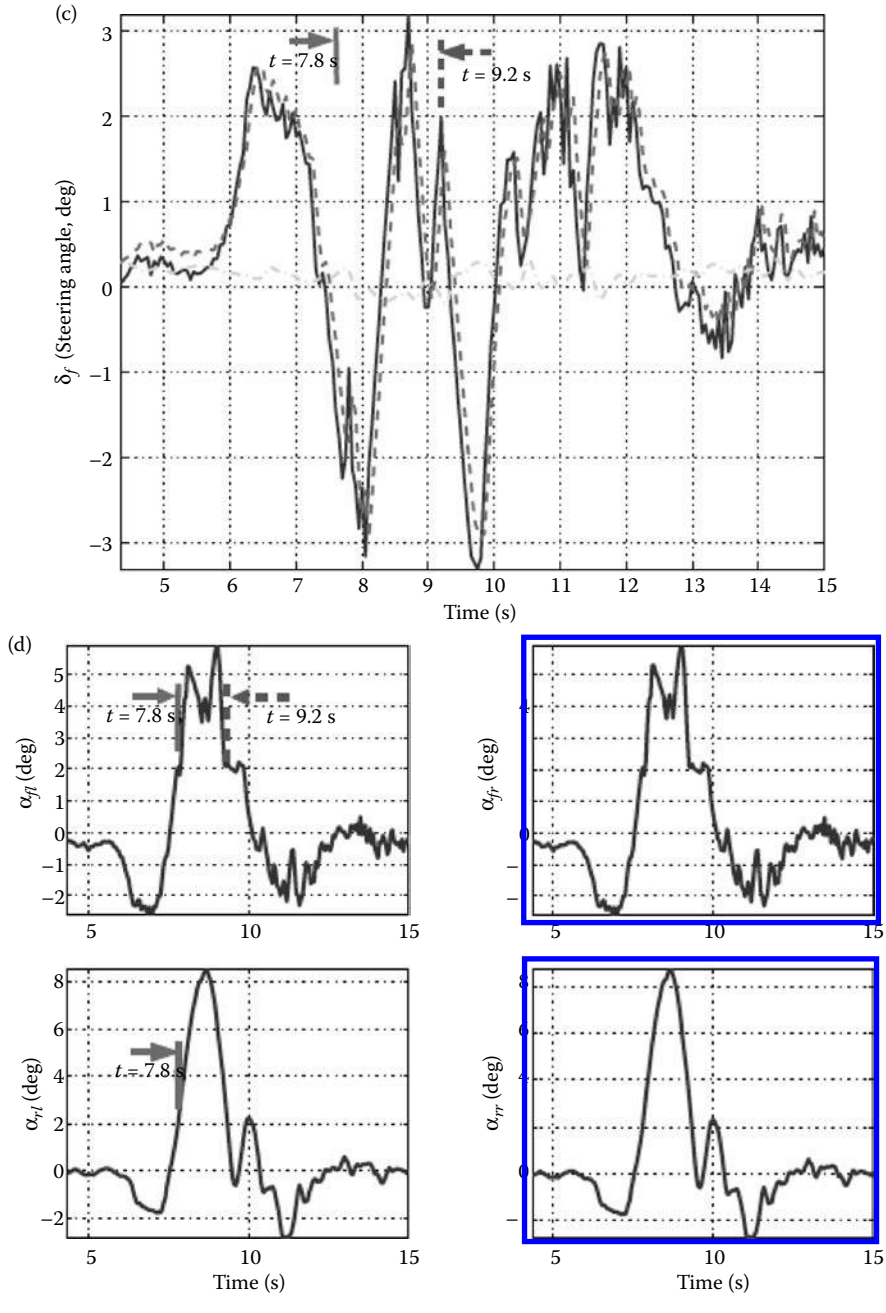

FIGURE 3.40 Continued.

desired torques computed by the controller, while the torques delivered by the braking system are plotted with dashed lines. In Figures 3.39c and 3.40c, the steering angles (desired in solid lines and actual in dashed lines) in experiments at 50 and 70 kph, respectively, are reported. We also observe the steering angle from the driver (dash-dotted lines) presented themselves as small driver disturbance during the path-following test that was to approximate an autonomously controlled steering robot vehicle. For the

TABLE 3.3 Experimental Results. RMS and Maximum Tracking Errors as Function of Vehicle Longitudinal Speed

$\dot{x}_{ref}$ (m/s)	ψ_{rms} (deg)	$\dot{\psi}_{rms}$ (deg/s)	Y_{rms} (m)	ψ_{max} (deg)	$\dot{\psi}_{max}$ (deg/s)	Y_{max} (m)	$\dot{x}_{max}$ (m/s)
14	1.51	3.15	0.24	5.19	15.91	0.91	7.32
19.4	3.93	7.29	1.22	10.91	26.99	3.42	11.30

other plots the same line styles have been used. The maximum and root mean-squared (RMS) errors are summarized in Table 3.3.

Next, we comment the above results in order to highlight (1) the coordination between steering and braking, (2) the execution of a countersteering maneuver in order to recover the control of the vehicle.

In order to discuss point (1), we recall that angles are assumed positive counterclockwise and braking the wheels at left and right sides generate positive and negative overall yaw moment, respectively. Consider the results at 50 kph in Figure 3.39. In Figures 3.39b and c, we observe that at around 12 s, a left side braking and a right to left steering start in order to track the yaw rate reference. At almost 13 s, the braking at the left side is stopped and, simultaneously, the braking at the right side, as well as a steering to the right, are started. A similar behavior is observed in experiments at 70 kph. Nevertheless, because of a more oscillatory behavior, the coordination between steering and braking is less clear.

Before discussing point (2), we recall that countersteering is a complex maneuver usually performed by skilled drivers in order to recover the control of the vehicle in emergency situations. In particular, due to excessive tire slip angles the vehicle might operate in regions of the tire characteristic where the lateral tire forces are decreasing functions of the tire slip angles. In such cases the vehicle might deviate from its nominal behavior and even become unstable. In order to stabilize the vehicle, a skillful driver can steer in the opposite direction with respect to the intended trajectory, in such a way to limit the tire slip angles. We observed that, with our control problem formulation, countersteering is performed, thanks to constraints (Equation 3.50) on the tire slip angles. Consider the experimental results in Figure 3.39, in Figure 3.39d we observe that, at 11 s, the rear tire slip angles exceed the upper bound, that is, 2.5 deg, and the controller responded with a brief counter steering (Falcone et al., 2009). Consider a small-angle approximation of Equation 3.33, written for the rear axle

$$\alpha_r = \frac{\dot{y} - b\dot{\psi}}{\dot{x}}. \tag{3.55}$$

Since $\partial\alpha_r/\partial\dot{\psi} < 0$, the steering angle (and hence the yaw rate) is suddenly increased in order to decrease the rear tire slip angle. Afterward the rear tire slip angles returns within the bounds and the steering angle decreases again in order to recover the path following.

Similar arguments explain the controller's behavior in the experiment at 70 kph where, between 7.8 and 9.2 s, both front and rear tire slip angles exceed the upper bound (see Figure 3.40d). In this case, by increasing the steering angle, front and rear tire slip angles are simultaneously decreased. Afterwards, the controllers steers to the right in order to recover the path following.

3.6.7 Summary

In this section, we have presented MPC approaches for autonomous path following via AFS. We have then further extended the presented approaches to coordinate AFS and individual wheel braking. Experimental results have been presented for the proposed algorithms. The computational aspects of the MPC design for the considered path-following application have been emphasized and different low complexity MPC schemes have been presented.

In the proposed LTV MPC algorithm, the steering and braking are computed as a solution of a quadratic program. Such a problem is formulated by linearizing, at each time step, the nonlinear vehicle model around the current state and the previous control input. The integrated steering and braking MPC

algorithm has been implemented on a dSPACE^{TM} rapid prototyping system with a sampling time of 50 ms in order to be experimentally validated. Experimental validation consisted of autonomous path-following tests, where the vehicle performed double-lane change maneuvers at high speed on snowy roads. Experimental results of tests at 50 and 70 kph have been shown and discussed. The presented results demonstrated three remarkable achievements: the proposed algorithm (1) coordinates the use of steering and braking in a systematic way to achieve the control objectives, (2) is able to stabilize the vehicle when working in wide operating regions of the tire characteristics where model nonlinearities become relevant, (3) systematically reproduces complex countersteering behavior, usually performed by skilled professional drivers in stabilizing the vehicle, as a natural consequence of MPC methodology with its soft constraints on the tire slip angles. Moreover, we have presented two alternative low-complexity MPC approaches for online solving nonlinear constrained optimization problems. The presented approaches are general and can be used to solve any vehicle stability control problem. In particular, further extensions of the algorithms presented in this section might include further actuators, such as active differentials and active and semiactive suspensions (Giorgetti, 2006), toward a fully active chassis control problem. Nevertheless, computational complexity is a serious barrier for real-time implementation, since the size and the complexity of the MPC controller increases as further actuators are added.

3.7 Concluding Remarks

Vehicle controls are having an increasingly more and more important role in today's automotive industry. The current article summarizes several such control applications, ranging from vehicle suspensions and electronic differential to active steering and ESC, and their extensions. In all of these cases, the tire is the central and critical element connecting vehicles with the road and transferring associated forces. Based on the experiences accumulated so far, the applications utilizing tire longitudinal force have been the most established and widespread. This includes ABS, TCS, and various aspects of stability controls and their extensions. On the other hand, the production applications based on tire lateral and vertical or normal forces are not as widespread in contemporary vehicles. This includes active front steer, rear wheel steer/four wheel steer, semiactive or continuously controlled damping, and load-leveling suspensions.

As has been observed in most vehicle control applications, the modeling and the estimation of the controlled plant are critical elements for the successful implementation of desired control concepts. Thus, the present chapter devotes a significant portion to address these important aspects of the new control design process. This process will be further facilitated by the prevailing trend that future vehicles will be equipped with a new generation of actuators and sensors. On the actuator side, this includes various forms of active steering as well as fully active and semiactive suspensions. On the sensor side, this includes cameras, GPS, and vehicle-to-vehicle/infrastructure communications. This will provide many additional opportunities in refining/optimizing vehicle handling and ride performance, and overall vehicle controllability, stability, and safety, while at the same time creating new and exciting functions.

References

Ackermann, J., Advantages of active steering for vehicle dynamics control. *29th Conference on Decision and Control*, Honolulu, HI, 1990.

Ackermann, J., Odenthal, D., and Bünte, T., Advantages of active steering for vehicle dynamics control. *32nd International Symposium on Automotive Technology and Automation*, Vienna, Austria, 1999.

Ackermann, J. and Sienel, W., Robust yaw damping of cars with front and rear wheel steering. *IEEE Trans. Control Systems Technology*, 1(1):15–20, 1993.

Assadian, F. and Hancock, M., A comparison of yaw stability control strategies for the active differential. *Proc. of 2005 IEEE International Symposium on Industrial Electronics*, Dubrovnik, Croatia, 2005.

Bahouth, G., Real world crash evaluation of vehicle stability control (VSC) technology. *49th Annual Proceedings, Association for the Advancement of Automotive Medicine*, September 11–14, Boston, 2005.

Bakker, E., Nyborg, L., and Pacejka, H.B., Tyre modeling for use in vehicle dynamics studies. *SAE paper No. 870421*, 1987.

Bedner, E., Fulk, D., and Hac, A., Exploring the trade-off of handling stability and responsiveness with advanced control systems. *SAE*, SAE 2007-01-0812, 2007.

Bernard, J.E. and Clover, C.L., Tire modeling for low-speed and high-speed calculations. *SAE paper No. 950311*, 1995.

Borrelli, F., Falcone, P., Keviczky, T., Asgari, J., and Hrovat, H., MPC-based approach to active steering for autonomous vehicle systems. *Int. J. Vehicle Autonomous Systems*, 3(2/3/4):265–291, 2005.

Borrelli F., Bemporad A., Fodor M., and Hrovat D., An MPC/hybrid system approach to traction control, *IEEE Transactions on Control Systems Technology*, 14(3), 541–552, 2006.

Brown, T. and Rhode, D., Roll over stability control for an automotive vehicle, *US Patent 6263261*, 2001.

Dang, J., Statistical analysis of the effectiveness of electronic stability control (ESC) systems—final report, *NHTSA Technical Report,* DOT HS 810 794, 2007.

Deur, J., Asgari, J., Hrovat, D., A 3D brush-type dynamic tire friction model, *Vehicle System Dynamics*, 40, 133–173, 2004.

Deur, J., Hancock, M., and Assadian, F., Modeling and analysis of active differential kinematics, *CD Proc. 2008 ASME Dynamic Systems and Control Conference*, Ann Arbor, MI, 2008a.

Deur, J., Hancock, M., and Assadian, F., Modeling and of active differential dynamics, *DVD Proc. IMECE2008, ASME Paper No. 2008–69248*, Boston, MA, 2008b.

Deur, J., Kranjèeviæ, N., Hofmann, O., Asgari, J., and Hrovat, D., Analysis of lateral tyre friction dynamics, *Vehicle System Dynamics*, 47(7), 831–850, 2009.

Deur, J., Ivanoviæ, V., Pavkoviæ, D., Asgari, J., Hrovat, D., Troulis, M., and Miano, C., On low-slip tire friction behavior and modeling for different road conditions, *CD Proc. of XIX IAVSD Symposium*, Milan, Italy, 2005.

Deur, J., Pavkovic, D., Hrovat, D., and Burgio, G., A model-based traction control strategy non-reliant on wheel slip information, in: *CD Proc. of 21st IAVSD International Symposium on Dynamics of Vehicles on Roads and Tracks*, Stockholm, Sweden, 2009.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D., Low complexity MPC schemes for integrated vehicle dynamics control problems. *9th International Symposium on Advanced Vehicle Control*, Kobe, Japan, 2006a.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D., A real-time model predictive control approach for autonomous active steering. *Nonlinear Model Predictive Control for Fast Systems*, Grenoble, France, 2006b.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D., Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control System Technol.*, 15(3):566–580, 2007a.

Falcone, P., Borrelli, F., Tseng, H.E., Asgari, J., and Hrovat, D., Integrated braking and steering model predictive control approach in autonomous vehicles. *Fifth IFAC Symposium on Advances of Automotive Control*, August 20–22, Aptos, CA, 2007b.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D., Linear time varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *Int. J. Robust Nonlinear Control*, 18:862–875, 2008a.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D., MPC-based yaw and lateral stabilization via active front steering and braking. *Vehicle System Dynamics*, 46:611–628, 2008b.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H.E., and Hrovat, D., Linear time varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *Int. J. Vehicle Autonomous Systems*, 7(3/4):292–309, 2009.

Falcone, P., Borrelli, F., Tseng, H.E., and Hrovat, D., On low complexity predictive approaches to control of autonomous vehicles, *Automotive Model Predictive Control of Lecture Notes in Control and Information Sciences*, 195–210. Springer, Berlin, 2010.

Fennel, H. and Ding, E., A model-based failsafe system for the continental TEVES electronic-stability-program, *SAE Automotive Dynamics and Stability Conference*, *SAE 2000-01-1635*, 2000.

Ferguson, S., The effectiveness of electronic stability control in reducing real-world crashes: A literature review, *Traffic Injury Prevention*, 8(4), 329–338, 2007.

Fukada, Y., Estimation of vehicle side-slip with combination method of model observer and direct integration, *4th International Symposium on Advanced Vehicle Control (AVEC)*, Nagoya, Japan, September 14–18, 201–206, 1998.

Fukada, Y., Slip-angle estimation for vehicle stability control, *Vehicle System Dynamics*, 32(4), 375–388, 1999.

Gill, P., Murray, W., Saunders, M., and Wright, M., *NPSOL–Nonlinear Programming Software*. Stanford Business Software, Inc., Mountain View, CA, 1998.
Gillespie, T., *Fundamentals of Vehicle Dynamics*, Society of Automotive Engineers Inc, 1992.
Giorgetti, N., Bemporad, A., Tseng, H. E., and Hrovat, D., Hybrid model predictive control application towards optimal semi-active suspension. *Int. J. Control*, 79 (5), 521–533, 2006.
Greenwood, D., *Principle of Dynamics*, 2nd edn., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1998.
Hac, A., Evaluation of two concepts in vehicle stability enhancement systems, *Proceedings of 31st ISATA*, *Automotive Mechatronics Design and Engineering*, Vienna, 205–212, 1998.
Hac, A., and Simpson, M. D., Estimation of vehicle sideslip angle and yaw rate, *SAE World Congress*, Detroit, MI, USA, March 6–9. *SAE 2000-01-0696*, 2000.
Helton, W., *Extending H^{∞} Control to Nonlinear Systems: Control of Nonlinear Systems to Achieve Performance Objectives*, Society for Industrial and Applied Mathematics, 1999.
Hrovat, D., Survey of advanced suspension developments and related optimal control applications, *Automatica*, 33(10), 1781–1817, 1997.
Hrovat, D., Asgari, J., and Fodor, M., Automotive mechatronic systems, Chap. 1 of *Mechatronic Systems Techniques and Applications, Vol. 2: Transportation and Vehicular Systems*, C.T. Leondes, Ed., Gordon and Breach international series in engineering, technology and applied science, The Netherland. 2000.
Hrovat, D. and Tran, M., Method for controlling yaw of a wheeled vehicle based on under-steer and over-steer containment routines, *US5576959*, 1996.
Kaminaga, M. and Naito, G., Vehicle body slip angle estimation using an adaptive observer, *4th International Symposium on Advanced Vehicle Control (AVEC),* Nagoya, Japan, Sep. 14–18, 207–212, 1998.
Keviczky, T., Falcone, P., Borrelli, F., Asgari, J., and Hrovat, D., Predictive control approach to autonomous vehicle steering. *Proc. Am. Contr. Conf.*, Minneapolis, Minnesota, 2006.
Kim, D., Kim, K., Lee, W., and Hwang, I., Development of Mando ESP (electronic stability program), *SAE 2003-01-0101*, 2003.
Lakehal-ayat, M., Tseng, H. E., Mao, Y., and Karidas, J., Disturbance observer for lateral velocity estimation, *JSAE 8th International Symposium on Advanced Vehicle Control (AVEC)*, Taipei, Taiwan, Aug. 20–24, 2006.
Liu, C. and Peng, H., A state and parameter identification scheme for linearly parameterized systems, *ASME J. Dynamic Systems, Measurement and Control*, 120(4), 524–528, 1998.
Lu, J., Messih, D., Salib, A., and Harmison, D., An enhancement to an electronic stability control system to include a rollover control function, *SAE 2007-01-0809*, 2007a.
Lu, J., Messih, D., and Salib, A., Roll rate based stability control—The roll stability control™ system. ESV 07-136, *Proceedings of the 20th Enhanced Safety of Vehicles Conference*, Lyon, France, 2007b.
Lu, J., Meyers, J., Mattson, K., and Brown, T., Wheel lift identification for an automotive vehicle using passive and active detection, US Patent 7132937, 2006.
Margolis, D.L. and Asgari, J., Multipurpose models of vehicle dynamics for controller design. *SAE Technical Papers*, 1991.
Maurice, J.P., Short wavelength and dynamic tyre behaviour under lateral and combined slip conditions. Ph.D. Thesis, TU Delft, Netherlands, 2000.
Mayne, D.Q. and Michalska, H., Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Automatic Control*, 38(11):1623–1633, 1993.
Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M., Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
Motoyama, S., Uki, H., Isoda, L., and Yuasa, H., Effect of traction force distribution control on vehicle dynamics. *Proc. 1992 Int. Symp. Advanced Vehicle Control (AVEC '92)*, 447–451, Yokohama, Japan, 1992.
Nishio, A., Tozu, K., Yamaguchi, H., Asano, K., and Amano, Y., Development of vehicle stability control system based on vehicle sideslip angle estimation, *SAE 2001-01-0137*, 2001.
Pacejka, H.B. and Sharp, R.S., Shear force development by pneumatic tyres in steady state conditions: A review of modelling aspects. *Vehicle System Dynamics* 20, 121–176, 1991.
Pacejka, H.B., *Tyre and Vehicle Dynamics*. Butterworth-Heinemann, Oxford, 2002.
Palkovics, L., Semsey, A., and Gerum, E., Rollover prevention system for commercial vehicles—additional sensorless function of the electronic brake system, *Vehicle System Dynamics*, 32, 285–297, 1999.
Pillutti, T., Ulsoy, G., and Hrovat, D., Vehicle steering intervention through differential braking, *Proceedings of American Control Conference*, 1667–1671, Seattle, WA, 1995.
Sawase, K. and Sano, Y., Application of active yaw control to vehicle dynamics by utilizing driving/braking force, *ISAE Review*, 20, 289–295, 1999.
Sawase, K., Ushiroda, Y., and Miura, T., Left-right torque vectoring technology as the core of super all wheel drive control (S-AWC), *Mitsubishi Motors Technical Review*, 18, 16–23, 2006.

Sierra, C., Tseng, E., Jain, A., and Peng, H., Cornering stiffness estimation based on lateral vehicle dynamics, *Vehicle System Dynamics*, 44(1) 24–88, 2006.
The MathWorks Inc., *Model Predictive Control Toolbox*, 2005.
Tseng, H.E., Ashrafi, B., Madau, D., Brown, T.A., and Recker, D., The development of vehicle stability control at Ford. *IEEE-ASME Trans. Mechatronics*, 4(3):223–234, 1999.
Tseng, H.E., Dynamic estimation of road bank angle. *Proceedings of the 5th International Symposium on Advanced Vehicle Control (AVEC)*, Ann Arbor, MI, 421–428, 2000.
Tseng, H.E., Dynamic estimation of road bank angle, *Vehicle System Dynamics*, 36(4–5), 307–328, 2001.
Tseng, H.E., A sliding mode lateral velocity observer, *6th International Symposium on Advanced Vehicle Control (AVEC),* Hiroshima, Japan, Sep. 9–13, 387–392, 2002.
Tseng, H.E. and Xu, L., Robust model-based fault detection for roll rate sensor, *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, pp. 1968–1973, 2003.
Tseng, H.E., Asgari, J., Hrovat, D., Van Der Jagt, P., Cherry, A., and Neads, S., Evasive maneuvers with a steering robot. *Vehicle System Dynamics*, 43(3):197–214, 2005.
Ungoren, A., Peng, H., and Tseng, H.E., A study on lateral speed estimation methods, *International Journal of Vehicle Autonomous System*, 2(1/2) 126–144, 2004.
Vehicle Dynamics Standards Committee. *Automotive Stability Enhancement Systems,* SAE Standards, Document number J2564, 2004.
Xu, L. and Tseng, H. E., Robust model-based fault detection for a roll stability control system, *IEEE Transactions on Control Systems Technology*, 15(3), 519–528, 2007.
Zanten, A., Bosch ESP systems: 5 years of experience, *SAE Automotive Dynamics and Stability Conference*, *SAE 2000-01-1633*, 2000.
Zegelaar, P.W.A., The dynamic response of tyres to brake torque variations and road unevennesses. Ph.D. Thesis, TU Delft, Netherlands, 1998.
Zhou, K. and Doyle, J., *Essentials of Robust Control*, Prentice-Hall, Englewood Cliffs, NJ, 1998.

4

Model-Based Supervisory Control for Energy Optimization of Hybrid-Electric Vehicles

Lino Guzzella
Swiss Federal Institute of Technology

Antonio Sciarretta
IFP Energies Nouvelles

4.1 Introduction

4.1.1 Motivation

The demand for individual mobility is inexorably increasing and the number of automobiles operated worldwide will continue to grow for the next 20–30 years (Guzzella, 2009). Unfortunately, fossil primary energy sources are limited, and particularly in the case of crude oil, the era of its cheap and plentiful supply is coming to an end. Clearly, these two developments can be balanced only by a combination of several changes in our individual mobility system. One central part of it will be the development of substantially more fuel-efficient vehicle propulsion systems.*

* It is clear that this will not suffice. Other necessary measures are a change to smaller and lighter vehicles, an increase in the utilization of alternative fuel sources, and a change in the mobility patterns.

One very promising approach toward achieving this objective is the development of hybrid-electric powertrains. On the one side, such propulsion systems can improve the rather poor thermal efficiency of standard internal combustion engines. On the other side, they pave the way to a partial electrification of individual mobility by combining short-range purely electric travel with long-range driving capability.

4.1.2 Problem Statement and Limitations

This chapter introduces methods and tools that are useful for the analysis and optimization of the energy consumption of hybrid-electric powertrains. After the introduction of the appropriate modeling approaches, first noncausal control algorithms and then causal control algorithms are presented. The first set of control algorithms is very useful as a benchmark for the second set, and noncausal algorithms also form the basis for a possible approach to the optimal design of hybrid-electric powertrains, with respect to both structures and parameters.

All controllers discussed below are working on the supervisory level, that is, they only control the flow of energies between the different nodes in the powertrain system. The commands issued by these control systems are reference values for lower-level controllers that have to interact with the actual hardware components. Many important aspects of powertrain dynamics can only be handled on those lower levels. Particularly, drivability (torque smoothness) and pollutant emissions of the thermal engine are difficult problems, which cannot be included easily in the supervisory control system. Fortunately, the timescales of these two sets of problems are substantially different, such that the hierarchical approach proposed in this chapter yields good results.

4.2 Modeling the Energy Flows in Hybrid-Electric Vehicles

4.2.1 Modeling Requirements

To be able to achieve the objectives stated in the introduction, the models proposed in this section must have the following properties:

- The numerical effort required to simulate complete powertrain systems must be low.
- The individual modules must be scalable, that is, the behavior of the corresponding devices must be represented by models that have a parametric description of the "size" of the devices.
- The interfaces of all modules must permit an arbitrary connection between all elements.

Equations describing the behavior of the most relevant system components that satisfy these requirements are presented below. More details can be found in Guzzella and Sciarretta (2007).

In the context of this chapter, it is important to distinguish between *forward* and *backward* models. While the first class of models reflects the accepted causality present in a specific physical system, the second class of models inverts that causality. A typical example of a forward model is the equation describing the longitudinal behavior of a road vehicle:

$$c_m \cdot \frac{d}{dt} v(t) = -\left[c_0 + c_2 \cdot v^2(t)\right] + F(t), \tag{4.1}$$

where $v(t)$ is the velocity and $F(t)$ the propulsion force. The coefficients $\{c_m, c_0, c_2\}$ parametrize the rolling friction, the aerodynamic friction, and the inertia effects. In this formulation, the force is the cause—therefore, it must be known *a priori*—and the velocity is the effect. The backward formulation inverts this causality, that is, the velocity becomes the cause and the force the effect:

$$F(t) = c_m \cdot \frac{d}{dt} v(t) + \left[c_0 + c_2 \cdot v^2(t)\right]. \tag{4.2}$$

Of course, for this approach to be useful, the velocity $v(t)$ must be known *a priori*. For instance, this situation is the case when standardized test cycles are to be followed. There are many such cycles,

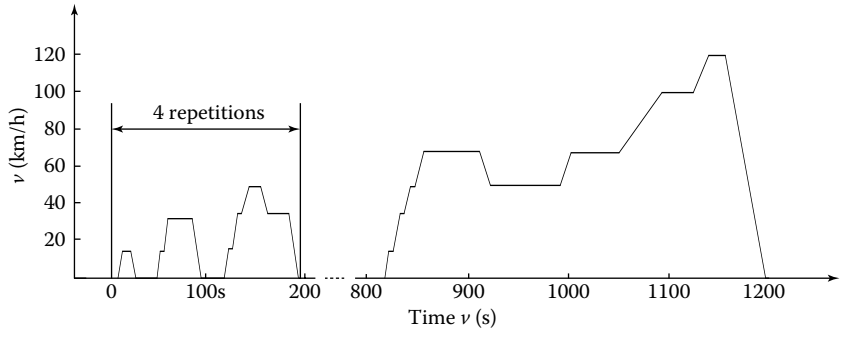

FIGURE 4.1 European driving cycle MVEG-95. Vehicle speed v (km/h) and time t (s).

for example, the MVEG-95 in Europe, illustrated in Figure 4.1, or the FTP-75 in the United States. In addition to such cycles defined by legislation, all car manufacturers have their own drive cycles that better reflect the average driving behavior and/or include additional influences, notably topographic elevation variations. All of these cycles cannot prescribe the velocity at each time instant t. The velocity $v(t)$ can be known only at specific points in time t_k such that

$$F(t) \approx c_m \cdot \frac{v(t_{k+1}) - v(t_k)}{h} + \left[c_0 + c_2 \cdot \left(\frac{v(t_{k+1}) + v(t_k)}{2} \right)^2 \right], \quad \forall t \in [t_k, t_{k+1}). \tag{4.3}$$

This approximation is quite accurate if the test cycle to be followed is a piecewise affine function of time (such as, for instance, the MVEG-95). In other cases, the approximation is acceptable only if the sampling interval h is small compared to the relevant dynamic effects (see below). Typically, h is equal to 1 s, but smaller or varying sampling intervals are used as well.

Forward models are necessary to design and optimize lower-level controllers, such as torque or air/fuel ratio control loops. The design of controllers that manage the energy flows of hybrid electric vehicles (HEV) and, even more so, the algorithms used to find the optimal size and structure of HEV powertrains rely on backward models. The key advantage of such models is the very low computational effort they induce in any simulations. Particularly, if vector-based algorithms are used, the simulation of a standard HEV powertrain model for several thousands of seconds requires only a few seconds on a standard PC.

4.2.2 Mechanical Systems

Equation 4.3, which describes the longitudinal vehicle dynamics, has been introduced in the last section. The parameter c_m describes the vehicle inertia. Assuming a rigid powertrain, this parameter can be written as

$$c_m = m + \frac{\Theta}{r^2 \cdot \gamma^2}, \tag{4.4}$$

where m is the total mass of the vehicle, Θ the inertia of all rotating parts before the gear box, γ the total gear ratio, and r the radius of the wheels. For many purposes, the second summand in Equation 4.4 can be approximated as $0.1 \cdot m$, but it is not very difficult to include the formulation of Equation 4.4 in all optimizations. Particularly when the gear ratio γ is a degree of freedom that is utilized in the optimizations, this approach is necessary. The parameter c_0 represents rolling friction and road inclination:

$$c_0 = m \cdot g \cdot (c_r + \sin(\alpha)) \approx m \cdot g \cdot (c_r + \alpha), \tag{4.5}$$

where c_r is the tire friction coefficient and α the road inclination measured in radians. The parameter c_2 represents the aerodynamic friction:

$$c_2 = \frac{1}{2} \cdot \rho \cdot c_w \cdot A_f, \tag{4.6}$$

TABLE 4.1 Typical Modeling Parameters Valid for HEV Passenger Cars

Parameter	Nominal Value (Range)		Unit
c_w	0.33	(0.25–0.40)	—
c_r	0.013	(0.008–0.015)	—
A_f	2.50	(2.00–3.00)	m^2
m	1500	(1000–2500)	kg
Θ	0.25	(0.20–0.30)	kg m^2
γ	—	(2.5–15.0)	—
ρ	1.15	(1.00–1.30)	kg/m^3
r	0.25	(0.20–0.30)	m
e	0.43	(0.40–0.45)	—
p_{m0}	1.8×10^5	$(1.5\times10^5–3.0\times10^5)$	Pa
V_0	230	(40–600)	V
R_0	0.3	(0.2–0.5)	Ω

Note: The gear ratio γ depends on the gear selected; the smallest value represents the highest gear, the highest value the first gear.

where ρ is the density of air, c_w the aerodynamic resistance coefficient, and A_f the area of the front of the vehicle. Table 4.1 lists typical ranges for all parameters introduced in this chapter.

Of course, in real vehicles, many additional losses are present. For instance, auxiliary devices require some power and the transmission causes additional friction losses. For the sake of brevity, these and other effects are neglected in this chapter.

4.2.3 Engine Systems

Internal combustion engines are very complex and not yet fully understood devices that transform the energy stored in hydrocarbons to mechanical work using the "detour" of heat generated by combustion. The high energy and power densities and the low cost of such engines are the main reasons for their enormous success in the last 100 years. For the purposes of powertrain modeling and control, the fuel consumption of internal combustion engines (spark and compression ignited) can be approximated sufficiently well using the following approach.

The subsequent definitions are needed below. First, the mean effective pressure p_{me} is introduced by

$$p_{me} = \frac{4 \cdot \pi \cdot T_{me}}{V_d}, \tag{4.7}$$

where T_{me} is the mean effective (useful) engine torque and V_d is the displaced volume of the four-stroke engine. Second, the mean fuel pressure is introduced by

$$p_{mf} = \frac{m_f \cdot H_l}{V_d}, \tag{4.8}$$

where m_f is the mass of fuel burned per engine cycle and H_l is the lower heating value of the fuel burned in the engine (~43 MJ/kg of fuel for gasoline and diesel fuel). With these preparations it is now possible to formulate a simple yet quite precise model of the fuel-to-work conversion efficiency of an engine:

$$p_{me} = e(\omega) \cdot p_{mf} - p_{m0}(\omega), \tag{4.9}$$

where $e(\omega)$ is a mildly engine speed-dependent internal efficiency value and $p_{m0}(\omega)$ a similarly speed-dependent friction pressure (Pachernegg, 1969). Typically, a second-order polynomial fit yields very good results, but for initial calculations the values of $e(\omega)$ and $p_{m0}(\omega)$ can even be assumed to be constant. Note

that Equation 4.9 does not depend on the engine size. Combined with Equations 4.7 and 4.8, that equation yields a scalable description of the energy conversion efficiency of any internal combustion engine.

The second approach to describe the efficiency of an engine is to use measured maps. As an example, Figure 4.2 shows the efficiency map of a modern spark-ignited (gasoline) engine. Using Equation 4.9 and constant values $e = 0.43$ and $p_{m0} = 180,000$ (N/m^2), the engine efficiency can be approximated as indicated in Figure 4.2 by the thin straight lines. In the relevant operating points, this approximation yields quite an accurate approximation, which can be made much more precise using speed-dependent coefficients $e(\omega)$ and $p_{m0}(\omega)$.

4.2.4 Electric Systems

Similar to internal combustion engines, electric motors can be described using a scalable description as introduced in Equation 4.9 or using measured maps. Figure 4.3 shows an example of such a map. These maps can encompass all four quadrants, but usually only two quadrants are measured due to the assumption of a symmetrical behavior for negative speeds.

Power electronic devices have very high efficiencies and are so fast that their losses and dynamical behavior can be neglected for the purposes of energy supervisory control. This is not true for battery systems. The following model is the simplest one that can be used for the purposes of energy management:

$$I_b(t) = \frac{V_0 - \sqrt{V_0^2 - 4R_0P_b(t)}}{2R_0}, \quad x(t) = \frac{Q_b(0) - \int_0^t I_b(\tau)_{d\tau}}{Q_{b,max}}, \tag{4.10}$$

where $I_b(t)$ is the battery current (positive when discharging), $P_b(t)$ the electric power delivered by ($P_b(t) > 0$) or stored in ($P_b(t) < 0$) the battery, and $0 < x(t) < 1$ is the state of charge (SoC) of the battery. The parameters in Equation 4.8 are the open-circuit voltage V_0, the internal resistance R_0, and the maximum battery capacity $Q_{b,max}$. All of these parameters can be assumed to be constant in a first analysis. Table 4.1 lists reasonable values. More precise calculations usually assume those parameters to depend on the SoC and the current of the battery.

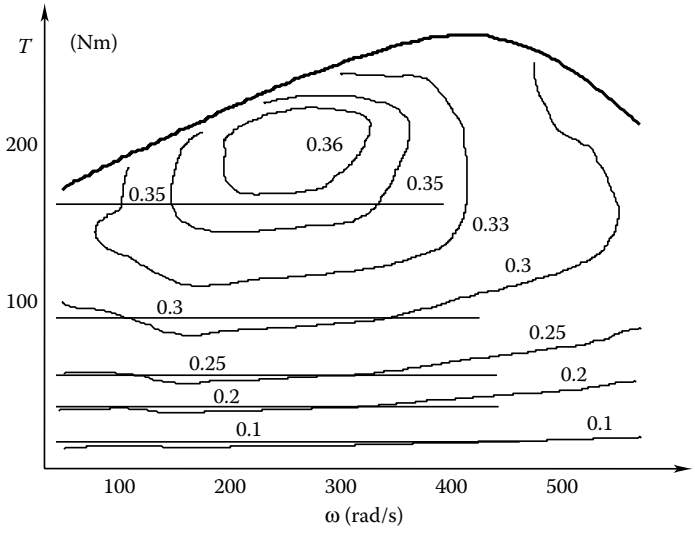

FIGURE 4.2 Measured (bold curves) and approximated (thin lines) engine efficiency map of a modern 2.2 l SI engine. Engine torque T (Nm) and engine speed ω (rad/s).

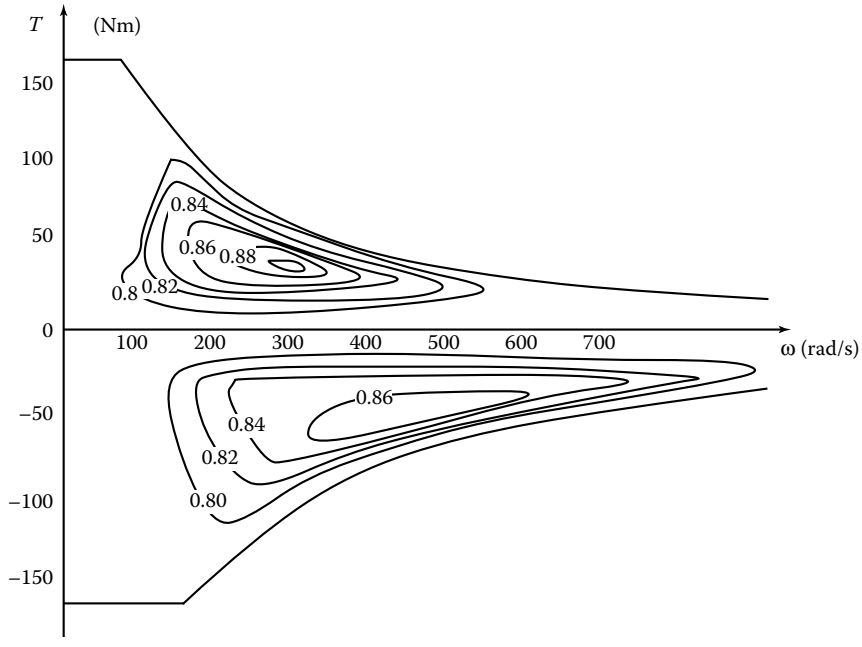

FIGURE 4.3 Measured efficiency map of a 15-kW electric motor. Motor torque T (Nm) and motor speed ω (rad/s).

4.3 Noncausal Control Methods

4.3.1 Offline Optimization

Optimizing the energy consumption of a hybrid powertrain consists of a two-level analysis. At a more abstract level, the energy management strategy can be designed for a given test drive cycle, through optimization of the power flows between the system components. Such a procedure is called an *offline* optimization, since the drive cycle must be known *a priori*. That is obviously not possible during the real operation of the system, except for some particular scenarios. Therefore, for a practical implementation, an *online* controller is necessary. Nevertheless, offline optimization is a very useful tool as it can be used to provide an optimal performance benchmark, which can then be used to assess the quality of any causal but suboptimal online controller. Moreover, in some cases, the optimal nonrealizable solution provides insights in the way such realizable control systems should be designed.

As with many physical systems, an HEV is characterized by disturbance inputs, control inputs, measurable outputs, and state variables. In an offline framework, the main disturbance is the test drive cycle that has to be followed. As discussed is Section 4.2, the knowledge of the vehicle speed profile directly yields the speed and torque levels required at the wheels. This total power is provided by the powertrain system components. The synthesis of the reference signals for each component is the objective of the supervisory controller. The nature of these variables strongly depends on the type of HEV architecture. For example, in a parallel hybrid the IC engine torque can be treated as a controlled variable, while the electric motor torque is constrained to fulfill the total torque required at the wheels. The nature of the state variables is obviously related to the dynamics of the systems, which generally include mechanical, thermal, electrical, and electrochemical subsystems. Usually, for the purposes of energy management, an HEV can be described using "backward" models. Thus the number of state variables strongly decreases, and it can be reduced to only include integral quantities such as the battery SoC $x(t)$.

Stated in mathematical terms, the supervisory control problem consists of finding the control law $u(t)$ that minimizes the fuel consumption

$$J_f = \int_0^{t_f} \dot{m}_f(u(t), x(t), w(t)) \cdot dt \tag{4.11}$$

over the prescribed drive cycle, when the disturbance inputs $w(t)$ are known *a priori*. This optimal control problem must respect several constraints, such as the dynamics of the state variables (usually the battery SoC)

$$\frac{dx(t)}{dt} = f(u(t), x(t), w(t)). \tag{4.12}$$

Of course, the minimization of J_f only is not sufficient, but it is necessary to force the battery SoC to stay within certain admissible boundaries during the whole trajectory (state constraints). Moreover, the terminal SoC (at the end of the cycle) must have a value that is close to the initial value in order to ensure a neutral balance of SoC that is required by the self-sustaining nature of any self-sustaining HEV system. Concerning this latter point, the optimization problem for a plug-in HEV would be rather different.

Mathematically speaking, a terminal state constraint is added to the criterion J_f to build a new multi-objective performance index

$$J = J_f + \varphi(x(t_f)). \tag{4.13}$$

The function $\varphi(\cdot)$ describes the type of constraint applied to the terminal SoC. Even though other definitions are possible and used in practice, the rest of this chapter only deals with hard constraints, that is, the final SoC must be strictly equal to the initial value,

$$x(t_f) = x(0) = x_0. \tag{4.14}$$

4.3.2 Dynamic Programming

One very common technique for solving the optimal control problem stated in the previous section is dynamic programming (DP) (Pu and Yin, 2007; Ao et al., 2008; Gong et al., 2008; Liu and Peng, 2008). Like other offline optimization techniques, all disturbances (in the case of deterministic DP) or at least their stochastic properties (in the case of stochastic DP (Johannesson et al., 2007)) must be known *a priori*.

DP uses the definition (Equation 4.13) of an overall cost function, but it extends this definition to any point of the time–state space, by defining the cost-to-go function

$$J_n(t, x(t)) = \varphi(x(t_f)) + \int_t^{t_f} \dot{m}_t(u(\tau), x(\tau), w(\tau)) \cdot d\tau. \tag{4.15}$$

This function depends on the control law adopted from the current time till the end of the cycle, $\pi = \{u(\tau)\}$, $\tau = t, \ldots, t_f$. The optimal cost to reach the terminal point (t_f, x_0) from any current point (t, x) is thus defined as

$$\Gamma(t, x) = \min_{\pi} J_\pi(t, x). \tag{4.16}$$

By definition, the value $\Gamma(0, x_0)$ corresponds to the optimal value of J_π that is sought.

To calculate the function $\Gamma(t, x)$, DP requires a discretization of the time–state space. Thus, the function Γ is only calculated at a fixed number of points $t_k = k \cdot \Delta t$, $k = 0, \ldots, N$, and $x_i = x_{min} + i \cdot \Delta x$,

$i = 0, \ldots, p$ ($p = (x_{max} - x_{min})/\Delta x$). The computation starts by setting

$$\Gamma(t_f, x_i) = \varphi(x_i). \tag{4.17}$$

In the case of a "hard" terminal constraint, Equation 4.17 can be reformulated as

$$\Gamma(t_f, x_0) = 0 \quad \text{and} \quad \Gamma(t_f, x_i) = \infty, \quad \forall x_i \neq x_0 \tag{4.18}$$

in order to make unfeasible any terminal SoC different from the target one. The computation proceeds backward in time solving the recursive algorithm

$$\Gamma(t_k, x_i) = \min_{u \in V} \left\{ \Gamma(t_{k+1}, x_i + f(u, x_i, w(t_k)) \cdot \Delta t) + \dot{m}_f(u, w(t_k)) \cdot \Delta t \right\}. \tag{4.19}$$

Equation 4.19 clearly shows that a backward model of the system must be used to evaluate the SoC variation and fuel consumption as a function of the current disturbance $w(t)$, the current state x_i, and the control input applied. The control inputs $u(t)$ are limited to a feasible subset that can depend on the state, $V(x(t))$. In practice, this subset also must be discretized to restrict the search to q values u_j, $j = 1, \ldots, q$. These values can vary at each time and as a function of the state.

The arguments of minimization are stored in a feedback control function $U(t_k, x_i)$ which is then used to reconstruct the optimal trajectories $x^\circ(t)$, $u^\circ(t)$, and consequently $\dot{m}^o_{fuel}(t)$. Starting from $t = 0$ and proceeding forward in time,

$$u^\circ(t_k) = U(t_k, x^\circ(t_k)), \quad x^\circ(t_{k+1}) = x^\circ(t_k) + f(u^\circ(t_k), x^\circ(t_k), w(t_k)) \cdot \Delta t. \tag{4.20}$$

Solving Equations 4.19 and 4.20 requires particular care since generally neither the state value $x_i + f(u, x_i, w(t)) \cdot \Delta t$ nor the state value $x^\circ(t_k)$ matches one of the possible points of the grid. Therefore, the corresponding values of Γ and U must be interpolated from the values calculated for the closest points of the grid. Several interpolation methods can be used, each having their specific benefits and drawbacks (Guzzella and Sciarretta, 2007).

Another problem arises when, as in Equation 4.18, an infinite cost is assigned to handle unfeasible states or control inputs. If an infinite cost is used together with an interpolation scheme, the infinity value propagates backward in the grid and the number of unfeasible states artificially increases. A number of techniques have been proposed to handle these interpolations correctly, using large but finite instead of infinite values or calculating exactly the boundaries between feasible and unfeasible states (Sundström et al., 2009).

Other techniques are aimed at reducing the time consumption of DP. In general, the computation burden of DP algorithms scales linearly with the problem time N, the number of discretized state values p, and the number of the discretized control input values q. To reduce the computational effort, one technique is based on the reduction of p. At each iteration, the state space is selected as a small fraction of the entire space, centered on the optimal trajectory evaluated in the previous iteration. Unfortunately, DP algorithms have a complexity that is exponential in the number of state variables n and control input variables m,

$$O(N \cdot p^n \cdot q^m). \tag{4.21}$$

This makes the algorithm only suitable for low-order systems. Fortunately, in HEV optimization problems, the only state variable is the battery SoC, and thus $n = 1$, while m usually is limited to two inputs (e.g., torque split and gear number).

4.3.3 Connection to the Minimum Principle

One alternative to DP for offline energy management optimization consists of solving the optimal problem Equations 4.11 through 4.13 by means of Hamilton–Jacobi theory and then using Pontryagin's

minimum principle (Bryson and Ho, 1975). This technique has certain advantages over DP in terms of computational burden and complexity of the algorithm. Moreover, it is inherently closer to its online counterparts. However, state constraints are more difficult to handle and convergence of the algorithm can be problematic. The starting point of this approach is a Hamiltonian function

$$H(u(t), x(t), w(t), \lambda(t)) = \dot{m}_f(u(t), x(t), w(t)) + \lambda(t) \cdot f(u(t), x(t), w(t)), \tag{4.22}$$

from where the optimal control law is calculated as

$$u^{opt}(t) = arg \min_u H(u(t), x(t), w(t), \lambda(t)). \tag{4.23}$$

The dynamics of the variable λ is given by the Euler–Lagrange equations,

$$\frac{d\lambda(t)}{dt} = -\frac{\partial H(u, x, w, \lambda)}{\partial x}, \tag{4.24}$$

in such a way that λ can be defined as a new state variable (costate) that is adjoint to x, whose dynamics are still given by Equation 4.22. Unfortunately, while the boundary condition for the state x is prescribed at the initial time, the boundary condition for λ is prescribed at the terminal time,

$$\lambda(t_f) = \frac{\partial \varphi(x(t_f))}{\partial x(t_f)}. \tag{4.25}$$

This is a typical two-point boundary condition problem, which must be solved numerically. A plethora of methods exist ("shooting" and similar algorithms, as discussed below), all of which have their drawbacks and advantages. Moreover, the solution obtained is open loop. That is, it is valid only for the specific initial λ. Extending this to a feedback solution can be difficult. Often, formulating a simple nonlinear least-squares problem on the final constraint of the adjoint variables is sufficient, particularly, if one solution for a particular case is known and other cases are solved using successive approximation methods (homotopy algorithms).

This procedure is particularly intuitive when the "hard" constraint expressed by Equation 4.18 is used. In that case, the unknown initial value of λ is the one that ensures the SoC equality (Equation 4.14). In most practical cases, the relationship between $\lambda(0)$ and $x(t_f)$ is monotonous and there is only one value $\lambda(0)$ that fulfills Equation 4.14. In particular, if $\lambda(0)$ is too low, $x(t_f)$ will be higher than x_0. On the other hand, if $\lambda(0)$ is too high, $x(t_f)$ will be lower than x_0. At the end of the iteration, a correction of $\lambda(0)$ thus can be made in accordance with the sign of $x(t_f) - x_0$, for example, using the bisection method. Figure 4.4 shows the flowchart of this approach. This method, sometimes referred to as a *shooting algorithm*, is often applied in combination with the further assumption that the right-hand term of Equation 4.24 is negligible. In fact, while the fuel consumption rate does not directly depend on the SoC, the function $f(\cdot)$ in the definition of Equation 4.10 does, through the battery parameters such as open-circuit voltage and internal resistance. However, in many cases, this dependency can be neglected (especially for the internal resistance), so that Equation 4.24 can be reduced to (Sciarretta and Guzzella, 2007)

$$\frac{d\lambda}{dt} \approx 0 \rightarrow \lambda(t) \approx \lambda(0). \tag{4.26}$$

In the case in which λ can be treated as a constant, the Hamiltonian function defined by Equation 4.22 acquires a new meaning. The SoC variation can be expressed as a function of battery terminal current I_b and nominal capacity $Q_{b,max}$ as

$$\frac{dx(t)}{dt} = f(u(t), x(t), w(t)) = -\frac{I_b(u(t), x(t), w(t))}{Q_{b,max}} \tag{4.27}$$

using a simple battery model and considering that positive currents discharge the battery. Under the assumptions stated above, the battery open-circuit voltage is a constant, such that Equation 4.22 can be

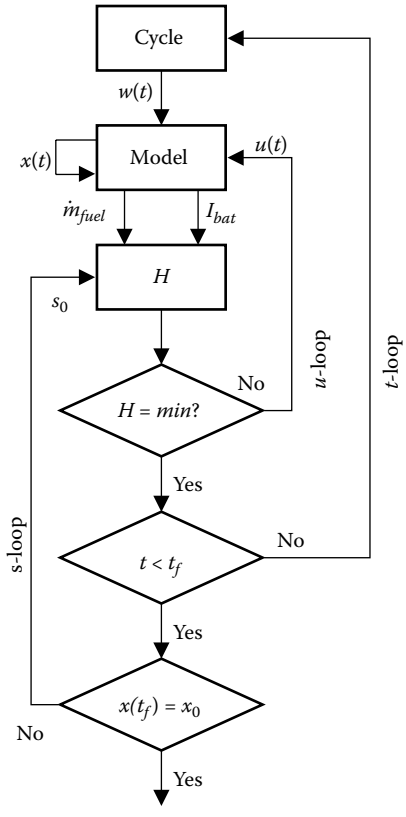

FIGURE 4.4 Flow chart of the bisection algorithm used to find the correct initial condition for the costate $\lambda(0)$.

written as

$$P_H(u(t), x(t), w(t), \lambda(t)) = H_l \cdot \dot{m}_f(u(t), x(t), w(t)) + s_0 \cdot I_b(u(t), x(t), w(t)) \cdot V_0, \tag{4.28}$$

where $s_0 = -\lambda_0 H_l/(Q_{b,max} V_0)$ is a new constant term that has the advantage over λ_0 of directly weighting two power terms in Equation 4.28, namely, the fuel chemical power $\dot{m}_f H_l$ and the electrochemical (inner) battery power $I_b V_0$. The parameter s_0 is called equivalence factor. It plays an important role in the approach described in Section 4.4.2.

4.4 Causal Control Methods

4.4.1 Online Optimization

In conventional ICE-based powertrains, the engine speed is a state variable that can only be influenced through appropriate commands to the transmission system, while the desired engine torque is defined by the driver. This torque setpoint is sent to a low-level engine controller, which controls the engine such that it delivers the desired amount of torque. In contrast, in HEVs the total torque required can be produced by different sources. Moreover, in some HEV architectures also the speed at which the engines and the motors operate can be chosen independently of the vehicle speed. Accordingly, HEVs have several degrees of freedom that must be chosen in *real-time* during the vehicle's operation.

This control problem is similar to the optimization problem introduced in Section 4.3.1. However, the main difference is that some of the disturbance inputs $w(t)$ are known only for times $\tau \leq t$, where t is

the current time. Moreover, the vehicle speed is measured, while the total torque required at the wheels is *interpreted* from the driver's action on the acceleration and brake pedals. This task is very critical and a proper interpretation substantially affects the drivability properties of the whole powertrain.

Several techniques have been proposed and are being used to accomplish supervisory control in such situations. A first distinction must be made between realizable heuristic and noncausal optimal controllers. The first class of controllers represents the state of the art in most prototypes and mass-production hybrids. They are based on Boolean or fuzzy rules involving various vehicular variables. The logics implemented by these strategies consist of determining first the state of the engine (on or off state) according to some transition rules. If the engine is running, it can be turned off—and a purely electric propulsion mode can be activated—only if several conditions are simultaneously met by the system, including the battery being sufficiently charged, the engine sufficiently warm, the battery sufficiently cold, the vehicle speed sufficiently low, the total power demand sufficiently low, and so on. Conversely, if the engine is stopped it must be turned on when at least one of the previous conditions is not met. If these rules prescribe the engine to be firing, a certain torque setpoint is then assigned to the engine, together with a speed setpoint if that is not fixed by the vehicle speed. The torque setpoint is a consequence of the power required at the wheels. However, this power is modulated in such a way as to optimize the operation of the battery. If the SoC is too low with respect to a target value, additional power is demanded from the engine in order to charge the battery. Conversely, if the SoC is higher than the target, the battery can assist the engine in fulfilling the power demand. Finally, as a function of the engine setpoints and the total power demand, heuristic controllers assign setpoints to all the other components, including at least one electric machine.

Of course, the structure of the heuristic rules can be very complex in a real controller, involving several measured variables that have to be compared with crisp or fuzzy thresholds, and extensively using look-up tables to determine values to be assigned to control quantities. The role of these data is very critical for the behavior of the controller. Usually, a substantial amount of calibration is needed in order to adapt these data to a given system and often to a given driving situation, that is, a test drive cycle.

A practical alternative to heuristic controller design is a design based on the optimal control laws presented in Section 4.3.3.

4.4.2 Equivalent Consumption Minimization Strategies

The control strategies based on Pontryagin's minimum principle (PMP) are collectively called equivalent consumption minimization strategies (ECMS) (Paganelli et al., 2000), although this acronym initially referred to a specific design that was derived heuristically. In fact, this acronym stems from the intuitive meaning of Equation 4.28 that expresses the Hamiltonian function to be minimized as the sum of the fuel power and the electrochemical power weighted by an equivalent factor s_0. The product $s_0 I_{bat} V_{oc}$ can be reinterpreted as the fuel consumption (in power units) that is equivalent to the electrochemical consumption. Therefore, the total cost function to be minimized is an equivalent fuel consumption resulting from the algebraic sum of the two contributions.

Of course in online controllers, the equivalence factor s_0 cannot be determined using a shooting method. Rather, it must be estimated in real-time as a feedback of the current state of the system as shown in Figure 4.5. The Hamiltonian cost function is constructed as

$$P_H(u, x(t), t) = P_f(u, t) + s(t) \cdot P_{ech}(u, x(t), t), \tag{4.29}$$

where the estimation of the equivalence factor is made at each instant. The control vector is then obtained as

$$u^{opt}(t) = \underset{u}{arg\,min}\, P_H(u(t), x(t), t). \tag{4.30}$$

It must be emphasized that the dynamics of $s(t)$ do not try to mimic the real dynamics of the costate $\lambda(t)$ as defined by Equation 4.24. These dynamics are simplified as in Equation 4.26, yielding a "true" equivalence factor that is constant during optimal operation. However, this constant value is not known

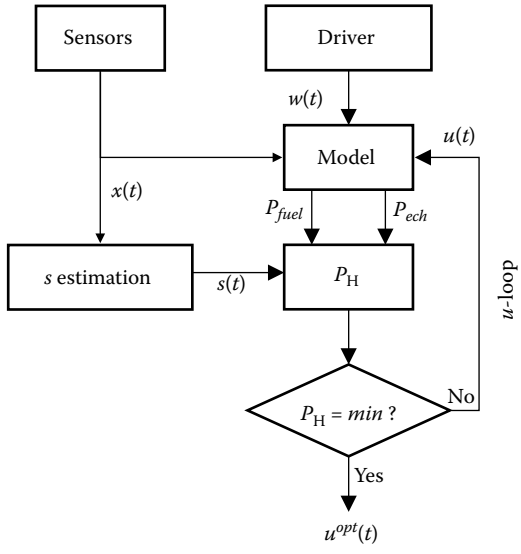

FIGURE 4.5 Flow chart of the real-time estimation algorithm used to find the equivalence factor s_0.

a priori since it depends on the overall (including future) driving conditions. Thus, the estimation of such a constant requires a correction term that obviously evolves with time. A simple adaptation algorithm can be designed as a function of the battery SoC as Ambühl et al., 2007; Kessels et al., 2008; and Chasse et al., 2009,

$$s(t) = s_c - k_p(x(t) - x_c). \tag{4.31}$$

This adaptation rule is based on the relationship between the equivalence factor (or, equivalently, the costate) and the battery SoC discussed in Section 4.3.3. Once a certain target value x_c is prescribed for the SoC, the rule (Equation 4.31) corrects any positive deviations of the current $x(t)$ from x_c by decreasing $s(t)$, that is, by favoring the use of electrochemical energy to discharge the battery. On the contrary, when the SoC is lower than its target value, $s(t)$ is increased to penalize any further use of the battery and to favor its recharging instead. The parameter x_c is chosen as a function of the battery technology, while s_c is a first guess of s_0 that can be adapted online.

In addition to the estimation of the equivalence factor, the practical implementation of ECMS requires a careful proceeding in many aspects (Chang et al., 2009; Phillips et al., 2009). The Hamiltonian P_H given by Equation 4.29 can be modified with additional terms to penalize situations that, though effective to improve global efficiency, could be undesirable from a drivability point of view. Such penalization terms might concern, for instance, changes of engine status, changes of engine load, changes of gear, and so on. In particular, the need to avoid too frequent engine starts or stops can lead to the introduction of additional refinements in the selection of the minimal cost function. For instance, P_H can be separately minimized for two subsets of the control vector corresponding to engine on and off. Then the comparison between the two candidate control vectors is subjected to hysteresis thresholds and timing issues.

4.4.3 Extensions of the ECMS Approach

Although they are unavoidable when the optimal s_0 is not known *a priori*, adaptation rules such as Equation 4.31 necessarily deteriorate the control performance with respect to the optimum, leading to suboptimal control designs. Of course, control parameters such as k_p and s_c can be carefully calibrated to reduce the performance losses. However, adapting these parameters to any real-life operation is usually a complex task, particularly in the presence of road altitude variations.

One possible solution is to improve the online estimation of the equivalence factor by including information coming from the vehicle environment. Such information can be generated using sensors (mainly global positioning systems (GPS) coupled with global information systems (GIS) to detect the future altitude profiles, as well as radar or a laser scanner to detect the presence of fixed or moving obstacles downstream of the vehicle) or received from emitting infrastructures. How to include these pieces of information in an ECMS framework is still a subject of research. Most of the approaches proposed usually consist of adapting some control parameters that are identical or equivalent to k_p and s_c in Equation 4.31. The adaptation can be driven by a full-scale optimization—for example, using DP—performed online using a detailed estimation of the future driving profile (Back et al., 2004; Sciarretta et al., 2004). In a simpler approach, the adaptation can be based on global energy balances using only estimations of future energy demand. Indeed, it is rather intuitive to extend the definition of Equation 4.31 to the case where energy can be stored on board not only in an electrochemical form but also as a potential energy. In that case, the equivalence factor estimation should depend not only on the current SoC but also on the current altitude $z(t)$, for instance, in a way such as

$$s(t) = s_c - k_p(x(t) - x_c) - k_z(z(t) - z_c) + \cdots \tag{4.32}$$

where z_c is a target altitude that can be obtained by a navigation system and $z(t)$ can be estimated using GPS and GIS. Rule (4.32) can be additionally extended to include kinetic energy and thus take into account accelerations or decelerations (Ambühl and Guzzella, 2009).

Other extensions of the ECMS might concern additional optimization criteria, for example, adding pollutant emissions to fuel consumption (Ao et al., 2009). Both are integral criteria deriving from instantaneous mass flow rates. However, combining these rates into a single performance index necessarily requires the introduction of user-defined weighting factors. Additional dynamics can be also considered with respect to the ECMS described in Section 4.4.1, for instance, vehicle speed or thermal levels. The latter, in particular, might prove very important in combination with a pollutant-based criterion. A thermal state variable with its costate adjoint to the Hamiltonian cost function might lead to the finding of optimal compromises between, for instance, turning off the engine to avoid pollutant emissions locally and keeping it on to avoid decreasing the catalyst temperature and reduce pollutant emissions later.

4.5 Software and Tools

4.5.1 Modeling Tools

MATLAB® and Simulink are commonly used numerical tools that can be used for modeling and control system design. There exist many software packages and simulation tools for MATLAB and Simulink, both free and licensed, for assessing the fuel consumption in hybrid-electric vehicles. As pointed out earlier, a key concept of these tools is the scalable modules of each of the powertrain components. This section presents two model libraries for MATLAB and Simulink that are used for vehicle propulsion system optimization. These tools make it possible for powertrain systems to be designed quickly and in a flexible manner and to calculate easily the fuel consumption of such systems. In general, the tools are based on the modeling assumptions presented earlier in this chapter.

The advanced vehicle simulator (ADVISOR), developed by the National Renewable Energy Laboratory, was available online as a free library of models used for vehicle powertrain modeling. The library contains several various powertrain components as well as energy management strategies. The library was bought by AVL and is available by license from 2003.

One of the free vehicle propulsion system libraries is the QSS toolbox (Guzzella and Amstutz, 1999; Rizzoni et al., 1999). The QSS toolbox was developed at ETH Zurich and is available for download at the URL www.idsc.ethz.ch/research/downloads. Due to the extremely short CPU time it requires (for a conventional powertrain, a speedup factor of 100–1000 is quite common on a regular PC), a QSS model is ideally suited for the optimization of the fuel consumption under various control strategies.

4.5.2 Optimization Software

Typically in HEV optimization, the optimization problem consists of two types of optimizations. The first is the dynamic optimization of the power split between the two energy converters in the powertrain. The second is the static optimization of different component sizes such as engine displacement and electric motor maximum power. Both the static and the dynamic optimization must be treated when optimizing the complete vehicle. The static optimization is commonly solved using an optimization algorithm capable of approximately solving nonconvex optimization problems. Some tools are available in MATLAB, while others, such as particle swarm optimization toolboxes and genetic algorithm toolboxes, can be found online.

As mentioned in Section 4.3.2, the dynamic optimization in the HEV is often solved using the DP algorithm. Of course, the DP solution can only be found if the complete future disturbances and reference inputs are known. In this sense, the solution is not causal. Nevertheless, the optimal solution is very useful, since it can be used as a benchmark to which all causal controllers can be compared with.

When implementing the DP algorithm, special attention must be given to minimizing the overall computational cost since the computational complexity of the DP algorithm is exponential in the number of states and inputs. Also, as described in Section 4.3.2, numerical problems can arise and produce a suboptimal solution. The implementation of suitable numerical algorithms that efficiently solve a given DP problem is, therefore, a nontrivial part of a hybrid vehicle optimization process.

In Sundström and Guzzella, 2009, a generic DP function that efficiently solves deterministic DP problems using MATLAB is presented. The function, called Dynamic Programming Matrix (DPM) function, implements the DP algorithm and leaves the model function to be implemented by the user. The largest reduction in computation time is realized if the model function is implemented to support matrix-valued inputs and outputs. If the model function uses matrix operations the DPM function typically solves the energy management problem in HEVs in less than 1 min of calculation on a standard desktop computer. The DPM function and two example problems are available at www.idsc.ethz.ch/research/downloads.

4.6 Example

4.6.1 System Description and Modeling

The simple case analyzed in this example is a parallel hybrid-electric powertrain. The engine and the electric motor act with identical rotational speed on the same shaft and no gear ratio changes are considered. To allow for pure electric driving, the engine can be decoupled from the powertrain using an actuated clutch. Since the engine cannot run below some minimum speed, this powertrain must be operated electrically at low speeds.

The goal is to evaluate the fuel-minimal power split between engine and motor for a given drive cycle. The powertrain model is formulated using a backward approach as introduced in Section 4.2. The parameters used for this example are summarized in Tables 4.1 and 4.2.

The force $F(t)$ required at the wheel to follow the drive cycle is computed with Equations 4.3 through 4.6. The torque and speed after the single gear transmission result as

$$T_g = \frac{F(t) \cdot r}{\gamma \cdot \eta_g^{sign(F(t))}}, \quad \omega_g = \frac{v(t_{k+1}) + v(t_k)}{2} \cdot \frac{\gamma}{r}, \tag{4.33}$$

where η_g is the efficiency of the transmission, which is assumed to be constant.

The torque T_g required at the transmission input must be delivered by a combination of engine and motor torque. Hence, the following balance must be satisfied:

$$T_g = T_{me} + T_{mm}, \tag{4.34}$$

TABLE 4.2 Modeling Parameters of the Single-Gear HEV—Example of Section 4.6

Parameter	Value	Unit
V_d	1×10^{-3}	m^3
q_0	200	Nm
q_1	0.7	Nms
q_2	-1×10^{-3}	Nms^2
H_l	42.5×10^6	J/kg
$\omega_{e,\,min}$	100	rad/s
$\omega_{e,\,max}$	600	rad/s
γ	4	—
η_g	0.9	—
η_m	0.9	—
$P_{m,\,max}$	50×10^3	W
$T_{m,\,max}$	300	Nm
$Q_{b,\,max}$	23,400	C

where T_{me} is the mean engine torque and T_{mm} is the mean motor torque. For this example, the engine torque is chosen as the control variable, that is,

$$u = T_{me}. \tag{4.35}$$

The engine efficiency is modeled as speed independent for simplicity. As discussed in Section 4.2.3, the engine's fuel consumption over one time step h can be approximated by

$$\Delta m_f = \frac{\omega_g \cdot h}{4\pi} \cdot m_f, \tag{4.36}$$

where m_f is the fuel mass consumed per engine cycle resulting from Equations 4.7 through 4.9 and Equation 4.35 as

$$m_f = \begin{cases} \dfrac{V_d \cdot p_{m0}}{H_l \cdot e} + \dfrac{4\pi}{H_l \cdot e} \cdot u, & \text{if } u > 0 \\ 0, & \text{else} \end{cases} \tag{4.37}$$

The maximum engine torque is approximated with a second-order polynomial

$$T_{me,max} = q_0 + q_1 \cdot \omega_g + q_2 \cdot \omega_g^2, \tag{4.38}$$

where the speed of the engine is limited to the interval $\omega_g \in [\omega_{e,min}, \omega_{e,max}]$. With the parameters used in this example, which are listed in Table 4.2, the resulting range of speeds where the engine can be fired is limited to 22.5 and 135 km/h.

The electric motor is assumed to have a constant efficiency η_m, such that the electric power is linked to the requested torque and the control signal by

$$P_b = \frac{\omega_g \cdot (T_g - u)}{\eta_m^{sign(T_g - u)}}. \tag{4.39}$$

The motor can deliver a maximum torque $T_{m,max}$ and a maximum mechanical power $P_{m,max}$. It is assumed here that these limits are symmetric for the motor and generator mode. The battery is modeled as described in Equation 4.10.

4.6.2 Optimal Solution

The optimal solution to the optimal control problem defined by Equations 4.11 through 4.14 and the model described in Section 4.6.1 is found with DP. The boundary condition on the state variable is $x(t_f) = x(0) = x_0 = 0.55$. The DPM function introduced in Sundström and Guzzella (2009) is used here. The algorithm yields as the result the optimal control signal $U(t,x)$ and the optimal cost-to-go $\Gamma(t,x)$ as a function of time and battery SoC.

The optimal control signal is illustrated in Figure 4.6 for the MVEG-95 drive cycle. This figure shows that the optimal control for this powertrain consists mainly of pure electric driving and recharging, which dominates at low SoC conditions. The solid graph represents the optimal state trajectory resulting from applying the optimal control $U(t,x)$ with the initial condition $x(0) = 0.55$.

The optimal cost-to-go $\Gamma(t,x)$ function resulting from DP can be used to compute the optimal equivalence factor introduced in Section 4.3.3. The Principle of Optimality shows that the optimal costate λ is the gradient of the cost-to-go $\Gamma(t,x)$ in x. Normalizing this optimal costate yields the standard equivalence factor

$$s^o(t,x) = -\frac{H_l}{Q_{b,max}V_0}\frac{\partial\Gamma(t,x)}{\partial x}. \tag{4.40}$$

Figure 4.7 represents the optimal equivalence factor for this powertrain example and for the MVEG-95 drive cycle. It shows that the equivalence factor increases with decreasing SoC. This increase is larger close to the boundaries of the state constraint. The optimal equivalence factor along the optimal state trajectory with the initial condition $x(0) = 0.55$ is indicated with the solid line and is shown as a function of time in the bottom plot. These results reveal that the equivalence factor is almost constant for most of the drive cycle. However, toward the end of the drive cycle, there is a substantial change of the value of s. This shows that the assumption of a constant equivalence factor holds as long as the characteristics of the drive cycle do not change considerably.

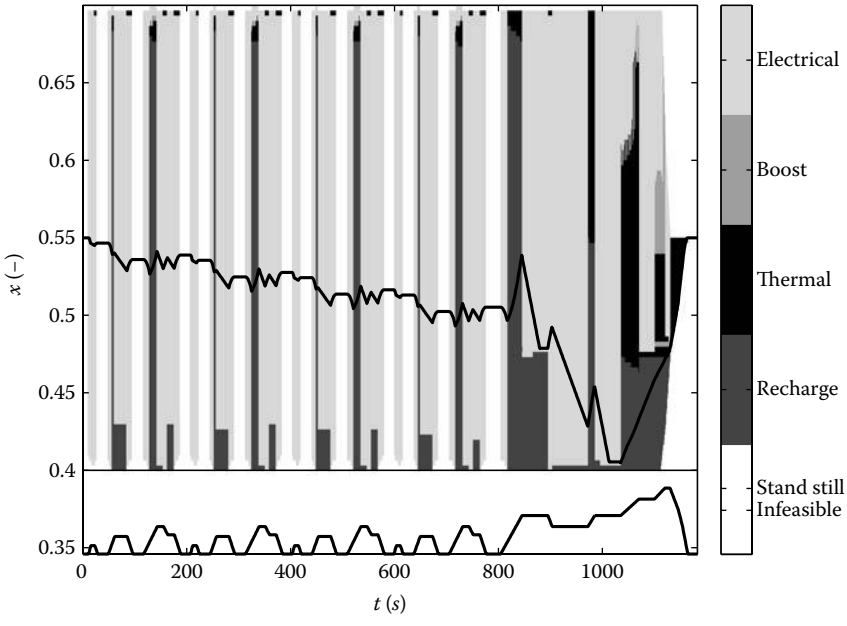

FIGURE 4.6 Optimal control signal as found with DP for the MVEG-95 driving profile.

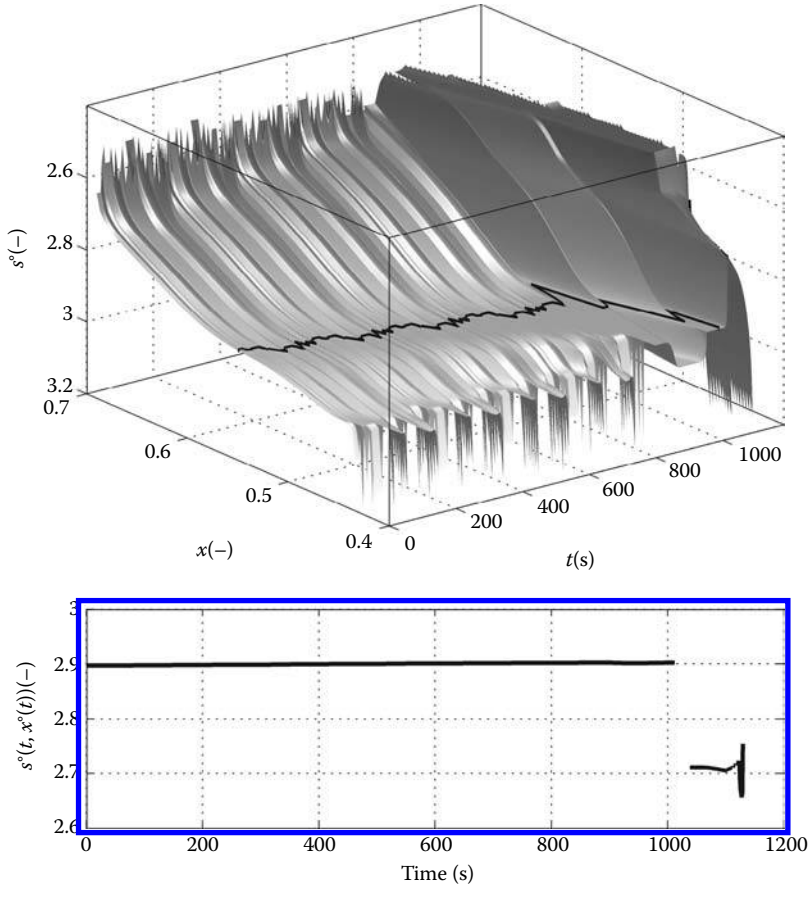

FIGURE 4.7 Optimal equivalence factor for the example on MVEG-95. Note the inverted vertical axis on the top plot.

References

Ambühl, D. and Guzzella, L., Predictive reference signal generator for hybrid electric vehicles, *IEEE Transactions on Vehicular Technology*, 58(9), 4730–4740.

Ambühl, D., Sciarretta, A., Onder, C.H., Guzzella, L., Sterzing, S., Mann, K., Kraft, D., and Küsell, M., A causal operation strategy for hybrid electric vehicles based on optimal control theory, *4th Braunschweig Symposium on Hybrid Vehicles and Energy Management*, Braunschweig, 2007.

Ao, G.Q., Qiang, J.X., Zhong, H., Mao, X.J., Yang, L., and Zhuo, B., Fuel economy and NO_x emission potential investigation and trade-off of a hybrid electric vehicle based on dynamic programming. *Proceedings of the IMechE, Part D—Journal of Automobile Engineering*, 222 (D10), 1851–1864, 2009.

Back, M., Terwen, S., and Krebs, V., Predictive powertrain control for hybrid electric vehicles. *IFAC Symposium on Advances in Automotive Control*, Salerno, 2004.

Bryson, E. and Ho, Y.C., *Applied Optimal Control*, Taylor & Francis, New York, 1975.

Chasse, A., Pognant-Gros, Ph., and Sciarretta, A., Online implementation of an optimal supervisory control for hybrid powertrains, *SAE International Powertrains, Fuels and Lubricants Meeting*, Florence, Italy, *SAE No. 2009–01–1868*, 2009.

Gong, Q.M., Li, Y.Y., and Peng, Z.R., Trip-based optimal power management of plug-in hybrid electric vehicles, *IEEE Transactions on Vehicular Technology*, 57(6), 3393–3401, 2008.

Guzzella, L., Automobiles of the future and the role of automatic control in those systems, *Annual Reviews in Control*, 33, 1–10, 2009.

Guzzella, L. and Amstutz, A., CAE-tools for quasistatic modeling and optimization of hybrid powertrains, *IEEE Transactions on Vehicular Technology*, 48(6), 1762–1769, 1999.

Guzzella, L. and Onder, C., *Introduction to Modeling and Control of Internal Combustion Engine Systems*, Springer-Verlag, Berlin, 2004.

Guzzella, L. and Sciarretta, A., *Vehicle Propulsion Systems—Introduction to Modeling and Optimization*, 2nd Edn, Springer Verlag, Berlin, 2007.

Johannesson, L., Asbogard, M., and Egardt, B., Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming, *IEEE Transactions on Intelligent Transportation Systems*, 8(1), 71–83, 2007.

Kessels, J.T.B.A., Koot, M.W.T., van den Bosch, P.P.J., and Kok, D.B., Online energy management for hybrid electric vehicles, *IEEE Transactions on Vehicular Technology*, 57(6), 3428–3440, 2008.

Liu, J.M. and Peng, H.E., Modeling and control of a powersplit hybrid vehicle, *IEEE Transactions on Control Systems Technology*, 16(6), 1242–1251, 2008.

Pachernegg, S.J., A closer look at the Willans-line, *SAE International Automotive Engineering Congress and Exposition*, Detroit, 1969.

Paganelli, G., Guerra, T.-M., Delprat, S., Santin, J.-J., Delhom, M., and Combes, E., Simulation and assessment of power control strategies for a parallel hybrid car, *Proceedings of the IMechE, Part D: Journal of Automobile Engineering*, 214(7), 705–717, 2000.

Phillips, A.M., McGee, R.A., Lockwood, J.T., Spiteri, R.A., Che, J., Blankenship, J.R., and Kuang, M.L., Control system development for the dual drive hybrid system, *SAE Paper No. 2009–01–0231*, 2009.

Pu, J. and Yin, C. Optimal fuel of fuel economy in parallel hybrid electric vehicles, *Proceedings of the IMecheE, Part D—Journal of Automobile Engineering*, 221 (D9), 1097–1106, 2007.

Optimal fuel of fuel economy in parallel hybrid electric vehicles, *Proceedings of the IMechE, Part D—Journal of Automobile Engineering*, 221(D9), 1097–1106, 2007.

Rizzoni, G., Guzzella, L., and Baumann, B., Unified modeling of hybrid electric vehicle drivetrains, *IEEE/ASME Transactions on Mechatronics*, 4(3), 246–257, 1999.

Sciarretta, A., Back, M., and Guzzella, L., Optimal control of parallel hybrid electric vehicles, *IEEE Transactions on Control Systems Technology*, 12, 352–363, 2004.

Sciarretta, A. and Guzzella, L., Control of hybrid electric vehicles—A survey of optimal energy-management strategies, *IEEE Control Systems Magazine*, 27(2), 60–70, 2007.

Sundström, O., Ambühl, D., and Guzzella, L, On implementation of dynamic programming for optimal control problems with final state constraints, *Oil & Gas Science and Technology—Rev. IFP*, DOI: 10.2516/ogst/2009020, 2009.

Sundström, O. and Guzzella, L., A generic dynamic programming MATLAB function, *IEEE Conference on Control Applications*, Saint Petersburg, Russia, 2009.

5

Purge Scheduling for Dead-Ended Anode Operation of PEM Fuel Cells

Jason B. Siegel
University of Michigan

Anna G. Stefanopoulou
University of Michigan

Giulio Ripaccioli
University of Siena

Stefano Di Cairano
Ford Motor Company

5.1 Introduction

Advances in the design and control of polymer electrolyte membrane (PEM) fuel cells (FCs) are necessary to significantly improve their durability and reduce their cost for large-scale commercial automotive

applications. Degradation has been observed and is associated with undesired reactions which occur during load following and start-up conditions. This degradation is accelerated by the varying spatio-temporal profiles caused by the local buildup of liquid water [1,2]. It is therefore desirable to avoid the accumulation of water and flooding or plugging of the channels, due to its deleterious effects on performance and stack life. Anode channel plugging, for example, can induce hydrogen starvation and, given the right conditions, trigger cathode-carbon oxidation and loss of active catalyst area [3–5].

To avoid excess water accumulation in the anode channel, high hydrogen flow rates are pushed through the stack, resulting in the flow-through anode (FTA) configuration. To increase the fuel utilization, a re-circulated anode (RCA) architecture is used. In this configuration water is removed from the gas stream leaving the channel and the hydrogen is recirculated back into the stack inlet, where it is combined with additional hydrogen from the storage medium and then humidified. This practice leads to higher cost and lower power densities due to the need for external humidification and anode recirculation loops. To achieve competitive cost and power density objectives, we develop a modeling and control methodology for water management in dead-ended anode (DEA) systems which can operate at low hydrogen flow rates without external humidification, but suffer from large spatial distributions of water (dry inlets and flooded outlets).

Liquid water formation, can appear under various conditions leading to stationary, cyclo-stationary, or erratic patterns inside PEMFCs [1,6] making it difficult to statistically assess or physically model the impact on degradation due to lack of repeatable data. In this chapter we develop and show plate-to-plate experiments with controlled flooding patterns [7], which are used to parameterize models of two-phase (water liquid and vapor), dynamic and spatially distributed transport phenomena inside a low-temperature fuel cell. We then reduce the model complexity first by deriving ordinary differential equations (ODEs) that capture the water front evolution inside the FC porous media and channels. The simplification reduces the computational effort, allows efficient parameterization, and provides insight for further simplification that leads to an explicit control law using model predictive control (MPC) methodology. The control objective is scheduling the anode purge period and duration to reduce wasted fuel, to prevent dehydration of the membrane, and to maximize the thermodynamic efficiency (terminal voltage at a given load) of the fuel cell.

5.2 Background: PEMFC Basics

Proton exchange membrane, also called PEMFCs, are electrochemical energy conversion devices that produce electricity by combining hydrogen and oxygen, typically from the air, to form water and a small amount of heat. The heart of PEMFCs is their polymer electrolyte membrane. The membrane provides a barrier to keep the hydrogen and oxygen separated. It must conduct protons easily, yet be electronically insulating to force the electrons though an external circuit and provide useful work. One such membrane material is Nafion© manufactured by Dupont, another competing product is provided by Gore. These polymer membranes have low melting temperatures, which limit operating temperature below 100°C. PEMFCs have several attributes, including low operating temperature and high efficiency (typically 50–70% for the fuel cell stack and 40–60% for the overall system), which make them good candidates for automotive and portable power applications.

The basic structure of the PEMFC is shown in Figure 5.1. The fuel and oxygen (air), are delivered across the active area through a series of channels. These channels are typically machined into the backplanes which are a conductive material so that the electrons can be transfered to the current collectors and hence complete an electric circuit. The ratio of channel width to rib (contact) width is an important design parameter affecting fuel cell performance. The gas diffusion layer (GDL) is a porous material used to distribute the reactant gases from the channel to the catalyst surface in the area under the ribs, and channels evenly. The GDL is typically formed from an electrically conductive carbon paper or cloth material. The GDL is also designed to promote the removal of product water

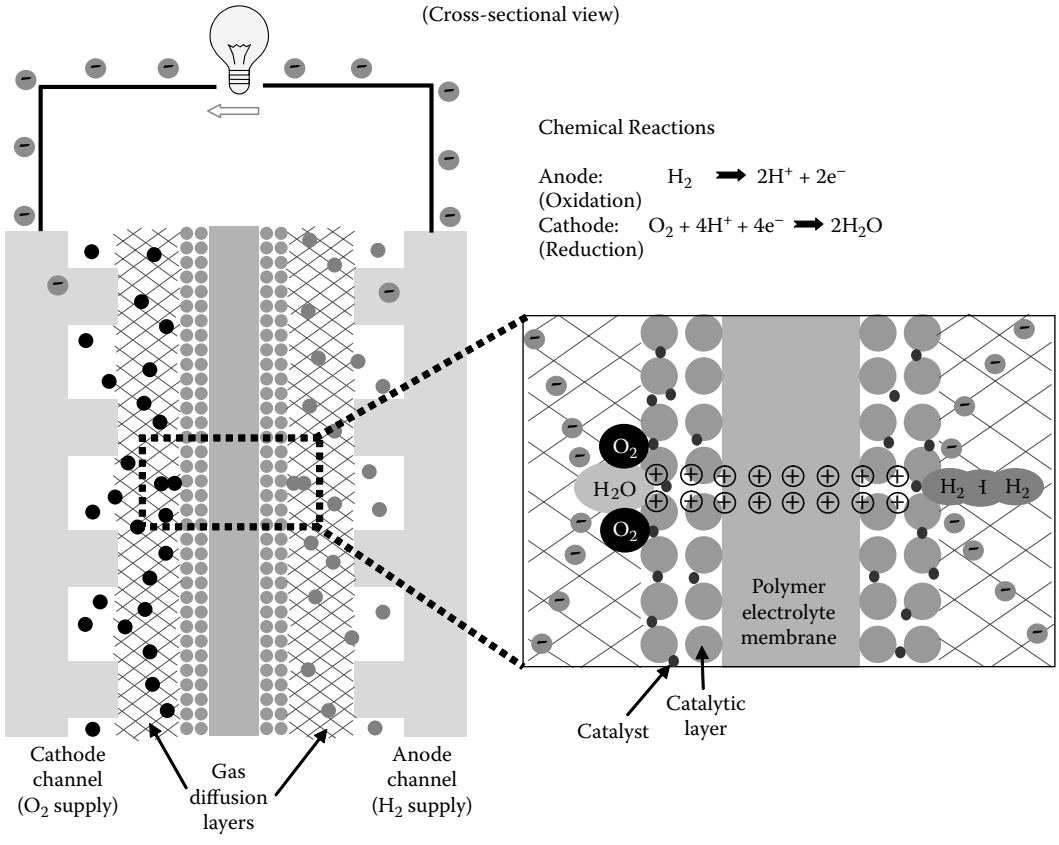

FIGURE 5.1 Basic fuel cell structure (not to scale). (Adapted from McCain, B. A., A. G. Stefanopoulou, and I. V. Kolmanovsky, *Chemical Engineering Science*, 63, 4418–4432, 2008.)

from the catalyst area, by treatment of the carbon with a hydrophobic coating such as Teflon. Finally, the catalyst layer (CL) contains platinum particles supported on a carbon structure. The CL is the place where the reaction takes place inside the fuel cell, and in order for the reaction to proceed on the cathode all three reactants, protons, oxygen, and electrons must be able to reach the Pt particle. Protons are conducted through the Nafion membrane material, electrons through the carbon support structure, and oxygen gas through the pore space. Therefore, each Pt particle must be in contact with all three portions of the cell [9]. A thin micro-porous layer (MPL) can also be inserted between the GDL and CL to increase the electrical contact, and aid in water removal from the catalyst or membrane hydration [10].

The membrane is a polymer which absorbs water, and the membrane water content λ_{mb}, defined as the number of moles of water per mole of SO_3H in the membrane, is a critical parameter for describing both proton transport, and the permeation of molecular species through the membrane. As the water content of the membrane increases, both the proton conductivity and the rate of gas permeation through the membrane also increases. Increased proton conductivity is good for fuel cell efficiency, but increased permeability increases the rate of molecular crossover through the membrane, which first, lowers the fuel cell efficiency and second, results in excess accumulation of water (plugging) and nitrogen (blanketing) of the anode channels, displacing or blocking hydrogen from reaching the catalyst sites. Modeling and managing this accumulation is the subject of this chapter.

5.3 Control of Fuel Cell Subsystems

The control of PEMFC systems can be organized into three main categories, each of which evolve on different time-scales, the fastest is reactant gas supply (air or oxygen and hydrogen), followed by thermal management, and water management. Although the control objectives, and dynamic behavior of each subsystem are tightly coupled, for example, excess reactant supply may be used to carry heat and product water out of the cell, these divisions provide a useful modeling framework. In this article we focus on water management, which has not been covered extensively in the control literature or studied by the control society. It is however considered currently as on of the barriers in commercialization of low-temperature PEMFC stacks [11]. First, we present the overall stack management system in order to provide adequate perspective and motivate the water management problem.

Figure 5.2a shows a basic fuel cell control architecture with many of the common control actuators for the reactant supply systems, hydrogen on the anode and air on the cathode. Although some small

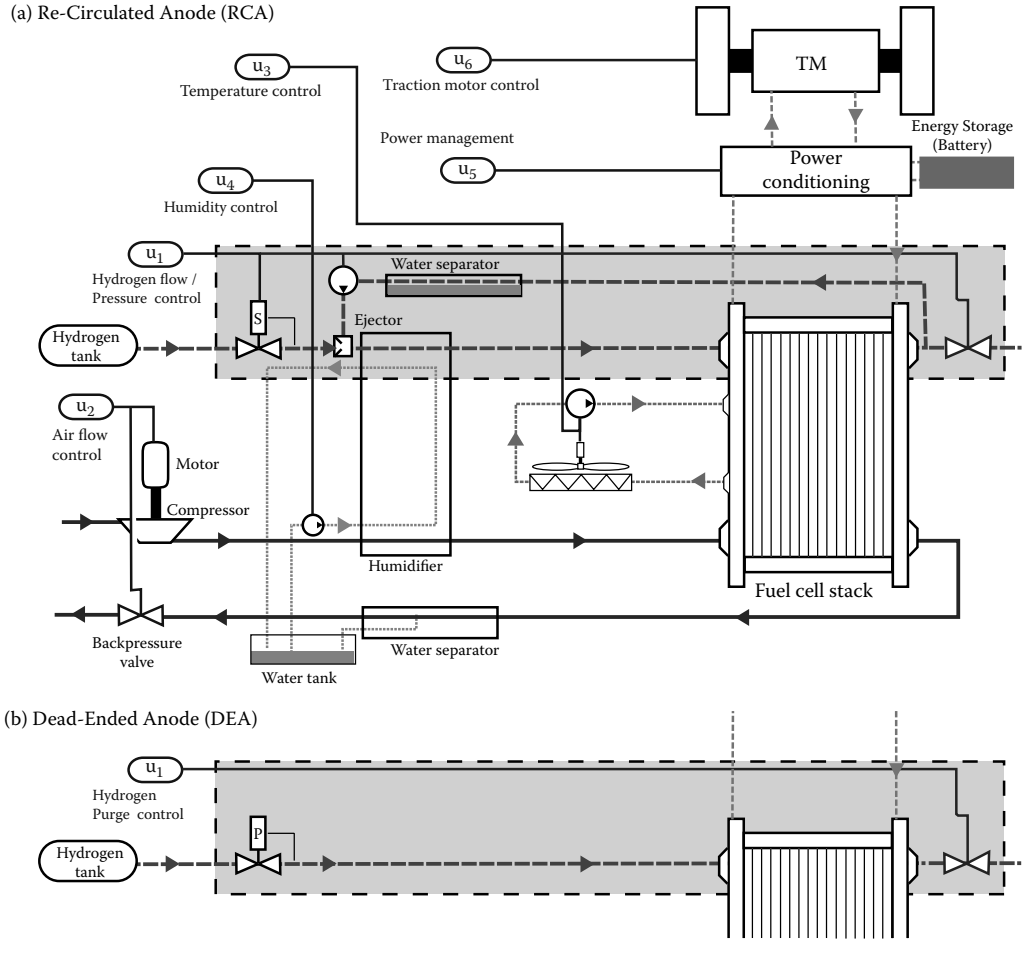

FIGURE 5.2 Fuel cell control actuators and sub-systems. (Modified from Pukrushpan, J. T., A. G. Stefanopoulou, and H. Peng, *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis and Feedback Design*. New York: Springer, 2000.) (a) Dashed gray box shows hardware required for RCA. (b) Simplified hydrogen delivery system for pressure regulated DEA, which reduces or removes the need for inlet humidification, ejector, water separator, and recirculation plumbing.

portable fuel cell systems are low-pressure ambient air breathing, which rely on free convection to deliver the oxidant, most automotive applications use forced air with a scroll or centrifugal compressor in order to increase the oxygen pressure in the channels and hence the power density of the system [12]. The actuator u_2 controls the supply rate and back-pressure of the air to the cathode side of the fuel cell. It has been shown that when the compressor is driven by the fuel cell, there exists a performance limitation on the rate of power increase in fuel cell stack [13] in order to prevent oxygen starvation. Various control techniques have been used to model and handle the constraints of the air compressor and manifold filling dynamics [12,14], including Feedback Linearization [15], and a reference governor [16]. In addition to buffering the fuel cell from damaging increases in load current, excess air supplied to the cathode aids the removal of liquid water from the channels [1,17,18].

Since excess air is supplied, typically 2–3 times the amount required to support the reaction, there is a very high flow rate of gas though the cathode channels and humidification of the incoming air is required to prevent drying the membrane. Bubblers, hotplate injection, and other novel humidifiers are employed for the inlet air humidification. The water temperature and flow rate, indicated by actuator u_4, though a membrane-type humidifier may be used to control the relative humidity of the incoming air [19,20]. This subsystem is tightly coupled with the air flow and thermal management subsystems [21], since the coolant water exiting the stack is then fed into the humidifier. The coolant flow rate and fan speed, u_3, can be used for control of the cooling system.

The overarching difficulty in the subsystem level control paradigm is that the control of air flow rate, cathode pressure, cathode inlet relative humidity, and stack temperature are all tied to the higher level control objective of water management. Water management is a critical issue for PEMFC operation to ensure long stack life and efficient operation. Two of the limiting factors which prevent the mainstream adoption of PEMFCs are cost and durability or stack lifetime [11]. One of the major factors affecting PEMFC durability is water, specifically the formation of liquid water inside the cell [1] and the wet–dry cycling of the membrane [22,23]. The formation of liquid water can block reactants from reaching the catalyst leading to starvation and carbon corrosion [3,5,24], a permanent degradation of the cathode support structure. Conversely, drying of the membrane increases protonic resistance which yields lower cell efficiency as shown in Equation 5.55. There needs to be sufficient water content in the membrane (high λ_{mb}), so that proton conduction through the membrane is easy, but flooding and channel plugging is undesirable, hence the need for water management [2,4,25]. This is where control engineering can provide the necessary tools to help develop and commercialize this technology.

5.4 Anode Water Management

Flow-through operation is used on both the anode and cathode of most laboratory/experimental hydrogen PEMFC systems. However, the fuel utilization of FTA operation is too low for commercial and portable systems. Fuel utilization is the rate of fuel consumption divided by the rate of fuel delivery, $U_{fuel} = (I_{fc}/(nF))/v_{fuel}$ [9], where I_{fc} is the fuel cell current in amperes (A), $n = 2$ is the number of electrons taking part in the reaction, F = 96,400 C mol^{-1} is Faraday's constant and v_{fuel} is the hydrogen delivery rate in mol s^{-1}. The fuel stoichiometry, $\lambda_{H_2} = 1/U_{fuel}$, is the inverse of utilization. The anode reactant subsystem, shown in Figure 5.2a, uses a recirculation loop to recycle excess hydrogen back through the fuel cell stack increasing the fuel utilization. However, the RCA requires hydrogen grade plumbing and hardware such as, an ejector/blower, water separator, and hydrogen humidification. These components add weight, volume, and expense to the system [26,27]. Note that the water must be removed from the gas exiting the anode before it goes to the ejector and then the dry fuel supplied to the anode must be rehumidified to prevent over-drying of the membrane.

Although the RCA subsystem can remove water from the gas stream, purging is still required to handle the nitrogen. A small amount of nitrogen crosses though the membrane, driven by the gradient in partial pressure from the air fed in the cathode. Over time this nitrogen accumulates in the anode feed system,

which dilutes the hydrogen fuel in the anode [28,29]. The dilution of H_2 lowers the fuel cell terminal voltage, and hence the efficiency. Therefore, the anode recirculation system needs to be purged at certain intervals to remove the accumulated inert gas.

We consider the use a DEA fed by dry hydrogen, shown in Figure 5.2b, which is pressure regulated to maintain anode pressure and supply exactly the amount hydrogen needed to support the reaction $\lambda_{H_2} = 1$. DEA operation does not have as stringent requirements as the RCA system on hydrogen inlet humidification due to the lower flow velocity in the channels; the water crossing through the membrane is enough to self-humidify the hydrogen fuel. The use of a pressure regulator, instead of a mass flow controller, and lack of water separator, hydrogen ejector [30,31] or blower for the recirculation loop and finally anode inlet humidification yields a system with lower cost and volume. In DEA operation, the binary control signal, $u_1 \in \{0, 1\}$, opens the downstream solenoid valve, causing a brief high-velocity flow through the anode channel as the pressure regulator opens in attempt to maintain the system pressure. The high velocity flow aids in the removal of liquid water droplets [1,18], which in the case of RCA would remain stationary due to the lower gas velocity. Several anode configuration and practical aspects of purging versus flow-through have been considered experimentally in [32–34]. In this work, we focus on a mathematically derived model-based scheduling of anode purges, both the interval between purges and duration of the purge, for water management and to yield high fuel utilization.

Similar to the RCA system discussed above, nitrogen and water crossover is a concern. In a DEA system the nitrogen is pushed toward the end of the channel, by the flow of reactants, where it accumulates. The accumulating N_2 forms a blanket, which completely displaces hydrogen from reaching the catalyst layer, effectively shutting down the power production from the affected region of the cell [35]. Water vapor gradients between the humidified cathode and the dry fed anode also drive excess water into the anode, which can cause significant liquid water accumulation. Unlike water vapor whose maximum partial volume is dictated by temperature, liquid can fill the entire free space in the channels, as shown in Figure 5.3 and will be discussed later. This liquid water accumulation in the channel water blocks the flow of reactants, which is referred to as channel plugging, and stops the production of power in the affected regions of the cell.

Gravity, and the gas velocity, driven by consumption of hydrogen, both pull the nitrogen and water, which are heavier molecules than hydrogen, toward the bottom of the channel. As the mass accumulation continues, a stratified pattern develops in the channel with a hydrogen-rich area sitting above a hydrogen-depleted region, as shown in Figure 5.4. The boundary between these regions is a time-varying front, which proceeds upwards toward the inlet [35]. The mass accumulation physically blocks hydrogen gas from reaching the anode catalyst sites, which is the mechanism for the experimentally observed, and recoverable voltage degradation [7,28,36]. Therefore, purges of the anode channel volume are necessary to clear the reaction product and inert gas from the channel. After the purge, the catalyst area contributing to the reaction increases, and hence the measured voltage increases.

Using cell voltage and input/output measurements such as temperature and pressure we aim to develop a model-based H_2 purge schedule, which avoids both anode drying and flooding conditions, since both conditions lead to recoverable voltage losses. With this PEMFC model, the estimation of liquid flow into the channel and nitrogen crossover rate, and hence the accumulation of both liquid water and nitrogen in the anode channel is used both for prediction of voltage drop, and as an indication of the level of flooding and nitrogen blanketing in the fuel cell. The combination of measurements such as of fuel cell voltage, or impedance, can also be used to add robustness to the open-loop model predictive methodology by estimating the internal states and membrane humidity, λ_{mb} [37].

Although thresholds based on time, Amp hours, or voltage could be used to initiate anode purging to prevent excessive voltage drop, these schemes do not take into account the transient voltage drop during an increase of load current, or dynamic evolution of water and nitrogen fronts in the anode channel. One could use the experimentally identified liquid water crossover rate [7] to find the rate of liquid water accumulation in the DEA channel [38]. Unfortunately, this method is not robust to membrane and GDL aging, and requires extensive parameterization. To develop a purging scheme based on a voltage

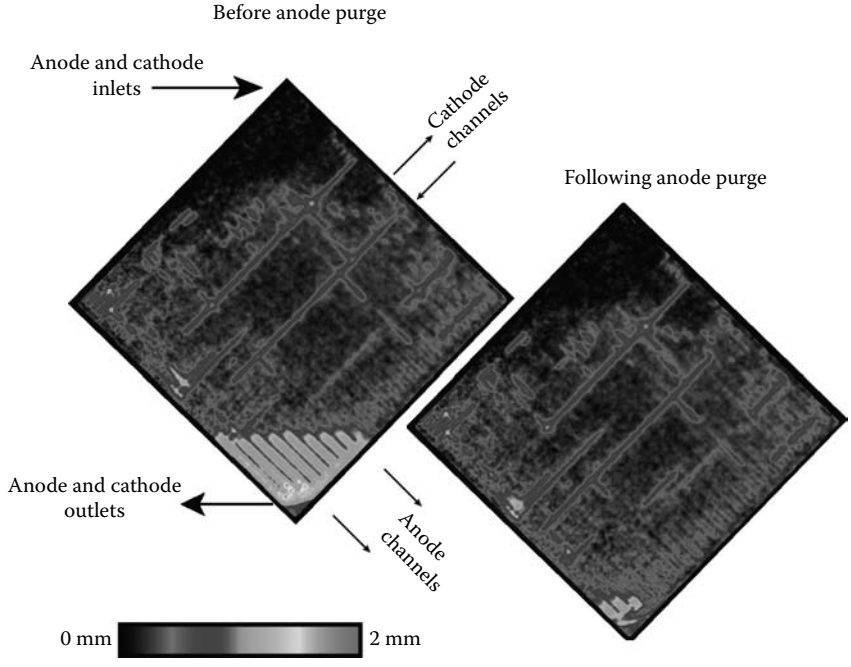

FIGURE 5.3 Neutron images of the fuel cell active area before and after anode purge events, indicating the actual cell orientation. The cell was operated at 566 mA cm^{-2}, 55°C, with fully humidified air at a stoichiometry of 200 % for cathode and dry hydrogen supplied to anode. 12% of the anode channel volume is filled with liquid water prior to the purge, and hence the fuel cell active area is reduced by 12%. The top 10% of the area is very dry due to the hydrogen supply, and also does not produce as much current as the middle region of the cell. (Adapted from Siegel, J. B., et al. *J. Electrochem. Soc.*, 155, pp. B1168–B1178, 2008.)

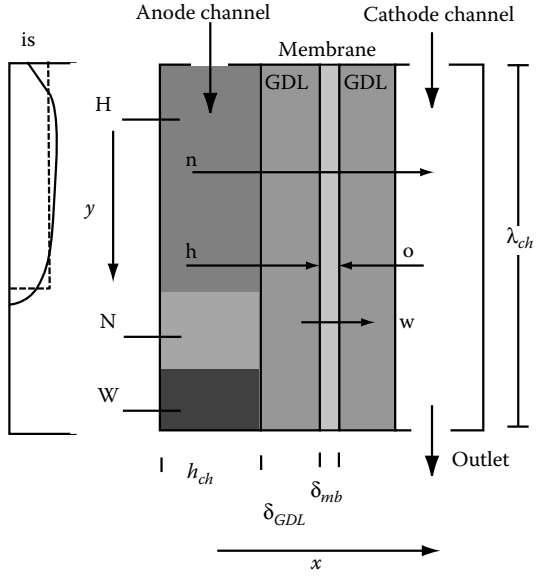

FIGURE 5.4 The one-dimensional FC modeling domain, along the *x*-axis, which denotes the through membrane direction. The *y* axis denotes the distance along the channel from inlet to outlet (not drawn to scale). This caricature shows the stratification of product water, inert nitrogen and the displacement of hydrogen from the bottom portion of the cell. The distributed current density, $i_{fc}(y)$, along the channel is shown to the left of this figure; for a realistic profile (solid line), and our approximation, namely the apparent current density (dashed line).

threshold can be labor intensive, requiring many experiments to develop a mapping for the correct voltage at which to purge under all possible operating conditions (temperature, pressure, load current, cathode inlet relative humidity, and cathode flow rate). Additionally, if the thresholds are developed with reasonable safety margins to account for the transient voltage drop during a step-up in current, they may yield overly conservative control laws, that is, purging too frequently, wasting fuel, and at the same time drying the membrane. Understanding, modeling, and predicting the front evolution and overall dynamics in DEA FC would allow judicious choice of purging interval and duration. Better purging strategy can reduce the H_2 wasted during purges and avoid over-drying the membrane. The advantage of developing a simple model is that simulation of the system can be performed in real time/on-line, and MPC or observer-based feedback algorithms can be utilized.

5.5 One Dimensional, Channel to Channel, GDL and Membrane Model

From the most simple model, a static polarization curve, shown in Equation 5.48, to fully dynamic 3-D multiphase models [39], the common objective is prediction of the fuel cell terminal voltage or performance. A review of fuel cell modeling approaches, can be found in [40]. Modeling reactant (gas) transport from the channel, through the GDL to the catalyst layer at surfaces of the membrane is critical for predicting the performance of the PEMFC, but equally important is the description of product water removal from the cathode catalyst surface. Liquid water can block the catalyst surface, reducing the number of active catalyst sites, or fill some of the pore spaces in the GDL inhibiting the flow of reactant gas to the CL.

Product water, which diffuses through the membrane back to the anode side will condense and accumulate in the anode channel, displacing hydrogen, as shown in the neutron imaging of Figure 5.3, and preventing it from reaching the anode catalyst layer which stops the production of power from the affected region of the cell. Under galvanostatic, or constant current operation the remaining "hydrogen-rich" regions of the cell must support a larger current density (A cm^{-2}), to account for the area which is nolonger active. Hence, the notion of apparent current density can be used to capture this effect [35,41]. Calculation of the apparent area from the anode channel masses is shown in Section 5.9.1. The schematic representation of anode channel filling with liquid water and nitrogen is shown in Figure 5.4. Since water condenses in both the GDL and channels, the water transport between the membrane and GDLs, and from the GDL to channels, determines the rate of water accumulation in the anode. Therefore, we consider a one-dimensional (1-D) modeling approach, for the GDL, from channel to channel [41].

A lumped volume approach is used for the channels, which constitutes a mass balance, described in Section 5.5.2. Similar work using a lumped channel model is shown in [42], where the PDEs describing reactant transport in the GDLs were further simplified and approximated by a single volume each, with the associated ODEs. Their model was tuned to represent the dynamic behavior of a Ballard Nexa PEMFC stack, which is auto-humidified, and as a result also utilized a saturated cathode feed gas supply. The limitations of a lumped volume channel model for capturing two-phase liquid and water vapor system behavior become apparent when subsaturated or dry reactants are fed to the cell. In this case, a large spatial gradient in partial pressure of water vapor develops along the gas flow path. This can be seen in Figure 5.3, where the liquid water accumulation occurs at the bottom of the cell, and a dry area corresponding to the darker region occurs near the top, at the inlet. The water and nitrogen accumulation creates a distribution of current density, which changes the membrane water content and affects the rate of water and nitrogen crossover. The importance of along the channel modeling for capturing water crossover and cell performance has been demonstrated in [43]. The trade-off between model accuracy and computational complexity, 1-D versus 2-D versus 3-D, is a difficult engineering challenge for the control of PEMFC systems. Our 1-D through the membrane modeling approach is a good compromise

since the apparent current density, shown in Figure 5.4, captures some of them along the channel-effects without increasing the computational complexity.

5.5.1 Overview of Modeling Domains

The fuel cell model is divided into five modeling domains to capture the relevant dynamics, corresponding to the five separate regions of the fuel cell in the through membrane direction (x); the anode channel, the cathode channel, the membrane, and the anode and cathode GDLs. Each channel domain contains three dynamic states (the mass in the cathode channel $\{m_{O_2,ca,ch}, m_{N_2,ca,ch}, m_{w,ca,ch}\}$, and anode channel $\{m_{H_2,an,ch}, m_{N_2,an,ch}, m_{w,an,ch}\}$), and the remaining domains have one state each (cathode GDL liquid water front location $x_{fr,ca}$, anode GDL front location $x_{fr,an}$, and membrane water content λ_{mb}), for a total of nine modeled states. Water states in the GDL and membrane are initially defined in Sections 5.5.4 and 5.5.5 as PDE equations, and further simplified to derive the states $x_{fr,ca}$, $x_{fr,an}$, and λ_{mb} in Section 5.7. Since the time constants for gas diffusion are much faster than the liquid states in the GDL, the spatial distribution of the gas species in the GDL can be approximated with the steady-state profiles [44]. The modeling domains and the interactions between each domain is shown schematically in Figure 5.5. The fuel cell stack load current I_{fc} is considered as a measured disturbance to the plant $D1$.

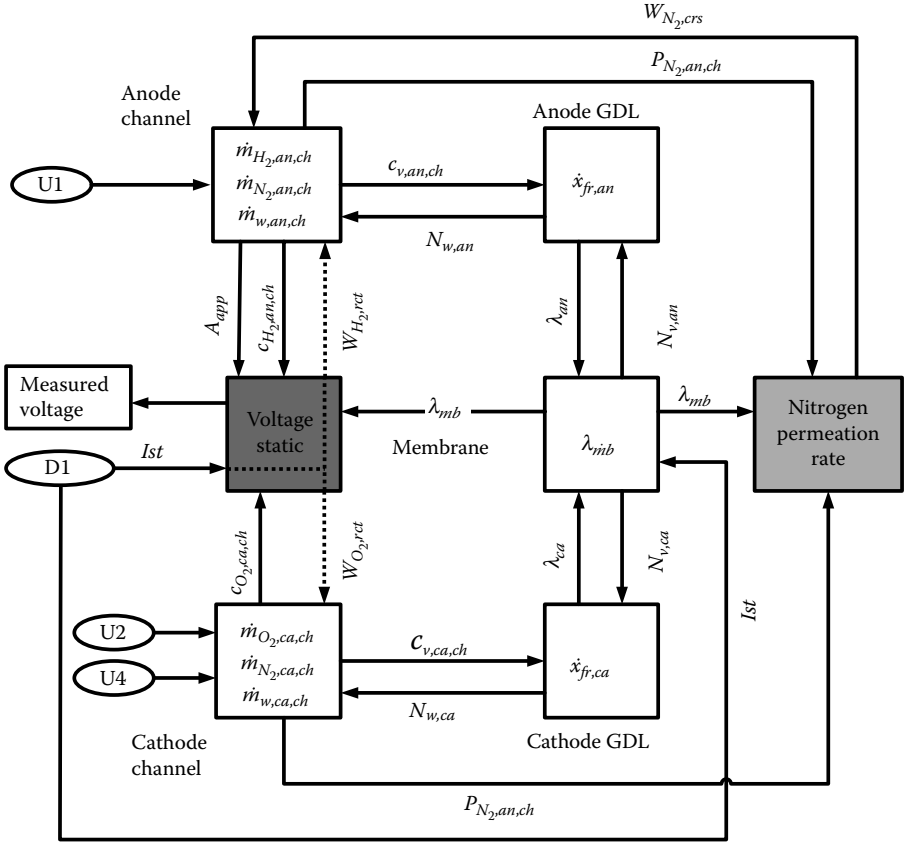

FIGURE 5.5 Schematic diagram showing the subsystem interconnections and modeled states. The anode channel has three states, the cathode channel has three states, one state for the anode GDL front location $x_{fr,an}$, one state for the cathode GDL front location $x_{fr,ca}$, and one state for membrane water content λ_{mb}.

5.5.2 Anode Channel Model

The lumped volume equations constitute a mass balance for each of the three constituent gases in the channel (H_2/O_2, N_2, and H_2O) similar to [41].

Three dynamic states are used in the channel,

$$\frac{dm_{i,an,ch}}{dt} = W_{i,an,in} + W_{i,an,GDL} - W_{i,an,out}, \tag{5.1}$$

where $i \in \{H_2, N_2, w\}$ corresponding to hydrogen, nitrogen, and water.

Liquid and vapor are combined into one state in the channel, it is assumed that vapor and liquid in the channel are at a phase transition equilibrium, therefore

$$m_{v,an,ch} = \min\left(m_{w,an,ch}, \frac{P_{sat}(T)\ V_{an,ch}\ M_v}{R\ T}\right) \tag{5.2}$$

where M_v is the molar mass of water, and $P_{sat}(T)$ is the temperature-dependent saturated vapor pressure [45]. The remaining water is considered to be in the liquid phase

$$m_{l,an,ch} = m_{w,an,ch} - m_{v,an,ch}. \tag{5.3}$$

The anode inlet gas flow rate is assumed to be dry hydrogen supply, therefore,

$$W_{N_2,an,in} = W_{w,an,in} = 0 \quad \text{and} \quad W_{H_2,an,in} = W_{tot,an,in}. \tag{5.4}$$

The total flow into the anode, $W_{tot,an,in}$, is calculated using Equation 5.10, and the anode inlet pressure $P_{an,in}$ which is a constant, since it is set via a pressure regulator.

Assuming steady-state conditions for the gases in the GDL [44], the hydrogen consumption at the catalyst surface accounts for the flux of hydrogen leaving the anode channel into the GDL, which is

$$W_{H_2,an,GDL} = -W_{H_2,rct} = -\frac{i_{fc}}{2F}\ M_{H_2}\ A_{fc}. \tag{5.5}$$

The combined liquid and vapor flux entering the GDL from the channel $W_{w,an,GDL}$ is found using Equation 5.45. Nitrogen permeation across the membrane, as a function of the channel partial pressures of nitrogen,

$$\begin{aligned} W_{N_2,an,GDL} &= -W_{N_2,ca,GDL} \\ &= \frac{k_{N_2,perm}\ M_{N_2}\ A_{fc}}{\delta_{mb}}\left(P_{N_2,ca,ch} - P_{N_2,an,ch}\right). \end{aligned} \tag{5.6}$$

The nitrogen permeation rate, $k_{N_2,perm}(T, \lambda_{mb})$, is a function of temperature and membrane water content from [28];

$$\begin{aligned} K_{N_2}(T, \lambda_{mb}) = \alpha_{N_2}&(0.0295 + 1.21 f_v - 1.93 f_v^2) \times 10^{-14} \\ &\times \exp\left[\frac{E_{N_2}}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right], \end{aligned}$$

where $E_{N_2} = 24000\ \text{J mol}^{-1}$, $T_{ref} = 303$, R is the universal gas constant, and $f_v(\lambda_{mb})$ is the volume fraction of water in the membrane (Equation 5.42).

The partial pressures of each gas is calculated from the mass using the ideal gas law,

$$P_{N_2,an,ch} = \frac{m_{N_2,an,ch}\ R\ T}{M_{N_2}\ V_{an,ch}}, \tag{5.7}$$

where $V_{an,ch}$ is the anode channel volume, and the total channel pressure is given by the sum of the partial pressures $P_{an,ch} = P_{H_2,an,ch} + P_{N_2,an,ch} + P_{v,an,ch}$.

The gas flow leaving the anode channel is zero during normal operation. During an anode purge the gas velocity is larger, but still within the laminar flow regime.

The individual gas species flows leaving the channel are calculated from the total flow through the outlet orifice by multiplying with the vector of mass fractions, x_j,

$$\begin{bmatrix} W_{H_2,an,out} \\ W_{N_2,an,out} \\ W_{w,an,out} \end{bmatrix} = x_j \, W_{tot,an,out} \tag{5.8}$$

where $W_{tot,an,out}$ is given by Equation 5.10 if $u_1 = 1$, and zero otherwise. The subscript $j = 1$ corresponding to the anode channel when $P_1 = P_{an,ch} \geq P_2 = P_{an,outlet}$ and $j = 2$ otherwise, which indicates back flow from the outlet manifold. When liquid water is present in the anode channel, we assume that it can cover the outlet orifice, and the gas mixture parameters are replaced with those corresponding to liquid water in Equation 5.10, until the liquid is cleared, that is,

$$x_1 = \begin{cases} \left[x_{H_2,an,ch},\ x_{N_2,an,ch},\ x_{v,an,ch}\right]^T, & m_{l,an,ch} = 0 \\ \left[0,\ 0,\ 1\right]^T, & m_{l,an,ch} > 0 \end{cases} \tag{5.9}$$

The density of the gas mixture is given by $\rho_{an,ch} = m_{tot,an,ch}/V_{an,ch}$.

The total mass flow into and out of the fuel cell channel volumes is given by Equation 5.10, where $C_{turb} = 0.61$ is the dimensionless discharge coefficient under turbulent conditions, D_h is the hydraulic diameter in m, A is the area of the orifice in m^2, $R_t = 9.33$ is the critical value from [46], ρ is the density of the gas in kg m^{-3}, $\nu = \mu/\rho$ is the kinematic viscosity in $m^2\ s^{-1}$ and P_1, P_2 are the orifice upstream and downstream pressures in Pa. The dynamic viscosity of the gas mixture μ is calculated from the mole fraction y of gas species in the channel [9].

$$W_{tot} = \begin{cases} A\rho_1 \left(C_{turb} \sqrt{\dfrac{2}{\rho_1}|P_1 - P_2| + \left(\dfrac{\nu_1 \, R_t}{2 \, C_{turb} \, D_h}\right)^2} - \dfrac{\nu_1 \, R_t}{2 \, D_h} \right), & \text{if } P_1 \geq P_2 \\ -A\rho_2 \left(C_{turb} \sqrt{\dfrac{2}{\rho_2}|P_1 - P_2| + \left(\dfrac{\nu_2 \, R_t}{2 \, C_{turb} \, D_h}\right)^2} - \dfrac{\nu_2 \, R_t}{2 \, D_h} \right), & \text{if } P_1 < P_2 \end{cases} \tag{5.10}$$

5.5.3 Cathode Channel Model

The control inputs u_2 and u_4, shown in Figure 5.2, affect the amount of air entering the cathode channel, the channel pressure, and the relative humidity of the air which is supplied. For simplicity, we map these control inputs to the parameters; $\lambda_{O_2,ca}$, the oxygen stoichiometry at the cathode inlet, $P_{tot,ca,in}$, the cathode inlet pressure, and $RH_{ca,in}$, the relative humidity air supplied to the cathode. In a physical system these values would be the dynamic outputs of the blower [12] and humidifier [19].

Three dynamic states are used for the cathode channel, corresponding to a mass balance

$$\frac{\mathrm{d}m_{i,ca,ch}}{\mathrm{d}t} = W_{i,ca,in} + W_{i,ca,GDL} - W_{i,ca,out}, \tag{5.11}$$

where $i \in \{O_2, N_2, w\}$ corresponding to oxygen, nitrogen, and water.

The oxygen flow into the cathode is calculated from the stoichiometric ratio

$$W_{O_2,ca,in} = \lambda_{O_2,ca} \, \frac{i_{fc}}{4F} \, A_{fc} \, M_{O_2}. \tag{5.12}$$

Using the mole fraction of dry air, $OMF_{ca,in} = 0.21$, we can calculate the nitrogen flow into the cathode

$$W_{N_2,ca,in} = \frac{M_{N_2}}{OMF_{ca,in} \, M_{O_2}} \, W_{O_2,ca,in}. \tag{5.13}$$

Finally using the vapor pressure at the cathode inlet, $P_{v,ca,in} = RH_{ca,in}\, P_{sat}(T)$, we can calculate the flow of water vapor into the cathode,

$$W_{v,ca,in} = \frac{P_{v,ca,in}\, M_v}{OMF_{ca,in}\,(P_{tot,ca,in} - P_{v,ca,in})\, M_{O_2}} W_{O_2,ca,in}. \tag{5.14}$$

Oxygen consumption at the catalyst surface accounts for the flux of oxygen leaving the cathode channel into the GDL

$$W_{O_2,ca,GDL} = -W_{O_2,rct} = -\frac{i_{fc}}{4F}\, M_{O_2}\, A_{fc} \tag{5.15}$$

and the nitrogen crossover rate is given by Equation 5.6.

The cathode channel outlet equations are similar to the anode outlet equation and skipped for brevity, a fixed cathode outlet pressure is assumed.

5.5.4 Water Transport through the Gas Diffusion Layer

The critical GDL states are the two water phases, which can be described by two coupled second-order partial differential equations (PDEs), for the 1-D time-varying liquid water distribution through the gas diffusion layer GDL [41,44,47], along the x-direction as shown in Figure 5.6. The model was originally developed in [41], and is reviewed here briefly. Liquid water transport in the GDL is governed by the

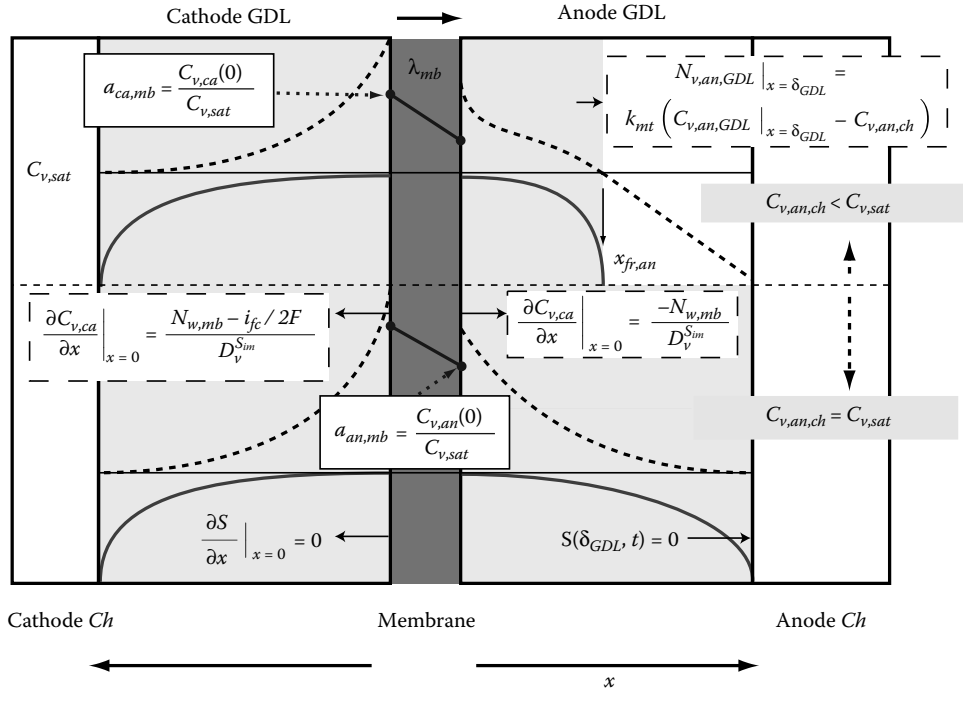

FIGURE 5.6 Water transport in the GDL and membrane. Dashed lines indicate the vapor distribution in the GDL, c_v, and solid blue lines represent the reduced water saturation, S. The boundary conditions (BCs) for the PDE models are shown, and arrows indicate positive x-direction in each region. The water distributions are shown for two distinct cases; saturated (bottom half) and sub-saturated (top half) anode channel conditions. In the case of subsaturated anode channel conditons, the transition between two phase (water liquid and vapor) and single phase (vapor only) is indicated by the front location, $x_{fr,an}$, as the point where $S \to 0$ in the GDL along the x-direction.

volume fraction of liquid water in the GDL

$$s(x,t) = \frac{V_l}{V_p}, \tag{5.16}$$

where V_l is the liquid water volume, and V_p is the open pore volume of the GDL. If there is a sufficient volume of water in the porous medium, termed the immobile saturation limit s_{im}, such that there exists connected liquid pathways, then liquid water can flow easily. When s falls below this critical value, there is no liquid flow through the GDL. The reduced water saturation $S(x,t)$, captures this phenomenon, where $S(x,t) = \frac{s(x,t)-s_{im}}{1-s_{im}}$, and $S = 0$ for $s < s_{im}$. Liquid flow in the GDL

$$W_l = -\epsilon A_{fc} \rho_l \frac{K}{\mu_l} S^3 \frac{\partial P_c}{\partial S} \frac{\partial S}{\partial x}, \tag{5.17}$$

is driven by the gradient in capillary pressure P_c. With the capillary pressure being

$$P_c = \frac{\sigma \cos(\theta_c)\sqrt{\epsilon}}{\sqrt{K}}(1.417\, S - 2.12\, S^2 + 1.263\, S^3), \tag{5.18}$$

where K is the absolute permeability, μ_l is the viscosity of liquid water, ϵ is the GDL porosity, σ is the surface tension between water and air, and θ_c is the contact angle of the water droplet on the GDL [48]. Other, more recent, capillary pressure models can be found in [49–51].

The PDE describing the liquid distribution water in the GDL is given by

$$\frac{\partial s}{\partial t} = -\frac{1}{\epsilon A_{fc} \rho_l} \frac{\partial W_l}{\partial x} - \frac{M_v}{\rho_l} r_v(c_{v,an}), \tag{5.19}$$

where M_v is the vapor molar mass, ρ_l is the density of the liquid water, and $r_v(c_{v,an})$ is the evaporation rate.

The liquid and vapor PDEs (Equations 5.19 through 5.21) are coupled through evaporation and condensation

$$r_v(c_{v,an}) = \begin{cases} \gamma\left(c_{v,sat}(T) - c_{v,an}\right) & \text{for } s > 0, \\ \min\left\{0, \gamma(c_{v,sat}(T) - c_{v,an})\right\} & \text{for } s = 0, \end{cases} \tag{5.20}$$

where γ is the volumetric condensation coefficient, $c_{v,sat}(T)$ is the vapor saturation concentration, and $c_{v,an}$ is the vapor concentration in the GDL.

The steady-state value is used for the distribution of water vapor in the GDL [44],

$$0 = \frac{\partial c_{v,an}}{\partial t} = \frac{\partial}{\partial x}\left(D_v^{sim} \frac{\partial c_{v,an}}{\partial x}\right) + r_v(c_{v,an}), \tag{5.21}$$

where $D_v^{s_{im}} = D_V D_{eff}(s = s_{im}, \epsilon)$ is the diffusivity of vapor inside the GDL porous medium when s is near s_{im} [44], D_V is the diffusivity of vapor in free space, and $D_{eff}(s,\epsilon)$ is the effective diffusivity [48] correction term.

The BCs complete the model of GDL water accumulation and transport. For $c_{v,an}(x,t)$, Neumann-type BCs are imposed at both sides. The channel (ch) boundary condition is

$$N_{v,an,GDL}\big|_{x=\delta_{GDL}} = k_{mt}\left(c_{v,an,GDL}\big|_{x=\delta_{GDL}} - c_{v,an,ch}\right), \tag{5.22}$$

where $c_{v,an,ch}$ is the vapor concentration in the channel, and $k_{mt} = Sh\, D_V / H_{ch}$ is the mass transfer coefficient [52]; related to the Sherwood number Sh, the free space diffusion coefficient for vapor in

hydrogen, D_V, and the channel height, h_{ch} which is the characteristic diffusion length. The slope of the water vapor distribution at the membrane is determined by the water vapor flux across the membrane

$$\left.\frac{\partial c_{v,an}}{\partial x}\right|_{x=0} = \frac{-N_{w,mb}}{D_v^{s_{im}}}, \tag{5.23}$$

where the membrane water molar flux $N_{w,mb}$ is governed by electro-osmotic drag and back diffusion [41], as shown in Equation 5.26.

For the liquid water PDE, mixed BC are again imposed. Specifically, water passing into the GDL from the membrane is assumed to be in vapor form due to the presence of a micro-porous layer, therefore $\left.\frac{\partial S}{\partial x}\right|_{x=0} = 0$. The liquid water flux from the GDL into the channel depends on the boundary condition at the GDL–channel interface. Liquid flows readily when there is sufficient water to form connected pathways, therefore we assume

$$S(\delta_{GDL}, t) = 0, \tag{5.24}$$

similar to [53]. Other possible models include using the liquid pressure in the channel [54] and could be considered in the future.

5.5.5 Membrane Water Transport

In order to model reactant and product transport, we finally come to the membrane, which is the heart of the PEMFC. The membrane provides a barrier to keep the hydrogen and oxygen separated. It must conduct protons easily, yet be electronically insulating to force the electrons through an external circuit and provide useful work. The most common membrane material Nafion, used in this work, is modeled widely in the literature. The membrane water content λ_{mb}, is defined as the number of moles of water per mole of SO_3H in the membrane. λ_{mb}, is a critical parameter for describing both proton transport, diffusion of water, electro-osmotic drag, and the permeation of the molecular species through the membrane.

The distribution of membrane water content is described by a PDE, the divergence of the water flux through the membrane

$$\frac{\partial \lambda_{mb}}{\partial t} = -\frac{EW}{\rho_{mb}} \frac{\partial N_{w,mb}}{\partial x}, \tag{5.25}$$

where $EW = 1100\,\mathrm{g\,mol^{-1}}$ is the equivalent weight of the membrane and $\rho_{mb} = 1.9685\ \mathrm{g\ cm^{-3}}$ is the membrane dry density. However, since the membrane is very thin, $\delta_{mb} = 25\,\mu\mathrm{m}$, it can be discretized in space and represented by a single ODE.

Membrane water flux, $N_{w,mb}$ (mol $\mathrm{cm^{-2}}$), from cathode to anode is calculated from the diffusion and electro-osmotic drag terms,

$$N_{w,mb} = -D_w(\lambda_{mb}, T)\frac{\partial \lambda_{mb}}{\partial x} - n_d(\lambda_{mb}, T)\frac{i_{fc}}{F}, \tag{5.26}$$

where $D_w(\lambda_{mb}, T)$ is water diffusion coefficient for water in the membrane (Equation 5.37), and $n_d(\lambda_{mb})$ is the coefficient of electro-osmotic drag (Equation 5.40) [55]; both of which are λ_{mb} and T dependent and increase with both membrane water content and temperature.

The BCs complete the description of transport in the membrane, coupling it with the GDL. Dirichlet BCs are imposed for λ_{mb}, at the left and right edges of the membrane, assuming the membrane is at equilibrium with the water vapor concentration in the GDL at the membrane surface, $c_{v,an}(0)$ and $c_{v,ca}(0)$. The equilibrium value, λ_{an}, is calculated using the membrane water uptake isotherm, $\lambda_{T,a}$ as shown in Equation 5.33 [41,55], and the water activity at the GDL–MB interface $a_{an,mb} = c_{v,an}(0)/c_{v,sat}$. The water flux leaving the membrane, $N_{w,mb}$ is the boundary condition for the GDL, as shown in Equation 5.23.

5.6 GDL Fronts Simplification

The liquid water profile forms a steep front inside the GDL. The steep drop in s as the transition between the two-phase and single-phase water areas in the GDL is the result of the sigmoidal function of the capillary pressure [48]. Other capillary pressure models [49–51], should also exhibit a similar behavior in the formation of a steep liquid water front and propagation mode [53]. The PDEs [8,41] describing water transport require a very fine discretization grid in order to accurately represent the front propagation. Adaptive grids developed to address the problem of groundwater transport [56] have been used to handle this type of problem, but they are not well suited for a reduced complexity control-oriented model. In this chapter, we leverage the nature of the sharp transition in the GDL liquid water volume. We define the two-phase front location in the GDL, along the x direction, in terms of an ODE, similar to [53], greatly reducing the computational complexity of the 1-D two-phase water model.

We assume that the liquid water propagates at a constant, tunable, volume fraction, s_*, which is slightly larger than the immobile limit s_{im}. In [57] the s_{im} value is estimated using neutron imaging data of the water accumulation. The caricature shown in Figure 5.7, with a square-shape two-phase front highlights the steep transition and the front propagation. Using the numerical solution of the physically derived PDEs in [8,41], and the experimental observations in [7] we infer that locally, liquid water first begins to accumulate in the GDL as shown in frame (Figure 5.7a). In the next frame (Figure 5.7b), the liquid water front has reached the GDL-channel boundary. Since there is little resistance to water entering the channel, it begins to spill out of the GDL and into the channel where it accumulates. Finally in frame (Figure 5.7c), liquid water fills the anode channel section completely and begins spreading along the y-direction, up the channel and against gravity.

In the following section we introduce the ODEs that describe the water dynamics in the membrane and the water front propagation in the GDLs. The proposed GDL model uses three nonlinear states (the anode, $x_{fr,ca}$, and cathode, $x_{fr,an}$, GDL water front location and the membrane water content, λ_{mb}) and three inputs (the anode and cathode channel vapor concentration, which are the states of the channel model, and the current density which is measured) to predict the dynamically evolving front locations in both anode and cathode side GDLs during flooding and drying as well as the dynamic changes in membrane water content. We present the detailed equations only for the anode and the membrane, and just highlight the modifications necessary for the cathode calculations.

5.7 Liquid Water Front Propagation in the GDL

The water dynamics in the GDL–MB–GDL unit model are governed by the membrane water content and the location of a liquid water front in the GDLs as follows. A single state can be used to model water

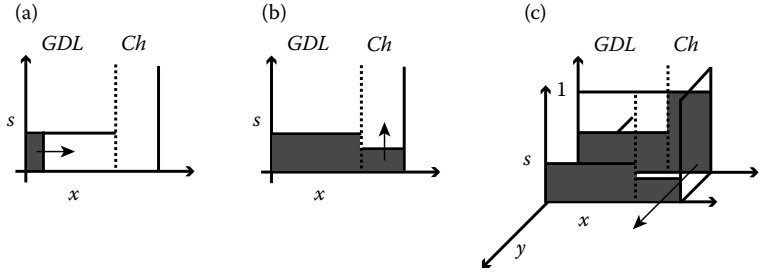

FIGURE 5.7 Evolution of liquid water fronts in the GDL and channel. In frame (a) liquid water fills the GDL up to $s = s_*$, then the two phase front propagates toward the channel. Next in (b) the liquid begins to accumulate in the channel. Finally in (c) once the channel section fills completely water begins to spread back up along the channel.

uptake into the membrane. Diffusion within the membrane is fast with respect to water uptake, therefore we can solve the diffusion and osmotic drag in each half of the membrane, using Equations 5.31 and 5.44, which yields a piecewise linear water content profile as indicated in Figure 5.8. The average membrane water content, λ_{mb}, is calculated from the water flux into the membrane from the cathode side, $N_{v,ca,mb}$, and out of the membrane from the anode side, $N_{v,an,mb}$,

$$\frac{d\lambda_{mb}}{dt} = K_{mb}(N_{v,ca,mb} - N_{v,an,mb}) \tag{5.27}$$

where $K_{mb} = EW/(\rho_{mb}\,\delta_{mb})$ is the membrane water uptake rate.

The location of the two phase (liquid–vapor) water front in the anode GDL, $x_{fr,an}$, is governed by the rate of water accumulation in the GDL that is condensing into the liquid phase, $N_{l,an}$. Hence, the front propagation is given by

$$\frac{dx_{fr,an}}{dt} = K_L \begin{cases} N_{l,an} & x_{fr,an} < \delta_{GDL} \\ \min(0, N_{l,an}) & x_{fr,an} = \delta_{GDL} \end{cases}, \tag{5.28}$$

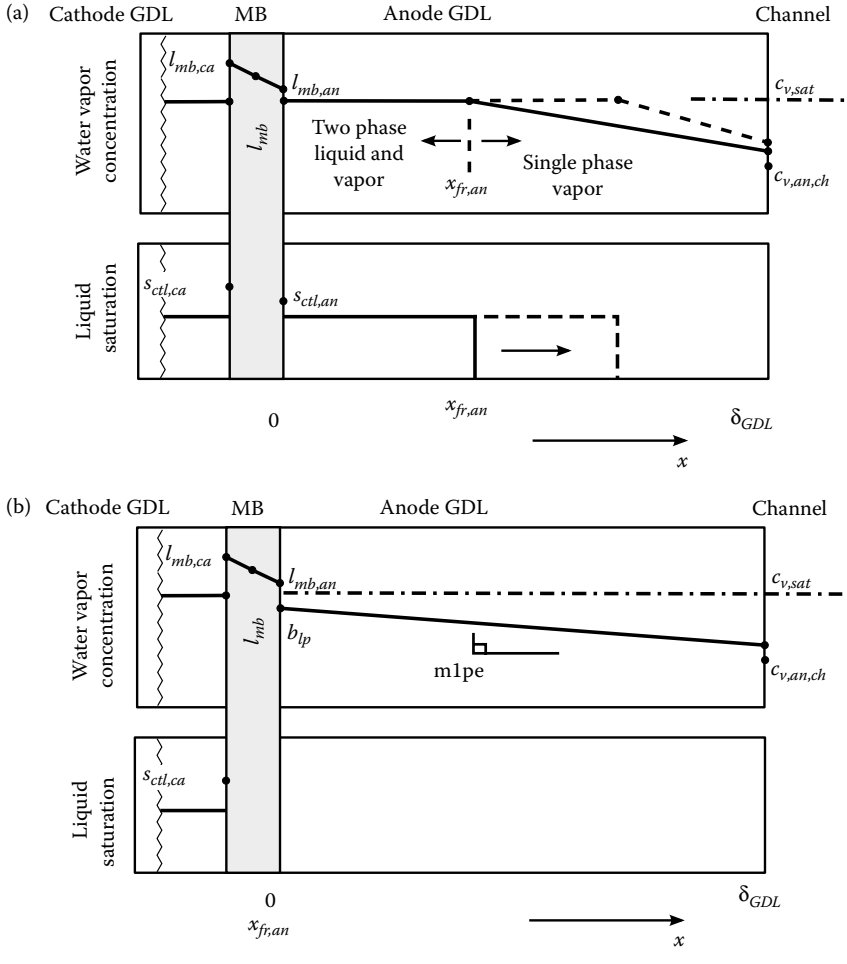

FIGURE 5.8 Two-phase water front evolution in the anode GDL for the unit fuel cell model, which represents a 1-D slice from Figure 5.1. The figures show the assumed profiles in membrane water content (gray region), GDL water vapor concentration (upper plots) and GDL liquid water saturation (lower plots) for single phase (b) and two-phase (a) conditions in the anode.

where $K_L = M_v/(\rho_l\, s_* \in \delta_{GDL})$ is a constant which accounts for the geometry and density of liquid water in the two-phase region assuming the front propagates with constant liquid saturation ($s = s_*$), as shown in Figure 5.8. The right-hand side of (Equation 5.28) depends on $N_{l,an}$, which is equal to the difference between the flux of water entering the GDL from the membrane, $N_{v,an,mb}$, and the flux of vapor leaving from the GDL and entering the channel $N_{v,an}$,

$$N_{l,an} = N_{v,an,mb} - N_{v,an}. \tag{5.29}$$

These fluxes are diffusive fluxes and hence depend on the water vapor concentration profile in the GDL, $c_{v,an}(x)$ which in turn depends on the liquid-phase front location $x_{fr,an}$.

This spatially and temporally dependent coupling between the liquid and vapor phase through the condensation and evaporation is simplified in this chapter by employing two assumptions. First, the liquid water dynamics are much slower than the gas dynamics, due to the difference in density 1000 times greater. Therefore, for the purpose of tracking the liquid front propagation in the fuel cell, we can take the gas states to be in steady state [8,53,58]. Second, we assume that all the condensation and evaporation occurs at the membrane–GDL interface (MB–GDL) $x = 0$ [58] and liquid-phase front location $x_{fr,an}$. With these assumptions, the steady-state solution of the vapor diffusion equation is a piecewise linear profile for the water vapor concentration, which follows the liquid water front location as seen in Figure 5.8,

$$c_{v,an}(x) = \begin{cases} \min(b_v, c_{v,sat}(T)) & x \le x_{fr,an} \\ m_v\, x + b_v & x > x_{fr,an}\,. \end{cases} \tag{5.30}$$

The vapor profile, $c_{v,an}(x)$, behind the front in the two-phase region $x \le x_{fr,an}$ is equal to the concentration at saturation $c_{v,sat}(T)$. In the vapor-phase region, between the front location and the channel $x > x_{fr,an}$, the linear profile is the result of Fickian diffusion.

The propagation of the front location in the cathode GDL, $\dot{x}_{fr,ca}$, is defined similarly, where $N_{l,ca}$ is given by Equation 5.43. For brevity, only the equations detailing the anode side will be presented, the cathode side equations are similar except where noted. The system can be described using an isothermal model or a slowly evolving temperature profile where the measured end plate temperature is imposed as the boundary condition at the GDL-channel (GDL-ch) interface and a few degrees higher (load-dependent) temperature at the membrane–GDL (MB–GDL) interface because heat is generated from the reaction. Here, we assume a spatially invariant temperature distribution and present the equations with explicit temperature dependencies only when variables are first introduced.

5.7.1 Membrane Water Transport

The water flux out of the membrane and into the anode GDL is governed by diffusion and osmotic drag

$$N_{v,an,mb} = 2\,\frac{D_w(\lambda_{mb}, T)\cdot\left(\lambda_{mb} - \lambda_{an}\right)}{\delta_{mb}} - \frac{n_d\, i_{fc}}{F}\,. \tag{5.31}$$

The first term in Equation 5.31 describes diffusion in the membrane, which is driven by the anode side gradient of the membrane water concentration as it is shown in Figure 5.8. The anode water gradient is defined by the state, λ_{mb}, and the water content at the membrane interface with the anode GDL, λ_{an}. Since the catalyst layer is very thin its effect can be lumped into λ_{an}. Therefore, we express λ_{an} as a function of catalyst flooding level, $s_{ctl,an}$, and the vapor concentration in the GDL, $c_{v,an}(0)$,

$$\lambda_{an} = (1 - s_{ctl,an})\,\lambda_{T,a} + s_{ctl,an}\,\lambda_{max}, \tag{5.32}$$

where $\lambda_{max} = 22$ [59] is the water content of a liquid equilibrated membrane and $\lambda_{T,a}$ is the membrane water uptake isotherm [41,55],

$$\lambda_{T,a} = c_0(T) + c_1(T)\,a + c_2(T)\,a^2 + c_3(T)\,a^3, \tag{5.33}$$

which is a function of the water activity, a, at the GDL–MB interface. The water activity in the GDL–MB interface is equal to the ratio of vapor concentration to the saturation value, $a_{an,mb} = c_{v,an}(0)/c_{v,sat}$. The

$c_i(T)$, $i \in \{0, 1, 2, 3\}$, values are calculated by a linear interpolation of uptake isotherms [55] which were measured at 30°C and 80°C,

$$c_i(T) = \frac{(c_{i,353} - c_{i,303})}{50}(T - 303) + c_{i,303}, \tag{5.34}$$

using the values in Table 5.1.

The dependence of membrane water content on catalyst liquid saturation is introduced to capture the observed anode flooding behavior. We propose that $s_{ctl,an}$ be a function of the liquid flux $N_{l,an}$ as follows:

$$s_{ctl,an} = \frac{\max(N_{l,an}, 0)}{N_{L,max}}, \tag{5.35}$$

where $N_{L,max}$ is the maximum liquid water flux the catalyst layer can handle before becoming completely saturated. $N_{L,max}$ should be inversely proportional to liquid water viscosity, therefore we choose the following functional form with an exponential temperature dependence:

$$N_{L,max}(T) = N_{L0}\left(\exp\left[N_{L1}\left(\frac{1}{303} - \frac{1}{T}\right)\right]\right), \tag{5.36}$$

where N_{L1} and N_{L0} are tunable parameters.

$$D_w(\lambda, T) = \begin{cases} a_w\, c_f\, \exp[2416(1/303 - 1/T)]\, \dfrac{M_v\, \lambda_{mb}\, \rho_{mb}}{\left(\rho_l\, EW + M_v\, \lambda_{mb}\, \rho_{mb}\right)}\, \dfrac{\mathrm{d}\log a_{mb}}{\mathrm{d}\log \lambda_{mb}}, & \lambda < \lambda_{a=1}(T) \\ D_{w0}(T) + D_{w1}(T)\cdot\lambda, & \lambda \geq \lambda_{a=1}(T) \end{cases} \tag{5.37}$$

$$D_{w0}(T) = \frac{a_w\, c_f\, \exp[2416(1/303 - 1/T)]}{\left(\rho_l\, EW + M_v\, \lambda_{a=1}(T)\, \rho_{mb}\right)}\, \frac{M_v\, \lambda_{a=1}(T)^2\, \rho_{mb}}{c_1(T) + 2\, c_2(T) + 3\, c_3(T)} - \lambda_{a=1}(T)\, D_{w1}(T) \tag{5.38}$$

$$D_{w1}(T) = \left(D_{w,303} + \frac{D_{w,353} - D_{w,303}}{50}(T - 303)\right)\, a_w\, c_f\, \exp[2416(1/303 - 1/T)] \tag{5.39}$$

Note 1: The liquid volume fraction in the catalyst, $s_{ctl,an}$, represents the fraction of the membrane which is in contact with liquid water. Therefore, $s_{ctl,an}$ in Equation 5.32 linearly interpolates between the water content of a vapor equilibrated membrane, $\lambda_{T,a=1}$, and the liquid equilibrated value $\lambda_{max} = 22$.

Note 2: There is no vapor gradient across the membrane when liquid is present in both GDLs, since the vapor concentration on both sides of the membrane is equal to $c_{v,sat}$, therefore, the difference between the catalyst flooding levels $s_{ctl,an}$ and $s_{ctl,ca}$ drives water transport through the membrane.

Note 3: The membrane water diffusion coefficient, $D_w(\lambda_{mb}, T)$ in Equation 5.37, has an exponential dependence on temperature, but only a linear dependence on membrane water content for values of λ_{mb} greater than 6 [55,62]. Typical membrane water content, when the cell is near flooding conditions, is in the range 9–14.

TABLE 5.1 Fuel Cell Parameters

$\{c_{0,303}, c_{1,303}, c_{2,303}, c_{3,303}\}$	$\{0.043, 17.81, -39.85, 36\}$ [60]
$\{c_{0,353}, c_{1,353}, c_{2,353}, c_{3,353}\}$	$\{0.3, 10.8, -16, 14.1\}$ [61]
$\{D_{w,303}, D_{w,353}\}$	$\{0.00333, 0.00259\}$ [62]
a_w ($\mathrm{cm}^2\ \mathrm{s}^{-1}$)	2.72E-5 [55]
c_f ($\mathrm{mol\ cm}^{-3}$)	0.0012 [55]
Sherwood number	$Sh = 2.693$ [52]

The second term in Equation 5.31 describes electro-osmotic drag, which pulls water from the anode to the cathode with the conduction of protons through the membrane, and therefore is dependent on the current density, i_{fc}. The drag coefficient is given by

$$n_d = \begin{cases} \lambda_{mb}/\lambda_{T,a=1} & \lambda_{mb} < \lambda_{T,a=1}, \\ K_{\lambda,T}(\lambda_{mb} - \lambda_{T,a=1}) + 1 & \lambda_{mb} \geq \lambda_{T,a=1}, \end{cases} \tag{5.40}$$

which depends on the membrane water content and temperature with

$$K_{\lambda,T} = \frac{(-1.834 + 0.0126\,T - 1)}{(\lambda_{max} - \lambda_{T,a=1})} \tag{5.41}$$

a linear interpolation between $n_d = 1$ for the vapor equilibrated membrane and $n_d = -1.834 + 0.0126\,T$, for the liquid equilibrated membrane [62]. The volume fraction of water in the membrane, f_v, is calculated using the following equation:

$$f_v = \frac{\lambda_{mb} V_w}{V_{mb} + \lambda_{mb} V_w}, \tag{5.42}$$

where $V_{mb} = EW/\rho_{mb}$ is the dry membrane volume (equivalent weight divided by density) and V_w is the molar volume of water.

5.7.1.1 Cathode Side Equations

The same set of equation are used on the cathode, with the inclusion of generated water and a change of sign which accounts for the use of a different coordinate axis

$$\frac{1}{2}\frac{i_{fc}}{F} - N_{v,ca,mb} = N_{v,ca} + N_{l,ca}, \tag{5.43}$$

$$N_{v,ca,mb} = 2\,\frac{D_w(\lambda_{mb}, T)\cdot\left(\lambda_{ca} - \lambda_{mb}\right)}{\delta_{mb}} - \frac{n_d\, i_{fc}}{F}. \tag{5.44}$$

5.7.1.2 Water Exchange with the Channel

The two-phase front location inside the GDL also determines the rate of vapor and liquid water exchange with channel. When the front location is inside the GDL, $x_{fr,an} < \delta_{GDL}$, then the water exchange with the channel is in the vapor phase only. When the front reaches the channel, $x_{fr,an} = \delta_{GDL}$, then the mass flow of water from the GDL into the channel is the sum of the liquid and vapor flux

$$W_{w,an,GDL} = A_{fc} M_v \left(N_{v,an} + \max(N_{l,an}, 0) \text{ if } x_{fr,an} = \delta_{GDL}\right) \tag{5.45}$$

The max(·) function prevents liquid water in the channel from entering into the GDL, since in the model formulation $N_{l,an} < 0$ represents a receding two-phase front inside the GDL.

5.8 Fitting Water Transport Parameters

The tunable parameters N_{L0} and N_{L1} in Equation 5.36, can be experimentally determined from the neutron imaging data by observation of the rate of liquid water accumulation in the anode channel during operation of the PEMFC under DEA conditions [7]. The quasi-steady-state accumulation rate is calculated from the time evolution of the system and plotted as function of current density, i_{fc}, and cell temperature, T, in Figure 5.9. The fuel cell operating conditions are chosen to maintain as close to uniform channel conditions as possible so that the lumped channel approximation remains valid. Therefore, experimental data with fully humidified cathode inlet gas feeds were used for the parameter

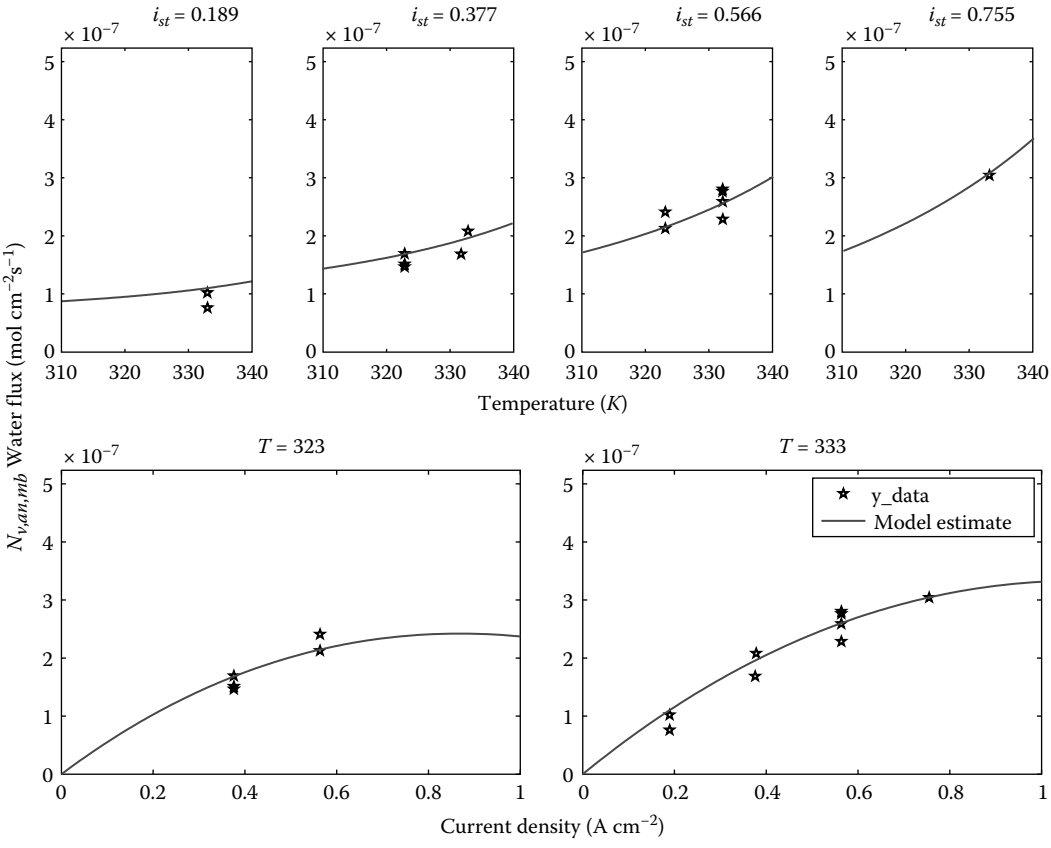

FIGURE 5.9 The fit parameters yield a model with a convex temperature relationship, and that is concave with current density.

identification. Under these operating conditions we can assume that both the anode and cathode GDLs are saturated, $x_{fr,an} = \delta_{GDL}$, when liquid water accumulation in the anode channel is observed; and the rate of liquid water accumulation in the anode channel can be attributed to the rate of water flux through the membrane. Under these assumptions, $c_{v,an,ch} = c_{v,ca,ch} = c_{v,sat}(T)$ and $x_{fr,an} = x_{fr,ca} = \delta_{GDL}$, we can solve for the equilibrium value of membrane water content, $\lambda = \lambda_{eq}$ in Equation 5.46, and the resulting membrane flux into the anode $N_{v,an,mb,eq}$. The function $N_{v,an,mb,eq}(T, i_{fc}, \lambda = \lambda_{eq}, N_{L0}, N_{L1})$ as shown in Equation 5.47, is fit to the measured liquid water accumulation data, using the nonlinear least-squares fitting routine in MATLAB®, for the parameters N_{L0} and N_{L1}. An analytic expression for Jacobian matrix is easily calculated, which speeds up the convergence of the parameter fitting. The tuned parameters are shown in Table 5.2. The specific functional form of $N_{L,max}$, given in Equation 5.36, is independent of the tuning procedure, and therefore the model could easily be parameterized using another functional relationship.

TABLE 5.2 Tuned Parameters for the Liquid Water Transport Model

s_*	0.37
N_{L0}	2.3434
N_{L1}	3991

The tuned model shows an exponential increase in the rate of water crossover and liquid accumulation in the anode channel with increasing temperature due to the exponential term in the diffusion coefficient Equation 5.37. The water crossover rate increases with current density, as the rate of water production increases, until the osmotic drag term begins to dominate Equation 5.31 at which point the water crossover rate begins to decrease with further increase of current density.

5.9 Fuel Cell Terminal Voltage

Fuel cell terminal voltage is the main measurable output of the system, and represents a static nonlinear output mapping of the states and operating conditions (temperature and load current). The electrochemical dynamics are very fast [12,63], so the mass transport of reactants into the cell, that is, the channel concentrations, and the removal of product water, represents the dynamic behavior observed in the measured voltage.

$$\lambda_{eq} = \frac{4\,F\,\lambda_{a=1}(T)\,N_{L,max}(T) - \lambda_{a=1}\,i_{fc} + \lambda_{\max} i_{fc}}{4\,F\,N_{L,max}(T)} \tag{5.46}$$

$$N_{v,an,mb,eq} = \frac{2\,FN_{L,max}(T)\,D_{ve}D_w(\lambda_{eq},T)\,(\lambda_{eq} - \lambda_{a=1}(T)) - i_{fc}\,\delta_{mb}\,n_d(\lambda_{eq},T)}{F\left(N_{L,max}(T)\,D_{ve}\,\delta_{mb} + 2\,D_{ve}\,D_w(\lambda_{eq},T)(\lambda_{max} - \lambda_{a=1}(T))\right)} \tag{5.47}$$

The inputs to the voltage model are: total current $I_{fc}(A)$, temperature T(K), membrane water content λ_{mb}, hydrogen partial pressure at the membrane surface $P_{H_2,an,mb}$ (Pa), and oxygen partial pressure at the cathode membrane surface $P_{O_2,ca,mb}$ (Pa) which is calculated in Equation 5.61. The pressure and concentration are related by the ideal gas law $P_{H_2,an,mb} = RTc_{H_2,an,mb}$, where R is the universal gas constant. The cell terminal voltage is calculated from the open circuit potential minus the concentration, over-potential and ohmic losses.

$$V_{cell} = E_{rev} - \eta_{act,ca} - \eta_{act,an} - \eta_{mb} - \eta_{ohmic} \tag{5.48}$$

The reversible voltage is given by

$$E_{rev} = E_0 - \frac{RT}{nF}\log\left(\frac{aH_2O}{aH_2\,\sqrt{(aO_2)}}\right) \tag{5.49}$$

where $E_0 = 1.229 - (T - T_0) \times 2.304 \times 10^{-4}$ [9]. The reactant and product activities are calculated from the concentrations $aH_2 = c_{H_2,an,mb}/C_{ref,H_2}$, $aO_2 = c_{O_2,ca,mb}/C_{ref,O_2}$ and $aH_2O = 1$ since liquid water product assumed. The subscript *ref* refers to the reference quantity, and subscript *ca*, *mb* refers to the cathode membrane surface.

In order to simplify the calculation of cell voltage, a hyperbolic sine function is used for the calculation of over-potentials, $\eta_{act,ca}$ and $\eta_{act,an}$, from the exchange current density, $i_{o,ca}$ and $i_{o,an}$,

$$\eta_{act,ca} = \frac{RT}{\alpha_{c,a}nF}\text{asinh}\left(\frac{i_{fc} + i_{loss}}{2i_{o,ca}}\right), \tag{5.50}$$

where $i_{fc} = I_{fc}/A_{fc}$ is the current density and i_{loss} is the lost current density due to hydrogen crossover, a tuned parameter which is listed in Table 5.3. $n = 2$ is the electron transport number, and $F = 96485$ C per mol of electrons is Faraday's constant. The hyperbolic sine is equivalent to the Butler–Volmer equation when the forward and reverse reaction coefficients ($\alpha_{c,a} = \alpha_{c,c}$) are equal [45]. The exchange current density is given by the following equation:

$$i_{o,ca} = i_{o,ref,ca}\left(\frac{c_{O_2,ca,mb}}{C_{ref,O_2}}\right)^{\gamma_{O_2}}\left(\frac{c_{H^+_{ca,mb}}}{C_{ref,H^+}}\right)^{\gamma_{H^+}}\exp\left(\frac{-E_c}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right), \tag{5.51}$$

where $i_{o,ref,ca}$ is the reference current density, c_* is the reactant concentration, γ is the concentration parameter, and E_c in the Arrhenius term is the activation energy for hydrogen oxidation on platinum [64].

TABLE 5.3 Tuned Parameters in the Voltage Equation

$i_{o,ref,ca}$	7E-8 (A cm^{-2})	Cathode exchange current
$i_{o,ref,an}$	0.05 (A cm^{-2})	Anode exchange current
i_{loss}	1E-3 (A cm^{-2})	Crossover current
D_{eff}	0.35	Diffusivity in GDL
R_{GDL}	0.275 (Ω cm^2)	Contract resistance

The cathode concentration parameter for the local proton activity, $\gamma_{H^+} = 0.5$ is given by [65]. Although the cathode reaction depends on the oxygen concentration as well as the activity of protons in the membrane [66], the proton activity term is typically neglected since there are sufficiently many protons under fuel cell normal operation, that is, $\left(c_{H^+_{ca,mb}}/C_{ref,H^+}\right)^{\gamma_{H^+}} \approx 1$.

Similarly for the anode side,

$$\eta_{act,an} = \frac{RT}{\alpha_{a,a} nF} \text{asinh}\left(\frac{i_{fc} + i_{loss}}{2i_{o,an}}\right), \tag{5.52}$$

where the anode exchange current density is

$$i_{o,an} = i_{o,ref,an}\left(\frac{c_{H_2,an,mb}}{C_{ref,H_2}}\right)^{\gamma_{H_2}} \exp\left(\frac{-E_c}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right). \tag{5.53}$$

The potential drop due to proton conductivity losses in the membrane is represented by the following equation:

$$\eta_{mb} = i_{fc} R_{mb}(T, \lambda_{mb}), \tag{5.54}$$

where the membrane resistance is calculated as follows:

$$R_{mb}(T, \lambda_{mb}) = \frac{\exp\left[1268\left(\frac{1}{T} - \frac{1}{303}\right)\right]}{-0.00326 + 0.005193\lambda_{mb}}\delta_{mb}, \tag{5.55}$$

δ_{mb} is the membrane thickness, and the membrane conductivity is a function of water content and temperature using the standard relationship from Springer et al. [60].

Finally, the GDL and contact resistances are lumped into R_{ohmic}, for the ohmic loss term,

$$\eta_{ohmic} = i_{fc} R_{ohmic}. \tag{5.56}$$

The typical graph associated with fuel cell operation is the polarization curve, shown in Figure 5.10, which is a plot of the terminal voltage V_{cell} vs the applied current density i_{fc}. The initial drop in cell voltage for small current density, is due to the activation losses $\eta_{act,an}$ and $\eta_{act,ca}$. The middle region of the polarization curve, is referred to as the ohmic region, which has a fairly linear $i - v$ relationship, and the losses for this range of current density are primarily attributed to the ohmic terms η_{mb} and η_{ohmic}.

The steep drop in cell voltage, shown in Figure 5.10 at high current density, is attributed to concentration loss at the membrane surface. This concentration loss comes from the diffusion of reactants, specifically oxygen, from the channel through the GDL to the CL. The concentration value, $c_{O_2,ca,mb}$, represents the amount of oxygen available in the catalyst layer, which decreases at high current density due to the finite resistance to gas transport through the GDL. Since the diffusion is a fast process, the 1-D steady-state

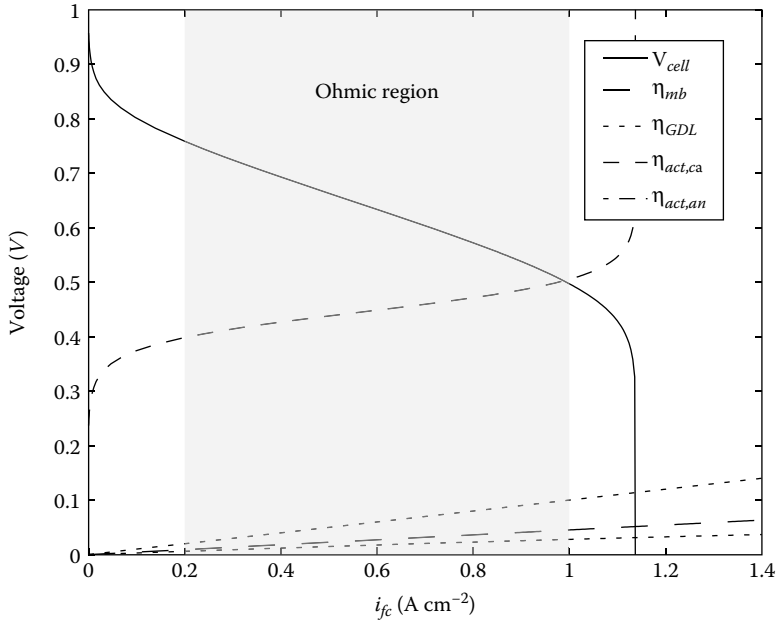

FIGURE 5.10 Fuel cell polarization curve.

profile can be considered assuming Fick's Law in the GDL,

$$N_{O_2,ca,GDL} = -D_{eff}(s,\epsilon)D_{O_2}\frac{\partial c_{O_2,ca,GDL}}{\partial x} \tag{5.57}$$

with a Neumann boundary condition at the membrane surface where the flux is equal to the consumption of oxygen

$$N_{O_2,ca,GDL}\big|_{x=0} = -N_{O_2,rct}, \tag{5.58}$$

and a Neumann boundary condition at the channel which is given by the mass transfer coefficient, k_{mt} [52],

$$N_{O_2,ca,GDL}\big|_{x=\delta_{GDL}} = k_{mt}\left(c_{O_2,ca,GDL}\big|_{x=\delta_{GDL}} - c_{O_2,ca,ch}\right), \tag{5.59}$$

where $c_{O_2,ca,ch}$ is the oxygen concentration in the channel. The consumption rate of oxygen, (mol cm^{-2} s^{-1}) is given by

$$N_{O_2,rct} = \frac{i_{fc}}{4F}. \tag{5.60}$$

The free space diffusion coefficient D_{O_2} m^2 s^{-1} is multiplied by an effective diffusivity coefficient, $D_{eff}(s,\epsilon)$, to account for the porous structure of the GDL, and the reduced volume due to liquid water accumulation in the diffusion medium [48]. In the case of constant diffusivity, $D^{sim}_{O_2} = D_{O_2}D_{eff}(s_{im},\epsilon)$, this reduces to

$$\begin{aligned} c_{O_2,ca,mb} &= c_{O_2,ca,GDL}\big|_{x=0} \\ &= c_{O_2,ca,ch} - \left(\frac{1}{k_{mt}} + \frac{\delta_{GDL}}{D^{sim}_{O_2}}\right)N_{O_2,rct}, \end{aligned} \tag{5.61}$$

The same equation holds true for the anode, where

$$N_{H_2,rct} = \frac{i_{fc}}{2F}. \tag{5.62}$$

For a derivation of the reactant profiles with nonconstant $D_{eff}(s(x))$, see [8].

5.9.1 Apparent Current Density and Reduced Cell Area

The effect of liquid water and nitrogen accumulation on the measured terminal voltage can be captured by an apparent current density, corresponding to the area of the hydrogen-rich portion of the cell. The use of apparent current density, i_{app}, in place of the nominal current density, i_{fc}, in the above voltage model describes the resulting voltage drop from mass accumulation in the channel.

$$i_{app} = \frac{I_{fc}}{A_{app}}, \tag{5.63}$$

where the apparent area is calculated from the nitrogen and water mass in the channel using the following equation:

$$A_{app} = A_{fc}\frac{y_{fr}}{L_{ch}}, \tag{5.64}$$

where L_{ch} is the channel length and y_{fr} is the front location. A simple model, which considers the nominal current density, channel pressure, and temperature, can be used to translate the mass of liquid water and nitrogen in the DEA channel into a front location

$$m_{l,an,ch}, m_{N_2,an,ch} \rightarrow y_{fr}, \tag{5.65}$$

based on the assumptions of saturated vapor conditions and uniform apparent current density in the region above of the front as shown in Figure 5.11.

$$y_{fr} = \begin{cases} L_w & m_{N_2} \leq m_{N_2,critical}, \\ y_{frN_2} & m_{N_2} > m_{N_2,critical}, \end{cases} \tag{5.66}$$

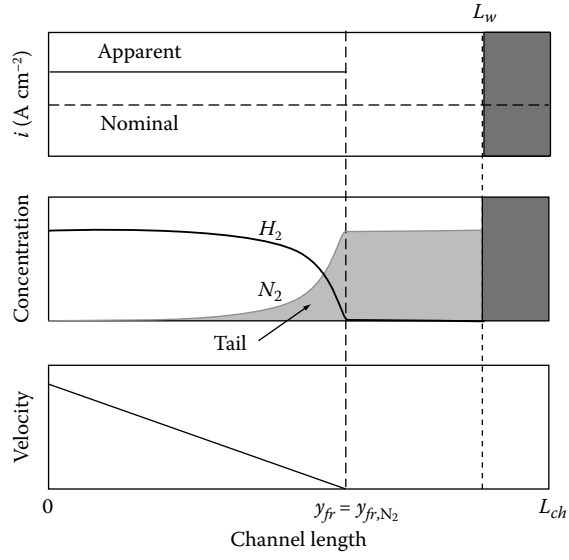

FIGURE 5.11 Nitrogen front propagation schematic (not to scale).

where liquid water front location, L_w, is defined by the channel dimensions, liquid water density ρ_l, and the mass of liquid water in the anode channel (Equation 5.3), using the following equation:

$$L_w = \left(1 - \frac{m_{l,an,ch}}{\rho_l V_{an,ch}}\right) L_{ch}. \tag{5.67}$$

The nitrogen front location y_{frN_2} is found by solving Equation 5.75 and the critical nitrogen mass at which the nitrogen front develops is given by

$$m_{N_2,critical} = M_{N_2} \frac{V_{ch}}{L_{ch}} \frac{(P_{an} - P_{v,sat}(T))}{(RT)} \left(\frac{\sqrt{2L_w D_{N_2,H_2}} \left(1 - \exp\left(-\frac{K_v i_{fc} L_w}{2D_{N_2,H_2}}\right)\right)}{\operatorname{erf}\left(\frac{\sqrt{K_v i_{fc} L_w}}{\sqrt{2D_{N_2,H_2}}}\right) \sqrt{K_v i_{fc} \pi}} \right). \tag{5.68}$$

In order to derive this relationship, we assume steady-state convection and Fickian diffusion in the anode channel

$$0 = \frac{\partial c_{N_2}}{\partial t} = \frac{\partial}{\partial y}\left(D_{N_2,H_2} \frac{\partial c_{N_2}}{\partial y}\right) - V(y) \frac{\partial c_{N_2}}{\partial y}. \tag{5.69}$$

The convective velocity, $V(y)$, (m s^{-1}) can be calculated from the consumption of hydrogen, assuming the flux of N_2 crossover is small relative to the reaction rate of H_2. The convective velocity is zero after the front location because no reaction occurs in this region.

$$V(y) = \begin{cases} K_v \frac{i_{fc}}{y_{fr}} (y_{fr} - y) & y \le y_{fr}, \\ 0 & y > y_{fr}, \end{cases} \tag{5.70}$$

where K_v is given by the following equation:

$$K_v = L_{ch} \frac{(w_{ch} + w_{rib})}{2F} \frac{RT}{P_{an}(w_{ch} h_{ch})}, \tag{5.71}$$

where P_{an} is the anode channel pressure and w_{ch}, h_{ch} and w_{rib} denote the channel width, height, and rib width.

The steady-state solution of (Equation 5.69), is of the form

$$c_{N_2}(y) = \begin{cases} c_1 + c_2 \operatorname{erf}\left(\frac{\sqrt{K_v i_{fc}}(y - y_{frN_2})}{\sqrt{2 y_{frN_2} D_{N_2,H_2}}}\right) & y \le y_{frN_2}, \\ (P_{an} - P_{v,sat}(T))/(RT) & y > y_{frN_2}, \end{cases} \tag{5.72}$$

where we can solve for c_1 and c_2 using the BCs $c_{N_2}(0) = 0$ (the system is fed with dry hydrogen) and $c_{H_2}(y_{frN_2}) = 0$, which implies $c_{N_2}(y_{frN_2}) = (P_{an} - P_{v,sat}(T))/(RT)$

Hence, for $y \le y_{frN_2}$, the distributed nitrogen concentration is

$$c_{N_2}(y) = \frac{(P_{an} - P_{v,sat}(T))}{(RT)} \left(1 + \frac{\operatorname{erf}\left(\frac{\sqrt{K_v i_{fc}}(y - y_{frN_2})}{\sqrt{2 y_{frN_2} D_{N_2,H_2}}}\right)}{\operatorname{erf}\left(\frac{\sqrt{K_v i_{fc} y_{frN_2}}}{\sqrt{2D_{N_2,H_2}}}\right)}\right) \tag{5.73}$$

The nitrogen front location, y_{frN_2} and the mass of nitrogen which does not contribute to the apparent current density, $m_{N_2,\text{tail}}$, can be found from the system state m_{N_2} using the following equation:

$$m_{N_2} = m_{N_2,\text{tail}} + m_{N_2,\text{blanket}}, \tag{5.74}$$

where the mass of nitrogen in the blanketed region $m_{N_2,\text{blanket}}$, and in the tail can be found by integrating Equations 5.72 and 5.73 along the channel to get

$$m_{N_2} = M_{N_2}\frac{V_{ch}}{L_{ch}}\frac{(P_{an} - P_{v,sat}(T))}{(RT)}\left((L_w - y_{frN_2}) + \frac{\sqrt{2y_{frN_2}D_{N_2,H_2}}\left(1 - \exp\left(-\dfrac{K_v i_{fc} y_{frN_2}}{2D_{N_2,H_2}}\right)\right)}{\operatorname{erf}\left(\dfrac{\sqrt{K_v i_{fc} y_{frN_2}}}{\sqrt{2D_{N_2,H_2}}}\right)\sqrt{K_v i_{fc}\pi}}\right), \tag{5.75}$$

which can be solved for the nitrogen blanketing front location numerically.

5.10 Simulation Results

Current density i_{fc}, stack temperature T, cathode stoichiometric ratio and cathode inlet relative humidity (dew point temperature) are measured from the experiment and used as the inputs model. A simulation of the experiment at NIST [7] with a humidified cathode inlet feed is shown in Figure 5.12. The anode channel water vapor concentration remains near the saturation value, decreasing only slightly during an anode purge. The membrane water content λ_{mb}, shown in the second subplot of Figure 5.12 is strongly dependent on current density and temperature, increasing with current density at $t = 3850$ (min) and decreasing with temperature at $t = 3785$ (min). The normalized front location is shown in third subplot of Figure 5.12, and the liquid water mass in the anode channel is shown in the last subplot.

The anode purges shown in Figure 5.13 capture the removal of liquid water from the GDL. This can clearly be seen by the flat sections in the plot of anode channel liquid water mass. Liquid water must re-fill the GDL before accumulation of liquid water in the anode channel begins again following an anode purge. The duration of the flat region in the channel liquid water trace depends on the "strength" of the anode purge, both its flow rate and duration, and the amount of liquid water present in the channel preceding the purge. Since the dry hydrogen flowing through the fuel cell must first remove all of the liquid water from the channel before causing the two-phase liquid water front location to recede in the GDL. Our model predicts roughly a 5% change in the anode front location due to the 1 s purge, and matches the time period before water accumulation in the channel resumes.

5.11 MPC Application

In this section, we discuss the application of the MPC methodology to the water flooding–plugging problem in the anode channel. An example on how to use MPC tools to address the water flooding–plugging was shown in [38] and will be added here for illustration of the steps needed to apply rigorous control techniques in the water management problem, relating to the scheduling of anode purges for the DEA operation of PEMFCs. The steps which we followed are outlined in Figure 5.14; starting with the high-fidelity PDE description of the system [41], prior to the inclusion of nitrogen effects [35], we identified that the liquid water behavior in the anode GDL and channel could be described simply by a two state, two-mode hybrid automata, where the states correspond to the propagation of the liquid water front in the GDL, followed by the accumulation of liquid in the channel. The control objectives are to avoid; (1) an unreasonable drop in fuel cell voltage, (2) too much water accumulation in the anode channel, which could lead to fuel starvation and carbon corrosion, (3) high hydrogen utilization.

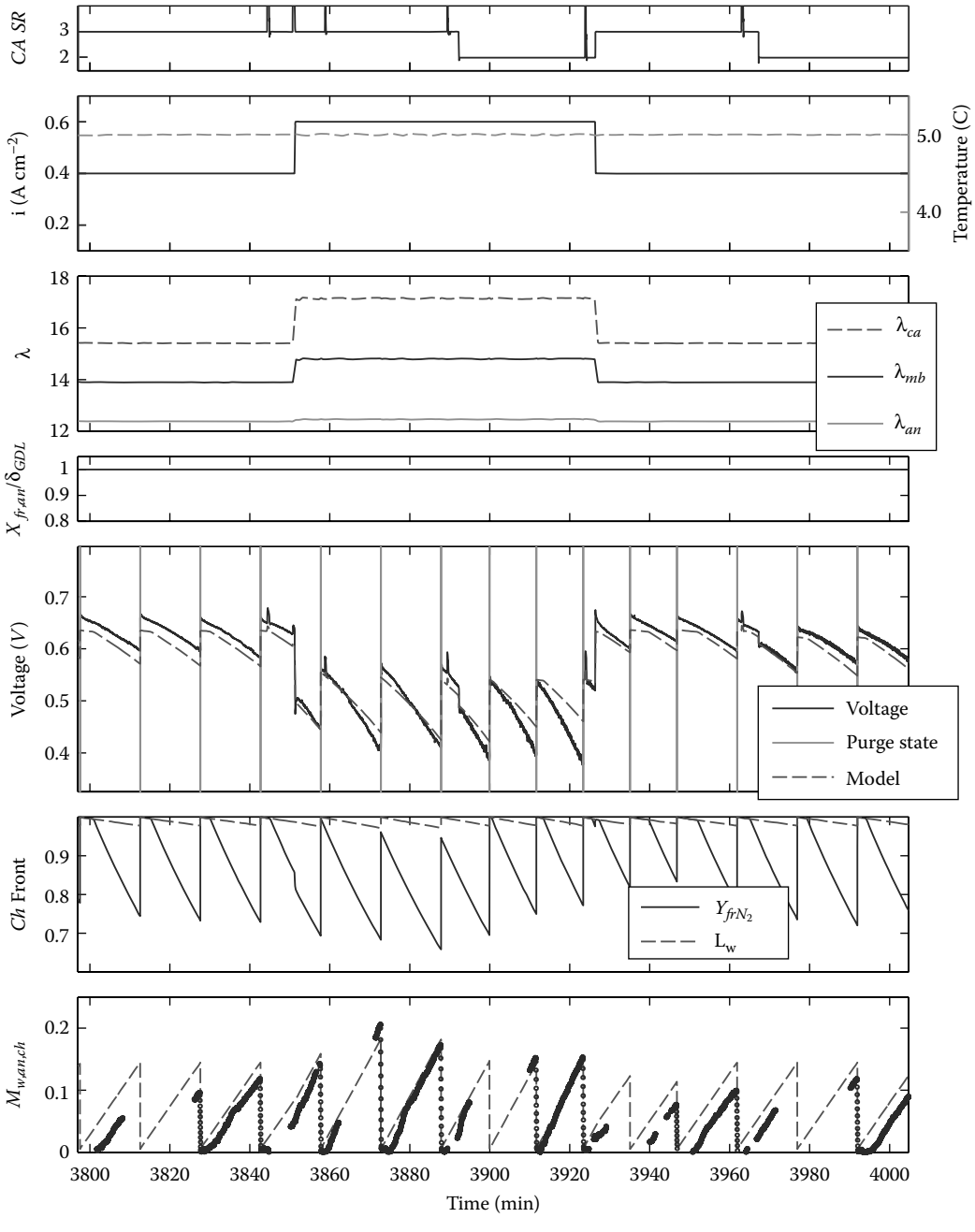

FIGURE 5.12 Simulation of August 7, 2007 Experiment at NIST. Cell temperature increased from 314 K to 323 K at $t = 3785$ (min). The model shows good agreement over the range of temperature and current density, but slightly under-predicts water transport to the anode at higher current density. (Adapted from Siegel, J. B., et al., *J. Electrochem. Soc.*, 155, pp. B1168–B1178, 2008.)

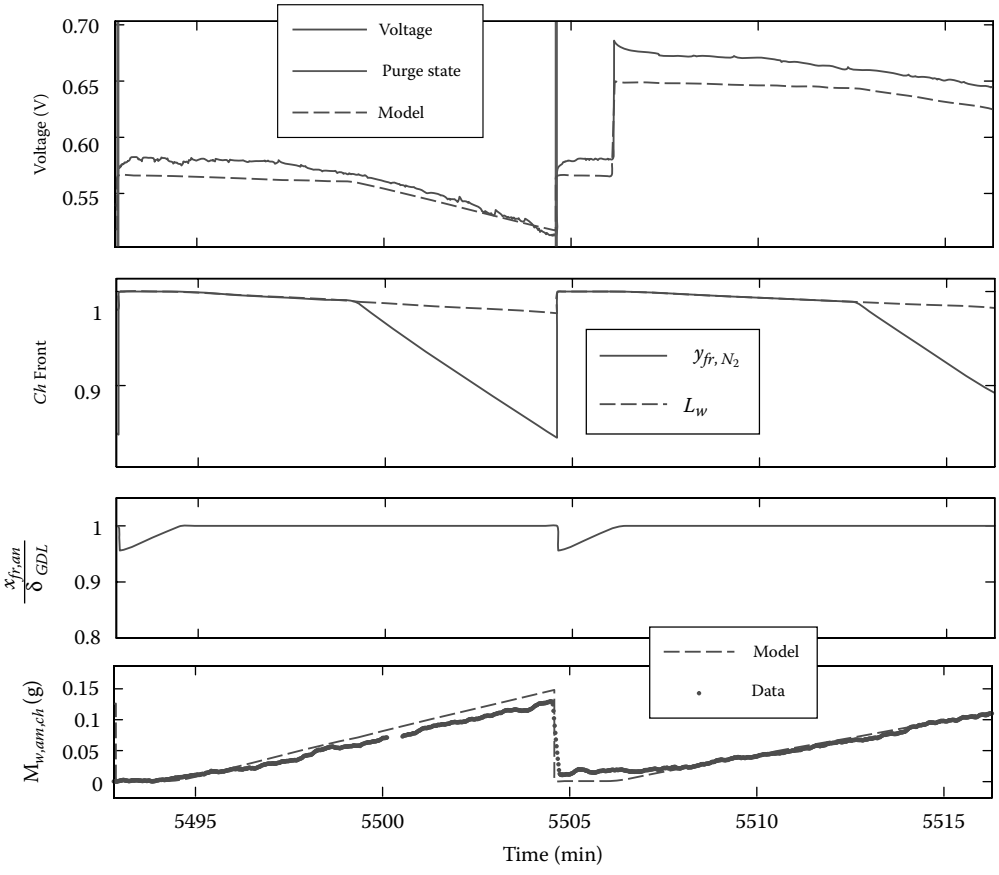

FIGURE 5.13 Zoomed plot showing model predictions vs measured data. After an anode purge, which removes liquid water from the GDL, liquid water accumulation in the anode channel resumes after only after the liquid front reaches back to the channel. Experimental conditions are fully humidified cathode inlet at a cell temperature of 60°C. There is a change in current density, from 0.6 → 0.4 (A cm^{-2}) and cathode stoichimetric ratio, from 3 to 4 at $t = 5506$ min.

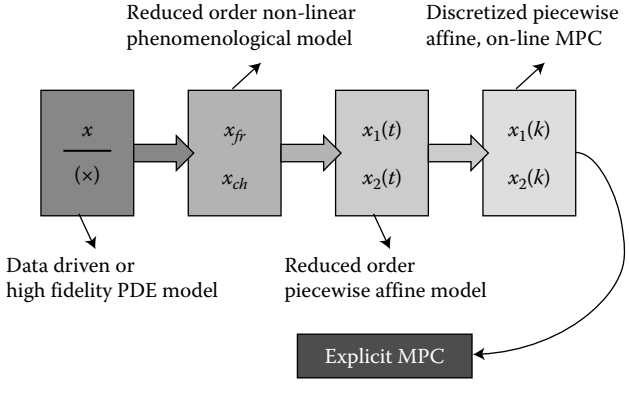

FIGURE 5.14 Modeling approach: Toward MPC application.

Hybrid dynamical models have been used in recent years to analyze and optimize a large variety of systems in which physical processes exhibit phase transitions in the form of switches. Several modeling formalisms have been developed to represent hybrid systems [67,68], including mixed logical dynamical (MLD) systems [69]. MLD is a discrete-time hybrid modeling framework that can be used to formulate optimization problems involving hybrid dynamics. The language HYSDEL (HYbrid Systems DEscription Language) was developed in [70] to obtain MLD models from a high-level textual description of the hybrid dynamics. MLD models can be converted into an equivalent piecewise affine (PWA) models [69] through automated procedures [71,72]. HYSDEL, MLD, and PWA models are used in the Hybrid Toolbox for Matlab [73] for modeling, simulating, and verifying hybrid dynamical systems and for designing hybrid model predictive controllers.

In the case when nitrogen accumulation in the anode channel is not limiting the system behavior, the water accumulation in a PEMFC can be managed with the following process. The water accumulation can be described as a hybrid automaton with a corresponding discrete-time PWA model. The procedure to obtain the hybrid model from the nonlinear model involves linearization and time discretization of the important continuous dynamics. Simulated data from the nonlinear equations and real data from neutron imaging [7] have been analyzed to derive the parameters of the PWA model. Once the system is modeled in discrete-time PWA form, it can be implemented as a HYSDEL model. A control-oriented MLD model is generated and used to synthesize the MPC algorithm for the purge management. Finally, simulation results for the closed-loop system using the high-fidelity nonlinear model are presented with guidelines for the selection of weights in the optimization problem. An equivalent explicit form of the control law is also derived, that allows the implementation in a controller hardware with reduced computational resources.

5.11.1 Hybrid Model for Control

The objective is to design a control law that generates the command for the purge valve u_1 based upon the state measurements or estimates of liquid water accumulation in the anode GDL and anode channel. This water accumulation is directly related to the fuel cell performance, the cell voltage through the accumulation of liquid water in the channel shown via simulation in Figure 5.12, and also observed in neutron imaging data [7].

The water propagation and accumulation within the layers of the fuel cell can be characterized by a moving front paradigm, shown in Equation 5.28 and also in [53], with two linear approximations for a given set of operating conditions. Hence, a natural formulation for the system is a PWA model.

The normalized liquid water front location, $x_{fr,an}/\delta_{GDL} \in [0, 1]$, in the GDL, shown in Figure 5.13, can be derived from the above model (Equation 5.28). The GDL front location will be used to derive the first state of the hybrid model. The second state of the hybrid model approximates the accumulation of water in the channel, x_{ch}, as it relates to the model parameter

$$x_{ch}(t) = m_{l,an,ch}(t) \times 10^4, \tag{5.76}$$

where the mass of liquid water in the channel, $m_{l,an,ch}$ in Equation 5.3, is scaled by 10^4 in order to improve the numerical stability of the model.

Until the liquid reaches the channel, see Figure 5.15b, the channel accumulation is attributed only to the condensation of the water vapor diffusing from the membrane to the channel through the GDL. Once the water front reaches the channel, the accumulation proceeds with a faster rate due to liquid water flux from the GDL to the channel as shown in Figure 5.15b. When the purge valve initially opens, liquid water is first removed from the channel, as shown in Figure 5.15c. Once the liquid water is removed from the channel, the liquid water front begins to recede back into the GDL, see Figure 5.15d, moving toward the membrane.

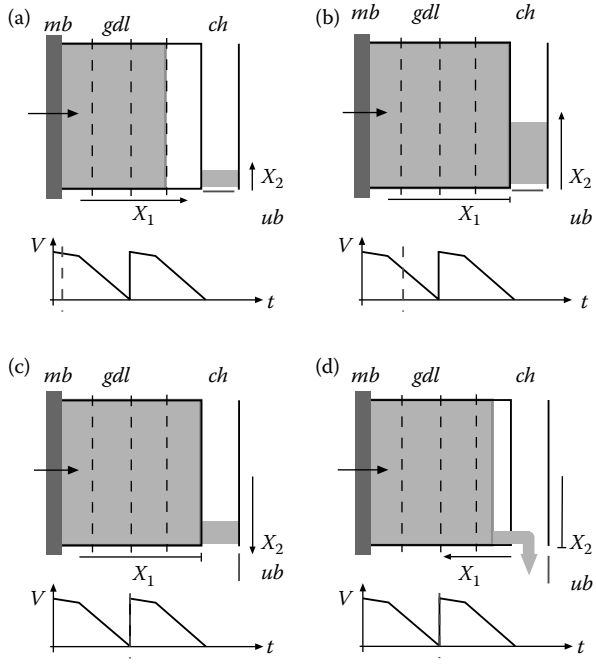

FIGURE 5.15 Evolution of the water front propagation in the GDL and accumulation in the channel.

5.11.2 Hybrid Automaton

This previously described behavior can be modeled as a linear hybrid automaton [74] with two discrete states, shown in Figure 5.16. The first discrete state value (P) is associated with the front propagation and recession through the GDL as shown in Figure 5.15a. When the water front reaches the channel, the discrete state switches to the dynamics of water accumulation (A) in the channel as defined by the max(·) function in Equation 5.45. Let $\vartheta \in \{P, A\}$ be the discrete state. The system dynamics are

$$\text{if} \quad \vartheta(t) = P \qquad \begin{cases} \dot{x_1}(t) = Q_{m2g} - Q_{g2a} \cdot u_1(t) \\ \dot{x_2}(t) = Q_{m2c} \end{cases} \tag{5.77}$$

$$\text{if} \quad \vartheta(t) = A \qquad \begin{cases} \dot{x_1}(t) = 0 \\ \dot{x_2}(t) = Q_{g2c} - Q_{c2a} \cdot u_1(t) \end{cases} \tag{5.78}$$

where $x_1(t)$ is the piecewise approximation of the water front position in the GDL, $x_{fr,an}(t)$, shown in Equation 5.28. The state $x_2(t)$ approximates the mass of water in the anode channel, $x_{ch}(t)$, as shown in Equation 5.76. Finally, the control input $u_1(t) \in \{0, 1\}$ is a boolean variable representing the valve position, where $u_1 = 1$ means that valve is open (purging). The normalized water flows from α to β denoted as $Q_{\alpha 2\beta}$* are used to describe the evolution of the two hybrid states. Specifically, the water flow from the membrane to the GDL, Q_{m2g}, is the rate of water accumulation in the GDL which causes the liquid water front propagation inside the GDL, related to $N_{L,an}$ in Equation 5.29. The water flow, Q_{m2c}, due to the condensation of the water vapor transported from the membrane through the GDL to the channel and water flow from the GDL to channel, Q_{g2c}, both lead to water accumulation in the channel, and are

* Note the change in notation for flow rate, Q, due to nonstandard units, instead of the W used previously.

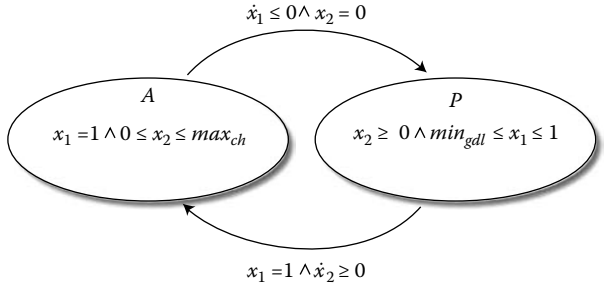

FIGURE 5.16 Hybrid Automaton for DEA operation of PEMFC.

related to $N_{v,an}$ and $N_{L,an}$ in Equation 5.45. Note that the water accumulation in the channel while the GDL front is moving, Q_{m2c}, is smaller than the water flow from the GDL to the channel when the GDL front reaches the channel, Q_{g2c}. Q_{g2c} can be determined directly from the nonlinear model using the equilibrium membrane water flux that was calculated when the front reaches the channel Equation 5.47. Q_{c2a} is the rate of decrease of water in the channel during the purging phase, while Q_{g2a} is the receding rate of backward movement of the front in the GDL. The water removal rate from the GDL is a strong function of the pore-size distribution and PTFE coating weight, hence Q_{g2c} and Q_{g2a} depend on the material, as well as on the total gas flow leaving the anode.

It is then possible to give the transition condition:

- $\mathrm{P} \rightarrow \mathrm{A} : [x_1 = 1,\ \dot{x_2} \ge 0]$,
- $\mathrm{A} \rightarrow \mathrm{P} : [x_2 = 0,\ \dot{x_1} \le 0]$,

where the notation $\mathfrak{X} \rightarrow \mathfrak{Y} : [f]$ indicates the state transition from discrete state $\mathfrak{X}$ to discrete state $\mathfrak{Y}$ occurs when the clause of f is true. The invariant sets associated to each discrete states A and P are

$$\begin{aligned} inv(A) &= \{(x_1, x_2) \in \mathbb{R}^2 : x_1 = 1,\ 0 \le x_2 \le max_{ch}\}, \\ inv(P) &= \{(x_1, x_2) \in \mathbb{R}^2 : min_{gdl} \le x_1 \le 1,\ x_2 \ge 0\}, \end{aligned} \tag{5.79}$$

where max_{ch} and min_{gdl} are respectively the maximum mass of liquid water that completely fills the channel volume and the minimum position of the front in the GDL.

Note here that the state x_1, that is, the level of GDL flooding does not affect the cell voltage because hydrogen can easily diffuse even through a partially flooded GDL. Due to the hydrophobic porous material typically used for GDLs in low-temperature fuel cells the water will not block all of the pores, hence hydrogen passage to the membrane will not be inhibited until the channel fills with water and blocks hydrogen from entering portions of the GDL.

5.11.3 Discrete Time Piecewise Affine System

The hybrid automaton Equation 5.78 can be converted into a PWA model [75] and then formulated in discrete time, $x[k] = x(k\ T_s)$, with sampling $T_s = 0.3$ s. The reformulated PWA system is

$$\text{if} \quad x_1 + x_2 < 1 + \delta_a \qquad \text{(discrete mode P)}$$
$$\begin{cases} x_1[k+1] = x_1[k] + Q_{m2g} T_s - Q_{g2a} T_s \cdot u_1[k] \\ x_2[k+1] = x_2[k] + Q_{m2c} T_s, \end{cases} \tag{5.80a}$$

$$\text{if} \quad x_1 + x_2 \ge 1 + \delta_a \qquad \text{(discrete mode A)}$$
$$\begin{cases} x_1[k+1] = x_1[k] \\ x_2[k+1] = x_2[k] + Q_{g2c} T_s - Q_{c2a} T_s \cdot u_1[k], \end{cases} \tag{5.80b}$$

where the switching condition has been changed to improve the numerical stability of the system. To take into account the effect of condensation defined by Q_{m2c} in Equation 5.78, the transition boundary is increased with a constant $\delta_a \in [0.02 - 0.07]$.

Note that with the formulation in Equation 5.80 the invariant sets cover $\mathbb{R}^2$, that is $\mathcal{S}_{inv}(P) \bigcup \mathcal{S}_{inv}(A) \equiv \mathbb{R}^2$, thus preventing the possibility of the system entering a region where the dynamics are not defined. This problem can arise in practice from numerical integration accuracy or sensor noise.

5.11.4 Performance Output

Finally, we consider the measured output of our system, the fuel cell voltage. Once anode channel flooding occurs, the resulting voltage degradation is associated with the accumulation of liquid water mass in the anode channel, $m_{w,ch}$. Hence, the model output is approximated by a linear dependence on x_2:

$$y(t) = v_0 - x_2(t) \cdot v_m, \tag{5.81}$$

where v_0 is the output voltage of the fuel cell in nonflooding conditions and v_m is a linear gain. This is a reasonable assumption when the fuel cell is operating in the linear or Ohmic region, as shown in Figure 5.10. Both v_0 and v_m depend on the operating conditions, the load generated I_{fc} in amperes (A), the fuel cell temperature T in kelvin (K), the cathode air supply ratio $\lambda_{O_2,ca}$ (unit less), assuming fully humidified cathode inlet, nonhumidified anode inlet, and pressure regulated anode conditions as in [41,47]. We denote the set of operating conditions, by $\Omega = [\, I_{fc}, \lambda_{O_2,ca}, T\,]$, that affect the flow rates and the performance output parameters v_0 and v_m.

5.11.5 Linearization of Nonlinear Model and Parameter Identification

In order to model the dynamics of the water front in a fuel cell on the anode side as a PWA, the parameter set $\mathbf{Q}(\Omega) = \{Q_{m2g}, Q_{g2a}, Q_{g2c}, Q_{c2a}, Q_{m2c}\}$, which represents the rate of increase and decrease of the water in the different modes, and the voltage parameters $\{v_0, v_m\}$ are identified. These parameters have been obtained by observing simulation data at constant conditions, around several operating points defined by different values for Ω from the nonlinear model, which has been tuned and validated on the experimental data obtained through neutron imaging [47]. Note that the PWA affine model does not continuously depend on the load current. Instead different values of Ω correspond to different parameters and thus different realizations of the PWA model. Tables 5.4 and 5.5 show the parameters as function of the stack current for $T = 333$ (K), $\lambda_{O_2,ca} = 2$. The water flow $Q_{g2a} = 0.028$ is independent of stack current. Note that the relation between I_{fc} and the rate of water accumulation is approximately linear.

The parameters Q_{m2g} and Q_{g2a} have been obtained by the analysis of $x_{fr,an}$, using simulations of the nonlinear model, under different conditions for the 53 cm^2 single cell. Q_{m2g} represents the rate of water front $x_1(t)$ measured in the steeper and initial part after the purge. Q_{g2a} is the rate of water decreasing and it is measured as the difference between $x_1(t) = 1$, front at the channel, and the water front position after the purge over the purge duration. Due to the normalization, Q_{m2g} and Q_{g2a} are dimensionless. To determine Q_{m2c}, Q_{g2c}, and Q_{c2a} the mass of liquid water in the channel $m_{w,ch}(t)$ from Equation 5.76 has been analyzed, see Figure 5.17. The parameters will be in [kg $\times 10^{-4}$ s^{-1}] due to the scaling factor

TABLE 5.4 Identified Water Transport Parameters

I_{fc} (A)	Q_{m2g}/I_{fc} (A^{-1})	Q_{g2c} (10^{-4} kg s^{-1})	Q_{c2a} (10^{-4} kg s^{-1})	Q_{m2c} (10^{-4} kg s^{-1})
10	1.6×10^{-5}	4.1×10^{-3}	4.62×10^{-4}	3.3×10^{-4}
20	1.65×10^{-5}	9.1×10^{-3}	18.3×10^{-4}	5.97×10^{-4}
30	1.7×10^{-5}	14.1×10^{-3}	33.7×10^{-4}	10×10^{-4}

TABLE 5.5 Identified Voltage Parameters

I_{fc} (A)	i_{fc} (A cm^{-2})	v_0 (mV)	$v_m\left(\frac{\text{mV}}{\text{kg}}\right)$
10	0.189	755	$3.5 \cdot 10^{-4}$
20	0.337	618	$5.4 \cdot 10^{-4}$
30	0.566	505	$7.7 \cdot 10^{-4}$

introduced in Equation 5.76. The parameters are determined as

$$Q_{m2c} = \frac{x_2(t_{A_s}) - x_2(t_{\delta_f})}{t_{A_s} - t_{\delta_f}}, \quad Q_{g2c} = \frac{x_2(t_{\delta_s}) - x_2(t_{A_s})}{t_{\delta_s} - t_{A_s}}, \quad Q_{g2a} = \frac{x_2(t_{\delta_f}) - x_2(t_{\delta_s})}{t_{\delta_f} - t_{\delta_s}}, \tag{5.82}$$

where t_{A_s} is the time instant when the accumulation starts and, t_{δ_s}, t_{δ_f} are respectively the initial and final time of the purge.

The parameter v_m in Equation 5.81 has been determined as

$$v_m = \frac{v(t_2) - v(t_1)}{m_{w,ch}(t_2) - m_{w,ch}(t_1)} \quad \forall (t_1, t_2) \tag{5.83}$$

while v_0 can be directly calculated by the physics-based voltage equation 5.48. The value of v_0 is the maximum voltage the fuel cell can supply at a given Ω.

Since we aim to control the purge input $u_1(t)$ as a function of x_1, x_2, y so that the reference commands x_{ref}, y_{ref} are tracked, it is necessary to define the references consistently with the actual operating conditions Ω. The value of x_{ref} can be set independently from the actual current or temperature as shown in the following sections, however y_{ref} depends on Ω continuously, since $y(t)$ is driven by v_0. If the objective is to maintain $y(t)$ at the maximum possible value, the reference at the initial time t_0 is $y_{ref} = v_0(t_0)$. In the case of a change in current will occur at time t_1, $v_0(t_1) \neq v_0(t_0)$, and the controller will try to minimize $v_0(t_1) - v_0(t_0)$ which can never converge to zero. To avoid this problem the nonlinear static function $v_0 = f(\Omega)$ has been embedded into a lookup table. Through this function the reference on $y(t)$ is calculated at each sample time. In the following this reference function is called *reference selector*.

5.11.6 MLD Model Validation

The hybrid dynamical model defined by the discrete PWA Equations 5.80 and 5.81 is implemented in HYSDEL. The MLD model obtained using the Hybrid Toolbox for Matlab [73] is equivalent to the PWA model [69]. The resulting MLD model, is described by linear dynamic equations subject to linear mixed integer equations, that is, inequalities containing both continuous and binary variables.

In order to account for modeling errors, constraints on the states of the system, the water front position and the channel water mass accumulation, are included in the modeling framework, to be used for the controller design. These constraints are chosen to be tighter than the limits for safe operation of the physical system,

$$\begin{aligned} min_{gdl} \leq \quad & x_1(t) \quad \leq 1, \\ 0 \leq \quad & x_2(t) \quad \leq max_{ch}. \end{aligned} \tag{5.84}$$

Therefore, the MLD model realization has 2 continuous states (x_1, x_2), 1 binary input (u_1, the purge signal), 1 output (y), 3 auxiliary variables (1 binary, 2 continuous) and 12 mixed-integer linear inequalities.

The model has been validated by running an open-loop simulation under different constant conditions. Figure 5.17 reports the states x_1, x_2 and the output y of the discrete time PWA model compared to the same data obtained simulating the full nonlinear model of the fuel cell at $I_{fc} = 10\,A$, $\lambda_{O_2,ca} = 2$, $T = 333\,K$, with purge period $\Delta = 900\,s$ (when $u_1 = 0$, that is, the system is dead-ended) and a purge duration $\delta_p = 0.3 - s$ (when $u_1 = 1$). Note that during the initial transient there is some model mismatch, though the error is always under 10%, due to the parameters **Q** being identified for steady-state conditions. The quality of

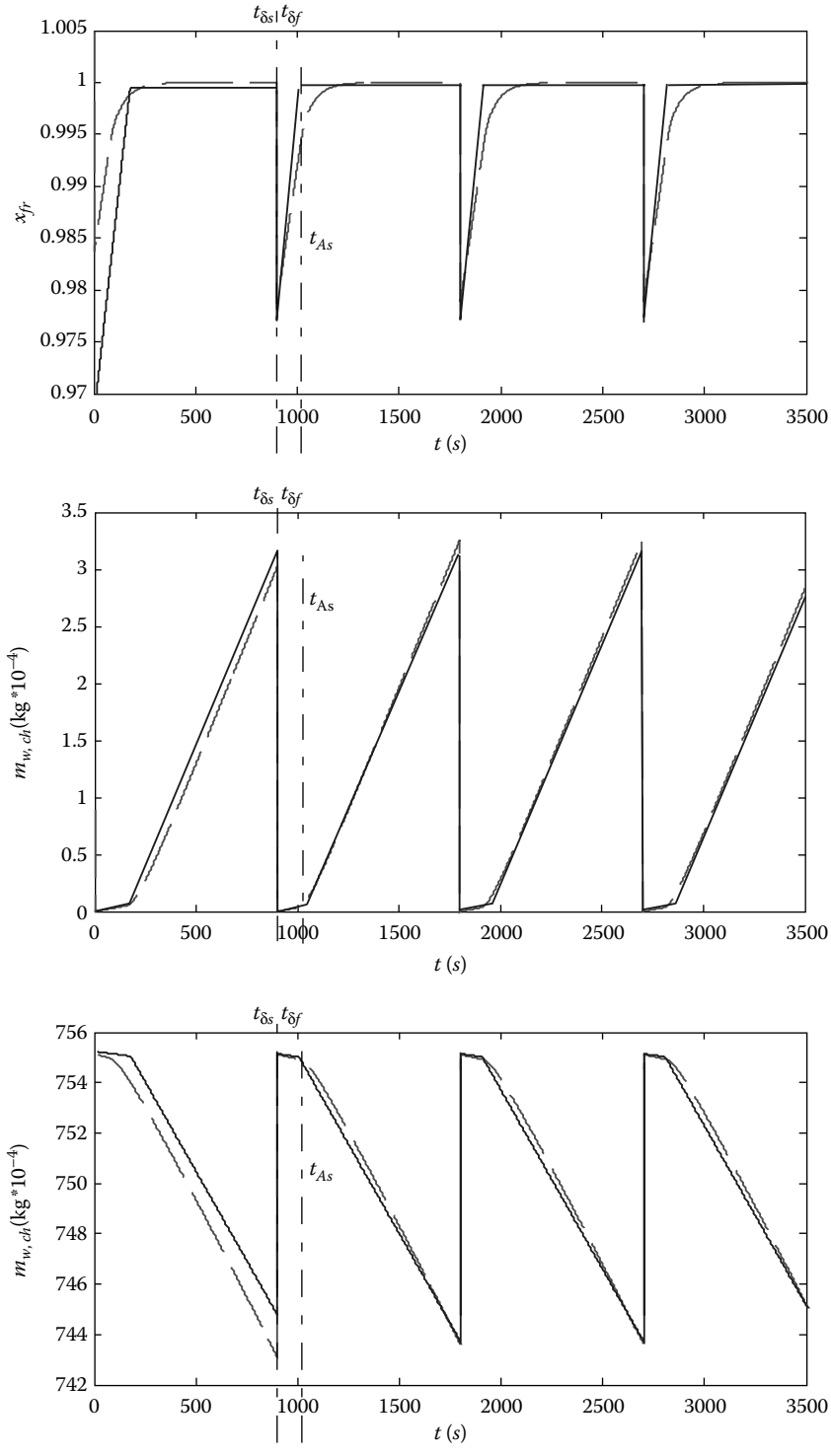

FIGURE 5.17 MPC prediction model validation.

the fit is adequate to predict the behavior of the water front even over a long time horizon as well as over a horizon in the range of seconds, as required for MPC.

5.11.7 MPC Based Online Optimization

MPC has found many industrial applications and it has been successfully applied to hybrid dynamical systems [76–78]. In this section we show how we can derive an MPC controller for the purge management in a fuel cell. In the MPC approach, at each sampling instant a finite horizon open-loop optimization problem is solved using the current state as the initial condition of the problem. The optimization provides an optimal control sequence, only the first element of which is applied to the hybrid system. This process is iteratively repeated at each subsequent time instant, thereby providing a feedback mechanism for disturbance rejection and reference tracking. The optimal control problem is defined as

$$\min_{\xi} \left(J(\xi, x(t)) \triangleq Z_\rho \rho^2 + \sum_{k=1}^{H} (x_k - x_{ref})^T S (x_k - x_{ref}) + \sum_{k=0}^{H-1} (u_k - u_{ref})^T R (u_k - u_{ref}) + (y_k - y_{ref})^T Z (y_k - y_{ref}) \right), \tag{5.85a}$$

$$\text{subj. to} \quad \begin{cases} x_0 = [\, x_{fr}(t),\, x_{ch}(t)\,]^T, \\ x_{k+1} = A x_k + B_1 u_k + B_3 z_k, \\ y_k = C x_k + D_1 u_k + D_3 z_k, \\ E_3 z_k \le E_1 u_k + E_4 x_k + E_5, \\ min_{gdl} - \rho \le x_1 \le 1 + \rho, \\ 0 - \rho \le x_2 \le max_{ch} + \rho, \\ \rho \ge 0 \end{cases} \tag{5.85b}$$

where H is the control horizon, z_k are the auxiliary variables, $x_k = [\, x_1[k],\, x_2[k]\,]^T$ is the state of the MLD system at sampling time k, $\xi \triangleq [u_0^T, \ldots, u_{H-1}^T, z_0^T, \ldots, z_{H-1}^T, \rho]^T \in \mathbb{R}^{3H+1} \times \{0,1\}^{2H}$ is the optimization vector, Z, R and T are weight matrices, and $Z_\rho = 10^5$ is a weight used to enforce the softened version (Equation 5.85b) of a constraint on accumulation. In particular, we define the reference signals used in Equation 5.85 as

$$y_{ref} \triangleq v_0, \quad u_{ref} \triangleq 0, \quad x_{ref} \triangleq [1\ 0]'. \tag{5.86}$$

The minimum distance to which the liquid front is permitted to recede into the GDL, $min_{gdl} = 0.3$, is chosen to prevent overdrying of the membrane. The maximum liquid water accumulation, $max_{ch} = 20 \times 10^{-4}$ kg, is chosen to prevent a large voltage drop due to water accumulation in the channel, since x_2 is related to the voltage by Equation 5.81. These constraints are treated as soft, and hence can be violated, but at the price of a large increase in cost. In this way, the MPC controller will try to avoid violating them as much as possible. Soft constraints which are chosen to be more conservative, help account for modeling error and mitigate the risk of damaging the system.

Problem (Equation 5.85) can be transformed into a mixed integer quadratic program (MIQP), that is, the minimization of a quadratic cost function subject to linear constraints, where some of the variables are binary. Even though this class of problems has exponential complexity, efficient numerical tools for its solution are available [79].

5.11.8 Switching MPC Controller

Since the MPC is based on a PWA model parameterized for a set of constant conditions, Ω, the performance of the controller degrades away from the nominal values. Since we are using state feedback, the controller is less sensitive to model mismatch, however if the model is completely different, closed-loop dynamics will certainly be unsatisfactory. In our case study, the stack current can vary over a wide range

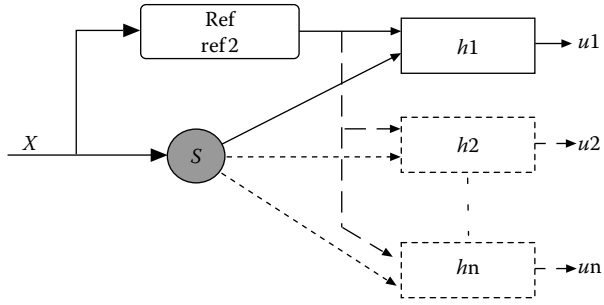

FIGURE 5.18 Switching MPC controller architecture.

of current density $i_{fc} \in [0.09 - 0.94]$ A cm^{-2}, and the parameters $Q(\omega)$ will also vary significantly, hence a stack of controllers based on different PWA models is proposed as depicted in Figure 5.18. A switching entity *Sw* activates a single controller depending on the actual operating conditions $C_i(\Omega(t))$ as well as the reference selector. The complexity in terms of computational time is not affected, because with this structure only the activated controller has to compute the control law at each sample time. The switching strategy implemented in *Sw* and used in simulation (Figure 5.20) is based on a threshold of current. A more robust switching strategy may be desirable, to avoid chattering phenomena due to noisy signals. A possible solution for improving hybrid state estimation is based on Kalman filtering coupled with stochastic discrete state estimators [80].

5.11.9 Simulation Results

The closed-loop behavior of the fuel cell with the MPC controller has been evaluated in simulation by using the high-fidelity nonlinear model described in the first half of the article. The controller designed using MPC (Equation 5.85), depends on the prediction horizon $H = 10$, and weights

$$Z = 1, \quad R = 1, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{5.87}$$

for a given PWA model parameterized at $\Omega = [10, 2, 333]$. In Figure 5.19, we consider a simulation scenario with a step change in I_{fc} to check the closed-loop robustness to a model mismatch and performance around the nominal value of Ω. The reference voltage, $y_{ref} = 755.25$ mV, and the initial conditions for the simulation are $x_1(0) = 1$, $x_2(0) = 1.13$, $y(0) = 749.43$ mV and $u_1(0) = 0$. At time $t = 423$ s the first purge is commanded by the controller. The purge duration is controlled to 0.3 s*. The voltage is restored to y_{ref}, as shown in Figure 5.19b, the anode channel is drained, Figure 5.19a, and the waterfront recedes into the GDL. After 692 s (Δ_{C_1}), at $t = 1115$ s, a second purge is required, the purge duration is again $\delta = 0.3$ s. At time $t = 1500$ s the stack current changes from 10 to 15 A, this leads to a sudden decrease of both actual and reference voltage, y and y_{ref}, respectively. At higher current density, the water accumulation and the voltage degradation rate are faster. The controller reacts by commanding purges at $t = 1719$ s and again at $t = 2171$ s, with a shorter period $\Delta = 452$ s. At time $t = 2500$ s the current increases to $I_{fc} = 20$ A and a purge is required. At $t = 2610.3$ s another purge is commanded and again every $\Delta = 110.3$ s. The controller tuned for lower current density is overly conservative at $I_{fc} = 20$, in an attempt to prevent a large voltage drop, the controller utilizes frequent purging at the expense of efficiency due to wasted hydrogen. This problem can be addressed by using switching MPC

* It will be evident later that $\delta \equiv T_s$ is not changing because the channel is fully purged in one sampling time, hence a faster or adaptive sampling rate is needed to control the water purged from the channel.

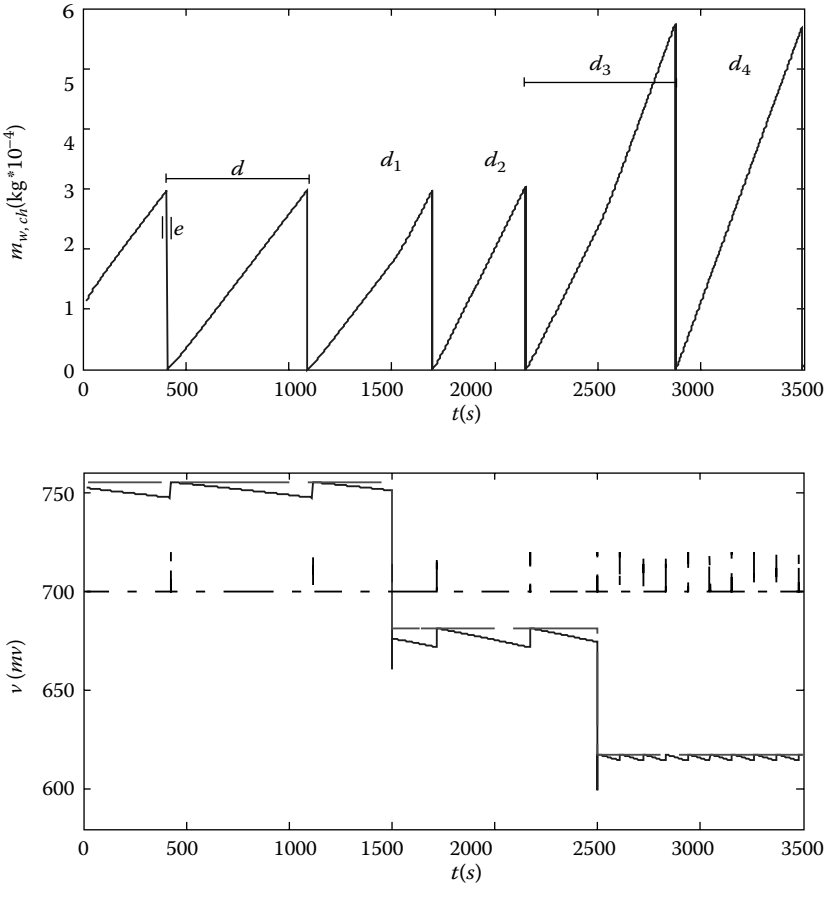

FIGURE 5.19 Closed-loop response.

controller with subcontrollers based on the PWA model parameterized at several desired operating conditions.

5.11.10 Simulation with Switching MPC

The second simulation, shown in Figure 5.20, presents the closed-loop performance with a switched MPC controller for two different sets of weights used in the cost function (Equation 5.85). The model inputs for the simulation are exactly the same as shown in Figure 5.19. The second controller is based on the PWA model parameterized at $\Omega = [20,\ 2,\ 333]$, with the switching threshold set at $I_{fc} = 16$ A. From $t = 0$ to $t = 2500$ the results are the same as the first simulation. At $t = 2500$ the current increases to 20 A and crosses the switching threshold and activating controller C_2. The solid line in Figure 5.20 represents a realization of the controller, C_{2A}, using the same weights as C_1 (Equation 5.87). By inspection of the solid line in Figure 5.20, before and after the load change at $t = 2500$ s, we see that the controller C_{2A} allows the same amount of water accumulation as at the lower current density by choosing a shorter purge period. In this case, the switched MPC can correctly predict the fast accumulation at higher currents and perform consistently with the tuning parameters.

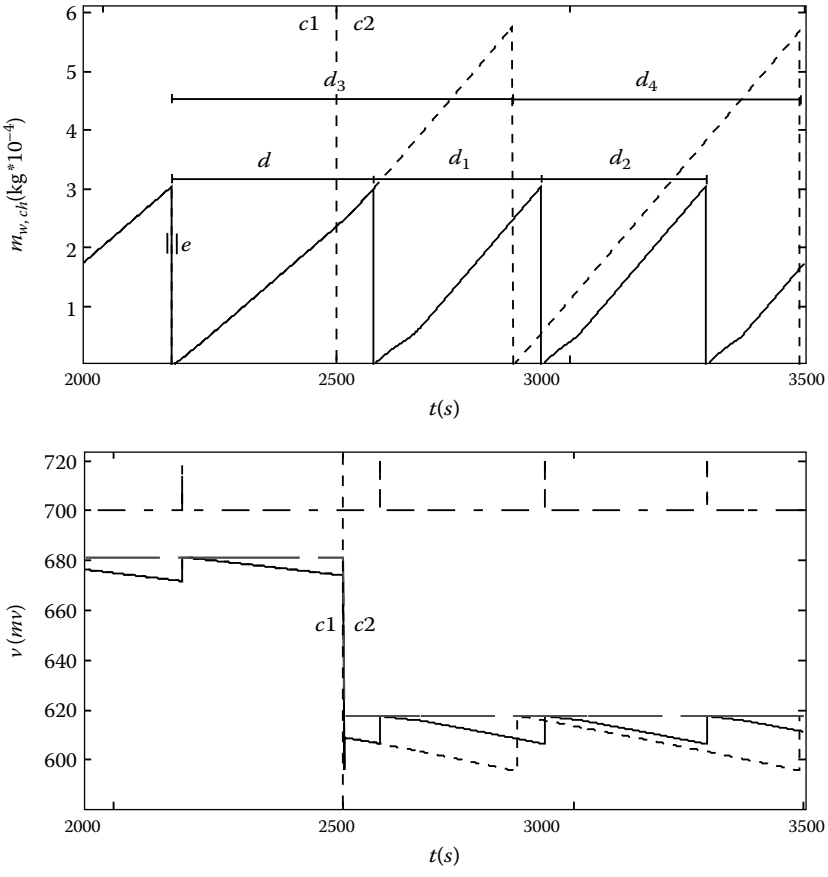

FIGURE 5.20 Closed-loop response for switching MPC.

The second realization of controller C_{2B}, shown as the dotted line in Figure 5.20, is computed with horizon $H = 10$, and weights

$$Z_2 = 1e - 1, \quad R_2 = 10, \quad S_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{5.88}$$

We select lower weights on x_2 and on y and a larger weight on u_1 to penalize the valve opening. The purpose is to reduce the hydrogen consumption at the expense of voltage and accumulation reference tracking, hence achieving a larger purge period. Although the controller C_{2B} has better hydrogen utilization, it has a lower safety margin in terms of avoiding a low voltage constraint, and could only be used during a highway driving scenario when changes in load current are infrequent.

It took approximately 173.3 s to simulate the first scenario of the closed-loop system on a PC Intel Centrino Duo 2.0 GHz with 2 GB RAM running the Hybrid Toolbox for Matlab [73] and the MIQP solver CPLEX 9 [79], of which 105.9 s are spent by CPLEX. That is an average of approximately 9 ms per time step. Because of the excessive CPU requirements for online optimization and because of the complexity of the software for solving the mixed-integer programs, the MPC controller cannot be directly implemented in an embedded micro-controller or in a computer not equipped with optimization software. To circumvent such implementation problems, in the next section

we compute an *explicit* version of the MPC controller that does not require online mixed-integer optimization.

5.11.11 Explicit Hybrid MPC Controller

Since the MPC controller based on the optimal control problem cannot be directly implemented in a standard embedded micro-controller, as it would require an MIQP to be solved online, the design of the controller is performed in two steps. First, the MPC controller is tuned in simulation using MIQP solvers, until the desired performance is achieved. Then, for implementation purposes, the explicit PWA form of the MPC law is computed off-line by using a combination of multiparametric quadratic programming [81] and dynamic programming, and implemented in the Hybrid Toolbox [73]. The value of the resulting PWA control function is identical to the one which would be calculated by the MPC controller (Equation 5.85), but the online complexity is reduced to the simple function evaluation instead of online optimization.

As shown in [82], the explicit representation $u(t) = f(\theta(t))$ of the MPC law (Equation 5.85) is represented as a collection of affine gains over (possibly overlapping) polyhedral partitions of the set of parameters

$$\theta = [x_1 \; x_2 \; y \; x_{1,ref} \; x_{2,ref} \; y_{ref}]'$$

For the control horizon $H = 10$ we obtain a PWA control law defined over 11,304 polyhedral regions. Reducing the horizon to $H = 3$ and eliminating the constraint on $x_2 \leq max_{ch}$ that is always inactive during our simulation scenarios the number of partition can be reduced to 64.

The performance with $H = 3$ is comparable to the performance achieved with a longer horizon. With careful tuning on the weights it is possible to obtain the same results shown in the previous section, but with a total simulation time reduced from 173.3 to 33.6 s on the same computer platform.

5.12 Conclusions and Future Work

In this chapter, a model for the water and nitrogen accumulation in the DEA of a PEMFC was presented. Several simplifying steps were presented in order to describe the slowly evolving liquid water front in the GDL and channel using ODEs, which yields a reduction in the computational complexity of the model when compared with the traditional approach of solving the coupled two-phase diffusion PDE. The inclusion of catalyst flooding and the resulting impact on membrane water transport highlights the level of complexity our model can capture. The model relies on calibration of three parameters for water transport, listed in Table 5.2, and the five parameters in the static voltage equation, which are listed in Table 5.3. After calibration, the model can be used to estimate the membrane water content, the rate of water crossover through the membrane and liquid water accumulation in the anode channel, all of which impact fuel cell performance.

We then presented a systematic approach for developing a controller for purge management in a fuel cell with DEA, which combines PWA modeling and hybrid MPC. The explicit implementation of the MPC, in the form of a PWA control law computed off-line, obviates the need for online optimization altogether and makes the overall approach suitable for implementation in devices with reduced computational resources. This implemention uses a state feedback of the front location in the anode GDL, x_{fr}, and the water level in the anode channel x_{ch}; hence, additional effort is required for the estimation of this information from conventional FC measurements, such as voltage. The promising results in terms of PWA modeling suggest a possible way to approach the problem with a hybrid observer, for instance based on stochastic hybrid state estimators [80].

References

1. E. Kimball, T. Whitaker, Y. G. Kevrekidis, and J. B. Benziger, Drops, slugs, and flooding in polymer electrolyte membrane fuel cells, *AIChE Journal*, vol. 54, no. 5, pp. 1313–1332, 2008.
2. N. Yousfi-Steiner, P. Mocoteguy, D. Candusso, D. Hissel, A. Hernandez, and A. Aslanides, A review on PEM voltage degradation associated with water management: Impacts, influent factors and characterization, *Journal of Power Sources*, vol. 183, pp. 260–274, 2008.
3. W. Baumgartner, P. Parz, S. Fraser, E. Wallnöfer, and V. Hacker, Polarization study of a PEMFC with four reference electrodes at hydrogen starvation conditions, *Journal of Power Sources*, vol. 182, no. 2, pp. 413–421, 2008.
4. W. Schmittinger and A. Vahidi, A review of the main parameters influencing performance and durability of PEM fuel cells, *Journal of Power Sources*, vol. 180, pp. 1–14, 2008.
5. J. P. Meyers and R. M. Darling, Model of carbon corrosion in PEM fuel cells, *Journal of the Electrochemical Society*, vol. 153, no. 8, pp. A1432–A1442, 2006.
6. N. Pekula, K. Heller, P. A. Chuang, A. Turhan, M. M. Mench, J. S. Brenizer, and K. Ünlü, Study of water distribution and transport in a polymer electrolyte fuel cell using neutron imaging, *Nuclear Instruments and Methods in Physics Research Section A*, vol. 542, pp. 134–141, 2005.
7. J. B. Siegel, D. A. McKay, A. G. Stefanopoulou, D. S. Hussey, and D. L. Jacobson, Measurement of liquid water accumulation in a PEMFC with dead-ended anode, *Journal of the Electrochemical Society*, vol. 155, no. 11, pp. B1168–B1178, 2008.
8. B. A. McCain, A. G. Stefanopoulou, and I. V. Kolmanovsky, On the dynamics and control of through-plane water distributions in PEM fuel cells, *Chemical Engineering Science*, vol. 63, no. 17, pp. 4418–4432, 2008.
9. R. P. O'Hayre, S.-W. Cha, W. Colella, and F. B. Prinz, *Fuel Cell Fundamentals*. Hoboken, NJ: Wiley, 2006.
10. A. Shah, G.-S. Kim, W. Gervais, A. Young, K. Promislow, J. Li, and S. Ye, The effects of water and microstructure on the performance of polymer electrolyte fuel cells, *Journal of Power Sources*, vol. 160, no. 2, pp. 1251–1268, 2006.
11. DOE, Hydrogen and fuel cell activities, progress, and plans, DOE, Tech. Rep., 2009. Available: http://www.hydrogen.energy.gov/pdfs/epactreportsec811.pdf.
12. J. T. Pukrushpan, A. G. Stefanopoulou, and H. Peng, *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis and Feedback Design*. New York: Springer, 2000.
13. K.-W. Suh and A. G. Stefanopoulou, Performance limitations of air flow control in power-autonomous fuel cell systems, *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 465–473, 2007.
14. A. Vahidi, A. Stefanopoulou, and H. Peng, Current management in a hybrid fuel cell power system: A model-predictive control approach, *IEEE Transactions on Control Systems Technology*, vol. 14, no. 6, pp. 1047–1057, November 2006.
15. W. K. Na and B. Gou, Feedback-linearization-based nonlinear control for pem fuel cells, *IEEE Transactions on Energy Conversion*, vol. 23, no. 1, pp. 179–190, March 2008.
16. A. Vahidi, I. Kolmanovsky, and A. Stefanopoulou, Constraint handling in a fuel cell system: A fast reference governor approach, *IEEE Transactions on Control Systems Technology*, vol. 15, no. 1, pp. 86–98, 2007.
17. F. Y. Zhang, X. G. Yang, and C. Y. Wang, Liquid water removal from a polymer electrolyte fuel cell, *Journal of the Electrochemical Society*, vol. 153, no. 2, pp. A225–A232, 2006.
18. E. Kumbur, K. Sharp, and M. Mench, Liquid droplet behavior and instability in a polymer electrolyte fuel cell flow channel, *Journal of Power Sources*, vol. 161, no. 1, pp. 333–345, 2006.
19. D. A. McKay, A. G. Stefanopoulou, and J. Cook, A membrane-type humidifier for fuel cell applications: Controller design, analysis and implementation, *ASME Conference Proceedings*, vol. 2008, no. 43181, pp. 841–850, 2008.
20. H. P. Dongmei Chen, Analysis of non-minimum phase behavior of PEM fuel cell membrane humidification systems, in *Proceedings of the 2005 American Control Conference*, pp. 3853–3858, vol. 6, 2005.
21. E. A. Müller and A. G. Stefanopoulou, Analysis, modeling, and validation for the thermal dynamics of a polymer electrolyte membrane fuel cell system, *Journal of Fuel Cell Science and Technology*, vol. 3, no. 2, pp. 99–110, 2006.
22. X. Huang, R. Solasi, Y. Zou, M. Feshler, K. Reifsnider, D. Condit, S. Burlatsky, and T. Madden, Mechanical endurance of polymer electrolyte membrane and PEM fuel cell durability, *Journal of Polymer Science Part B: Polymer Physics*, vol. 44, no. 16, pp. 2346–2357, 2006.

23. F. A. d. Bruijn, V. A. T. Dam, and G. J. M. Janssen, Review: Durability and degradation issues of PEM fuel cell components, *Fuel Cells*, vol. 8, pp. 3–22, 2008.
24. C. A. Reiser, L. Bregoli, T. W. Patterson, J. S. Yi, J. D. Yang, M. L. Perry, and T. D. Jarvi, A reverse-current decay mechanism for fuel cells, *Electrochemical and Solid-State Letters*, vols 8-6, pp. A273–A276, 2005.
25. M. Schulze, N. Wagner, T. Kaz, and K. Friedrich, Combined electrochemical and surface analysis investigation of degradation processes in polymer electrolyte membrane fuel cells, *Electrochimica Acta*, vol. 52, no. 6, pp. 2328–2336, 2007.
26. A. Karnik, J. Sun, and J. Buckland, Control analysis of an ejector based fuel cell anode recirculation system, in *American Control Conference*, 2006, Minneapolis, MN, 6 pp, June 2006.
27. R. K. Ahluwalia and X. Wang, Fuel cell systems for transportation: Status and trends, *Journal of Power Sources*, vol. 177, no. 1, pp. 167–176, 2008.
28. S. S. Kocha, J. D. Yang, and J. S. Yi, Characterization of gas crossover and its implications in PEM fuel cells, *AIChE Journal*, vol. 52, no. 5, pp. 1916–1925, 2006.
29. R. Ahluwalia and X. Wang, Buildup of nitrogen in direct hydrogen polymer–electrolyte fuel cell stacks, *Journal of Power Sources*, vol. 171, no. 1, pp. 63–71, 2007.
30. A. Y. Karnik and J. Sun, Modeling and control of an ejector based anode recirculation system for fuel cells, in *Proceedings of the Third International Conference on Fuel Cell Science, Engineering, and Technology*, Ypsilanti, MI, FUELCELL2005–74102, 2005.
31. A. Karnik, J. Sun, A. Stefanopoulou, and J. Buckland, Humidity and pressure regulation in a PEM fuel cell using a gain-scheduled static feedback controller, *IEEE Transactions on Control Systems Technology*, vol. 17, no. 2, pp. 283–297, 2009.
32. P. Rodatz, A. Tsukada, M. Mladek, and L. Guzzella, Efficiency improvements by pulsed hydrogen supply in PEM Fuel cell systems, in *Proceedings of IFAC 15th World Congress*, Barcelona, Spain, 2002.
33. S. Hikita, F. Nakatani, K. Yamane, and Y. Takagi, Power-generation characteristics of hydrogen fuel cell with dead-end system, *JSAE Review*, vol. 23, pp. 177–182, 2002.
34. L. Dumercy, M.-C. Péra, R. Glises, D. Hissel, S. Hamandi, F. Badin, and J.-M. Kauffmann, PEFC stack operating in anodic dead end mode, *Fuel Cells*, vol. 4, pp. 352–357, 2004.
35. E. A. Müller, F. Kolb, L. Guzzella, A. G. Stefanopoulou, and D. A. McKay, Correlating nitrogen accumulation with temporal fuel cell performance, *Journal of Fuel Cell Science and Technology*, vol. 7, no. 2, 021013, 2010.
36. A. Z. Weber, Gas-crossover and membrane-pinhole effects in polymer–electrolyte fuel cells, *Journal of the Electrochemical Society*, vol. 155, no. 6, pp. B521–B531, 2008.
37. H. Gorgun, F. Barbir, and M. Arcak, A voltage-based observer design for membrane water content in PEM fuel cells, in *Proceedings of the 2005 American Control Conference*, vol. 7, June 2005, pp. 4796–4801.
38. G. Ripaccioli, J. B. Siegel, A. G. Stefanopoulou, and S. Di Cairano, Derivation and simulation results of a hybrid model predictive control for water purge scheduling in a fuel cell, in *The 2nd Annual Dynamic Systems and Control Conference*, Hollywood, CA, USA, October 12–14, 2009.
39. S. Dutta, S. Shimpalee, and J. Van Zee, Three-dimensional numerical simulation of straight channel PEM fuel cells, *Journal of Applied Electrochemistry*, vol. 30, no. 2, pp.135–146, 2000.
40. D. Cheddie and N. Munroe, Review and comparison of approaches to proton exchange membrane fuel cell modeling, *Journal of Power Sources*, vol. 147, no. 1-2, pp. 72–84, 2005.
41. D. A. McKay, J. B. Siegel, W. Ott, and A. G. Stefanopoulou, Parameterization and prediction of temporal fuel cell voltage behavior during flooding and drying conditions, *Journal of Power Sources*, vol. 178, no. 1, pp. 207–222, 2008.
42. A. J. d. Real, A. Arce, and C. Bordons, Development and experimental validation of a PEM fuel cell dynamic model, *Journal of Power Sources*, vol. 173, no. 1, pp. 30–324, 2007.
43. P. Berg, K. Promislow, J. S. Pierre, J. Stumper, and B. Wetton, Water management in PEM fuel cells, *Journal of the Electrochemical Society*, vol. 151, no. 3, pp. A341–A353, 2004.
44. A. Stefanopoulou, I. Kolmanovsky, and B. McCain, A dynamic semi-analytic channel-to-channel model of two-phase water distribution for a unit fuel cell, *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1055–1068, 2009.
45. M. M. Mench, *Fuel Cell Engines*. Hoboken, NJ: John Wiley & Sons, 2008.
46. W. Borutzky, B. Barnard, and J. Thoma, An orifice flow model for laminar and turbulent conditions, *Simulation Modelling Practice and Theory*, vol. 10, no. 3-4, pp. 141–152, 2002.
47. J. B. Siegel, D. A. McKay, and A. G. Stefanopoulou, Modeling and validation of fuel cell water dynamics using neutron imaging, in *Proceedings of the American Control Conference*, June 11–13, Seattle, WA, pp. 2573–2578, 2008.

48. J. Nam and M. Kaviany, Effective diffusivity and water-saturation distribution in single and two-layer PEMFC diffusion medium, *International Journal of Heat Mass Transfer*, vol. 46, pp. 4595–4611, 2003.
49. J. T. Gostick, M. W. Fowler, M. A. Ioannidis, M. D. Pritzker, Y. Volfkovich, and A. Sakars, Capillary pressure and hydrophilic porosity in gas diffusion layers for polymer electrolyte fuel cells, *Journal of Power Sources*, vol. 156, no. 2, pp. 375–387, 2006.
50. B. Markicevic, A. Bazylak, and N. Djilali, Determination of transport parameters for multiphase flow in porous gas diffusion electrodes using a capillary network model, *Journal of Power Sources*, vol. 172, pp. 706–717, 2007.
51. E. C. Kumbur, K. V. Sharp, and M. M. Mench, Validated Leverett approach for multiphase flow in PEFC diffusion media ii. Compression effect, *Journal of the Electrochemical Society*, vol. 154, pp. B1305–B1314, 2007.
52. Z. H. Wang, C. Y. Wang, and K. S. Chen, Two-phase flow and transport in the air cathode of proton exchange membrane fuel cells, *Journal of Power Sources*, vol. 94, pp. 40–50, 2001.
53. K. Promislow, P. Chang, H. Haas, and B. Wetton, Two-phase unit cell model for slow transients in polymer electrolyte membrane fuel cells, *Journal of the Electrochemical Society*, vol. 155, no. 7, pp. A494–A504, 2008.
54. M. Grötsch and M. Mangold, A two-phase PEMFC model for process control purposes, *Chemical Engineering Science*, vol. 63, pp. 434–447, 2008.
55. S. Ge, X. Li, B. Yi, and I.-M. Hsing, Absorption, desorption, and transport of water in polymer electrolyte membranes for fuel cells, *Journal of the Electrochemical Society*, vol. 152, no. 6, pp. A1149–A1157, 2005.
56. W. Huang, L. Zheng, and X. Zhan, Adaptive moving mesh methods for simulating one-dimensional groundwater problems with sharp moving fronts, *International Journal for Numerical Methods in Engineering*, vol. 54, no. 11, pp. 1579–1603, 2002.
57. J. B. Siegel, S. Yesilyurt, and A. G. Stefanopoulou, Extracting model parameters and paradigms from neutron imaging of dead-ended anode operation, in *Proceedings of FuelCell2009 Seventh International Fuel Cell Science, Engineering and Technology Conference*, Newport Beach, CA, 2009.
58. S. Basu, C.-Y. Wang, and K. S. Chen, Phase change in a polymer electrolyte fuel cell, *Journal of the Electrochemical Society*, vol. 156, no. 6, pp. B748–B756, 2009.
59. L. Onishi, J. Prausnitz, and J. Newman, Water-Nafion equilibria. Absence of Schroeder's paradox, *Journal of Physical Chemistry B*, vol. 111, no. 34, pp. 10 166–10 173, 2007.
60. T. Springer, T. Zawodzinski, and S. Gottesfeld, Polymer electrolyte fuel cell model, *Journal of the Electrochemical Society*, vol. 138, no. 8, pp. 2334–2341, 1991.
61. J. Hinatsu, M. Mizuhata, and H. Takenaka, Water uptake of perfluorosulfonic acid membranes from liquid water and water vapor, *Journal of the Electrochemical Society*, vol. 141, pp. 1493–1498, 1994.
62. S. Ge, B. Yi, and P. Ming, Experimental determination of electro-osmotic drag coefficient in Nafion membrane for fuel cells, *Journal of the Electrochemical Society*, vol. 153, no. 8, pp. A1443–A1450, 2006.
63. J. Laraminie and A. Dicks, *Fuel Cell Systems Explained*, 2nd ed. Hoboken, NJ: Wiley InterScience, 2003.
64. F. Barbir, *PEM Fuel Cells: Theory and Practice*. Burlington, MA: Elsevier, 2005.
65. D. M. Bernardi and M. W. Verbrugge, A mathematical model of the solid-polymer-electrolyte fuel cell, *Journal of the Electrochemical Society*, vol. 139, no. 9, pp. 2477–2491, 1992.
66. J. S. Newman., *Electrochemical Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
67. M. Branicky, Studies in hybrid systems: Modeling, analysis, and control, Ph.D. dissertation, LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA, 1995.
68. W. Heemels, B. D. Schutter, and A. Bemporad, Equivalence of hybrid dynamical models, *Automatica*, vol. 37, no. 7, pp. 1085–1091, 2001.
69. A. Bemporad and M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
70. F. Torrisi and A. Bemporad, Hysdel—a tool for generating computational hybrid models, *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235–249, 2004.
71. A. Bemporad, Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form, *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 832–838, 2004.
72. T. Geyer, F. Torrisi, and M. Morari, Efficient mode enumeration of compositional hybrid models, in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, A. Pnueli and O. Maler, Eds. Berlin, Springer-Verlag, vol. 2623, pp. 216–232, 2003.
73. A. Bemporad, *Hybrid Toolbox—User's Guide*, jan 2004, http://www.dii.unisi.it/hybrid/toolbox.
74. T. Henzinger, The theory of hybrid automata, in *Logic in Computer Science. LICS'96. Proceedings, Eleventh Annual IEEE Symposium on*, pp. 278–292, 1996.

75. S. Di Cairano and A. Bemporad, An equivalence result between linear hybrid automata and piecewise affine systems. in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, pp. 2631–2636, 2006.
76. F. Borrelli, A. Bemporad, M. Fodor, and D. Hrovat, An MPC/hybrid system approach to traction control, *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 541–552, 2006.
77. G. Ripaccioli, A. Bemporad, F. Assadian, C. Dextreit, S. Di Cairano, and I. Kolmanovsky, Hybrid modeling, identification, and predictive control: An application to hybrid electric vehicle energy management, in *Hybrid Systems: Computation and Control*, vol. 5469. Heidelberg: Springer Berlin, pp. 321–335, 2009.
78. N. Giorgetti, G. Ripaccioli, A. Bemporad, I. Kolmanovsky, and D. Hrovat, Hybrid model predictive control of direct injection stratified charge engines, *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 5, pp. 499–506, 2006.
79. ILOG, Inc., *CPLEX 9.0 User Manual*, Gentilly Cedex, France, 2003.
80. S. Di Cairano, K. Johansson, A. Bemporad, and R. Murray, Dynamic network state estimation in networked control systems, in *Hybrid Systems: Computation and Control*, ser. *Lecture Notes in Computer Science*, M. Egerstedt and B. Mishra, Eds. Berlin: Springer-Verlag, no. 4981, pp. 144–157, 2008.
81. A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
82. F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, Dynamic programming for constrained optimal control of discrete-time linear hybrid systems, *Automatica*, vol. 41, no. 10, pp. 1709–1721, 2005.

II

Aerospace

6

Aerospace Real-Time Control System and Software

Rongsheng (Ken) Li
The Boeing Company

Michael Santina
The Boeing Company

6.1 Introduction

6.1.1 Control-Centric Aerospace Systems

Control system theory and design practices are used widely across the aerospace industry, especially in creating aircraft, missile and spacecraft guidance, navigation and control (GN&C) systems. They have been critical to the success of many aerospace systems in the past and will be in the future, including the historically significant control-centric systems/missions such as Apollo moon landing mission, International Space Station, Space Shuttle, Boeing 747 aircraft, and Global Positioning System (GPS). Some of the typical examples of "control theory intensive systems" are listed below:

- Aircraft and missile flight control system (FCS)
- Aircraft navigation systems
- Aircraft and missile vehicle management systems
- Aircraft flight management systems
- Aircraft fuel management systems
- Aircraft collision avoidance systems
- Aircraft payload pointing control systems
- Spacecraft attitude control systems
- Spacecraft thermal control systems
- Spacecraft power management systems
- Spacecraft orbit and attitude determination systems
- Spacecraft vehicle management systems
- Spacecraft payload pointing control systems

These systems are becoming more and more sophisticated and their implementations typically involve significant amount of real-time software and in fact it is virtually impossible to find an aerospace control system today that does not use software as the primary means of implementing its complicated logic and algorithms.

This chapter deals with the topic of flight software for the real-time control systems used by the aerospace systems. This topic requires a separate treatment due to the safety or mission critical nature of aerospace systems that are not as frequently found in the other industries.

This chapter discusses the special nature of aerospace system real-time software; the safety and criticality standard; the development process; the architecture; the development, test, integration, verification and validation (V&V) approaches, and options. Emphasis will be given to topics that are interesting to control system engineers and analysts.

6.1.2 Advancement of Control Systems That Are Enabled by Computing Systems

Over the past 40 years, the growth of capability of control systems exactly parallels the growth of the computing systems. The computing system enables the implementation of more and more complicated control algorithms, and that in turn, demands faster, more reliable and more powerful computing capabilities; which drive the development of computer hardware and software technology.

Early control systems used mechanical mechanism (or mechanical computers) to implement control algorithms.

Analog computer was later used to provide more capabilities at lower cost, with lower mass and volume. Digital computers, especially what was then called microcomputers, indicating that the central processing unit (CPU) being on a single chip, has been the revolutionary force for the entire human civilization and undoubtedly has been the most important enabler for modern control systems.

The further development of high-density, highly integrated circuits such as field programmable gate array (FPGA) and application specific integrated circuit (ASIC) has also made it possible that general purpose computing engines (e.g., CPUs) can be replaced by ASICs and FPGAs and the software can be replaced by hardware logic in many applications.

6.1.3 Increasing Role of Control System Engineers with the Technology Advancement

Obviously, control system engineers need to be responsible for the development of system concept, system requirement, system architecture, detailed system design, detailed algorithm design; and then the validation of the detailed algorithms by simulation, the validation of the system design by simulation, and the integration and test of the system. Beyond these responsibilities, the advancement of technology and tools and the demand for better/cheaper/fast development of systems are requiring control system engineers to take more responsibilities in the implementation of the system in terms of the software or "Code" for FPGAs and ASICs.

This increased role is driven by the ever-increasing complexity and the cost of system and software development. Specifically,

1. The increase of the complexity has made it necessary for the architecture of the algorithm and the software to be developed by people who have the domain knowledge. Combined control system domain expertise and system and software architecture expertise have proven to be critical for the efficient and successful development of complex, performance-demanding, safety-critical aerospace systems and software.
2. The increase of the complexity has made it necessary that the person who writes and debugs the software that implements the algorithm to have sufficient domain knowledge to be efficient.
3. The modern iterative development approach (such as IBM's Rational Unified Process (RUP), which has proven to be superior to the traditional waterfall process for software development, demands the capability for fast iterations, which consists of tightly-coupled requirement, architecture, detailed design, implementation, integration, and test activities. Traditional "system-engineering-throw-over-wall-to-software-engineering" approach simply does not fit the iterative development approach for complex control systems.

This increasing role is also enabled by the advanced software, FPGA and ASIC development tools, and modern system and software architecture. Specifically,

1. Modern layered software architecture allows control application software to be mostly independently developed with focus on the control system itself.
2. Advanced tools, languages, and libraries and component-based architecture allow the application software to be developed at a high level of abstraction. For example, component-based programming using C++ and Java and autocode generation from algorithms expressed in MATLAB®, Simulink®, or MATRIXx™ have in many cases removed the need for the inefficient hand-off of the design to a different skill team for the implementation and debugging.

Consequently, the increasing role requires aerospace control system engineers to not only have the domain expertise on aerospace control system engineering but also to have expertise in software

engineering especially in the areas of software architecture, software integration, and test and software development process. In fact, this chapter is motivated by these needs.

6.2 Architecture of Aerospace Real-Time Control Systems

6.2.1 Typical Sensor/Computer/Actuator/User Interface Logical Architecture

Among the many different aerospace control systems found in the industry, they exhibit the basic pattern of "sensor/computer/actuator/User Interface (UI)" architecture, as illustrated in Figure 6.1.

It is advantageous to view this architecture as a "logical architecture" rather than "the physical architecture." When viewed as a logical architecture, the "sensor" or "actuator" elements are logically providing the capabilities of the sensors or the capabilities of the actuator. The physical implementation of the sensor or actuator itself can contain computers and other electronics. Similarly the computer in this architecture is a "logical element" that provides the capabilities of a computer. Physically, the logical computer can be implemented by a single computer in the modern sense, a cluster of computers or even an ancient mechanical or analog computer.

An important key consequence of this simple architecture is that most of the system design and development efforts focus on what happens inside the computer. That is, the "logical software" which is typically realized by a real computer or several computers or even by FPGAs or ASICS.

6.2.2 Layered System Architecture

Figure 6.2 provides another view of "layered" architecture for the system. The "layered" architecture will be discussed in more detail in Section 6.3. For the purposes of this section, it is important to point out that in a layered architecture, the system is organized into groups of "components," each group called a layer. Typically, each layer is desired to address a different level of "concern" and requires different skills and knowledge base to define, design, and implement.

As shown in Figure 6.2, the bottom layer is the sensor, actuator, and UI hardware which provides the "input information and actions" and "output information and actions" between the system and the outside world. In other words, it is this layer of hardware devices that "interact" with the "outside world."

The next layer, which includes the sensor, actuator, and UI device interface hardware, provides the hardware interface between the computer hardware and the actual devices. If standard interfaces are used,

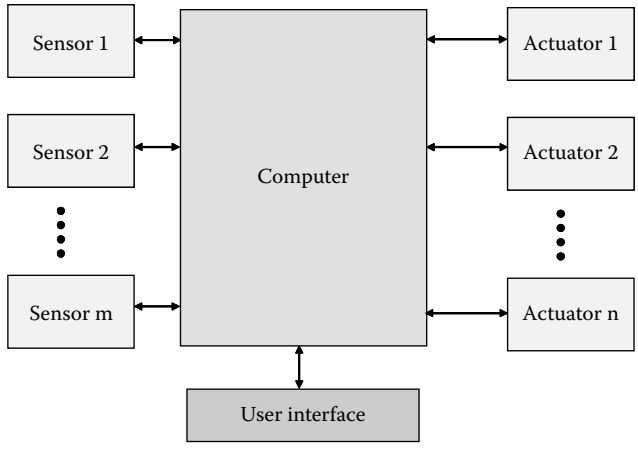

FIGURE 6.1 Generic sensor/computer/actuator/UI architecture.

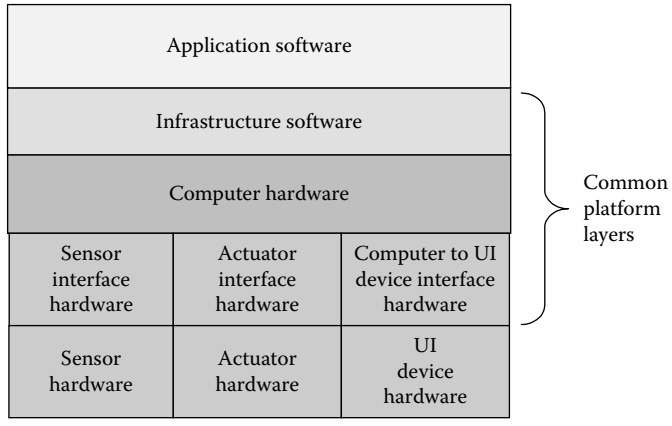

FIGURE 6.2 Layered system architecture for typical sensor/computer/actuator/UI pattern.

such as standard A/D, standard D/A, and standard serial data bus, this layer can be constructed using the standard hardware and can be used for systems constructed for very different purposes. Some examples of standard serial data buses are MIL-STD-1553, ARINC 429, ARINC 629, Ethernet, RS232, RS422, USB, IEEE 1394, and Spacewire.

The computer hardware layer provides the capability to execute software that performs computation and implements logic and algorithms that process the "input information" that comes from the sensor/actuator/user hardware interface layer and produce the "output information" for that layer as well.

The infrastructure software layer provides a "software abstraction" of the "computation" and "input/output" (I/O) capabilities provided from the hardware layers. This software abstraction greatly simplifies the task of developing the application software.

Finally, the application software implements the aerospace control system's detailed logic and algorithms which are executed by the computer hardware. The fact that the application software sits on the top of infrastructure software indicates that the development of application software does not have to be concerned with many details of the hardware but only has to be concerned with a "software abstraction" of the hardware and this is enabled by the infrastructure software.

The infrastructure software layer, the computer hardware layer, and the sensor/actuator/UI device hardware layer form a system that can be designed in such a way that the same design can be used for a large class of systems. We call these three layers a "platform" which can be reused for many different applications when it is "hooked up" with different sensor, actuator and UI devices, and different application software.

Consequently both the top and bottom layers are specific to a particular application system and the middle three layers are for general purposes. The bottom hardware layer provides the specific sensor, actuator, and UI devices that are required for the specific application. The application software implements the specific logic and algorithms for the specific application. The key requirements of this system architecture are therefore the following: (1) the application software talks to the infrastructure software through a standard interface and (2) the sensor, actuator, and UI devices talk to the interface hardware through standard interfaces.

This approach has been consciously and unconsciously applied to the architecture design of various systems in the industry and there has been a trend that the platform layers are becoming more and more general. Furthermore, the control application software and sensor and actuator hardware devices are becoming more and more independent of the "platform layers." In fact, Integrated Modular Avionics (IMA), which has been implemented for the modern Boeing 787 and Airbus A380 aircraft, represents the most recent implementations of this approach.

6.2.3 Distributed versus Centralized Physical Architecture

Distributed and centralized physical architectures are two important variations of the architecture when the logical system is realized by physical systems. When the logical computer is implemented by a physically centralized computing engine, the physical architecture is centralized. When the logical computer is implemented by physically distributed computing engines, the architecture is distributed. These two variations have their advantages and disadvantages and the choice of a particular architecture always requires careful evaluation and trade study for the specific application and specific objectives.

6.2.4 Integrated Modular Avionics

IMA is a significant advance in the aerospace control system architecture. IMA integrates systems that are traditionally implemented by separate physical "boxes" on aircraft by a uniform general purpose network of computers interconnected by high-speed communication data buses such as the modern Avionics Full-Duplex Switched Ethernet (AFDX) or the older ARINC 429 data bus. The sensor and actuator hardware are connected to this network and are accessible by any one of the computing engines across the network. The software modules that were traditionally developed individually for each of the "boxes" are now application software modules running on the computer network on the top of common hardware and low-level software resources.

IMA offers significant advantages in terms of reducing the development cost, hardware cost and weight, and the maintenance cost. For example,

1. Using a common Application Program Interface (API) to access the hardware and network resources greatly simplifies the hardware and software integration effort.
2. IMA allows the application developers to focus on the application layer, and consequently enables much more efficient development and simpler integration and test.
3. IMA reuses extensively tested hardware and lower-level software of the "platform layers" over time and across projects and even the community.
4. IMA allows application software to be relocated and reconfigured on spare computing engines if the hosting computing engine is detected faulty during operations. This reduces the cost of redundancy and fault management in the system.

IMA has been adopted by F22-Raptor [1], Airbus A380 [2], and Boeing 787, and several other important avionics systems [3].

ARINC 653 (Avionics Application Standard Software Interface) is a software specification for space and time partitioning. It defines an API for software of avionics, following the architecture of IMA. IMA implementations rely on ARINC 653 compliant real-time operating system/middleware to allow the memory and time available from a computer to be partitioned in a way as if they were multiple hardware computers.

6.2.5 System Architecture Development Approach

The development of system architecture is driven by two types of inputs: (1) the functional and usage requirement which is often provided as "use cases" or as "functional requirements" and (2) the quality attributes of the architecture which is often provided as "quality attribute scenarios."

The architecture can be developed with or without using an architecture model. It is highly recommended that the architecture is developed with the help of architecture models that can be visualized. Use of architecture modeling languages is recommended. For example, graphical modeling languages such as Integration Definition for Function Modeling (IDEF0), Unified Modeling Language (UML), and System Modeling Language (SysML) can help significantly to elicit, document and communicate the architecture. Use of traditional control system block diagrams also helps significantly the description of the architecture.

The functional and usage requirements drive the capabilities provided by the system for which the architecture is to be developed. The quality attributes drive the architecture decisions that are made not only to address functional requirements but also to address developmental and operational concerns such as how easily the design can be changed to accommodate changes of the requirements; how easily the system can be operated; and how the design reuses the capabilities the organization developed in the past.

There are many different methodologies available to derive the architecture from the above-mentioned two types of inputs. It is a good practice, however, to go through the following most important steps and each of the key steps can be augmented or elaborated using various methodology and tools:

1. Derive "black box functional requirement" by usage and requirement analysis. The black box requirements are the "external requirements" on the system that are independent of the internal design. It is most important to decide first what the "black box requirements" are to allow enough freedom of the internal design and the optimizations of the architecture. This step can be done informally without the help of an architecture model or can be derived from "use cases," that is, the usage scenarios using traditional IDEF0 functional analysis or using the more modern UML and SysML-based approach.
2. Propose candidate logical and/or physical system decompositions, that is, decompose the system into subsystems or components. This step can be done in many different ways. Regardless of what approach is used, strong domain knowledge is the key to deciding on how the system can be decomposed into smaller systems and how the smaller systems can work together to achieve the capabilities of the total system while still trying to achieve good "quality attribute response." The following approaches are recommended:
   a. Motivate the logical or physical decomposition by "functional decomposition." By decomposing the "main" capability of the system (not every capability), it often provides enough hints as to how the logical system or physical system can be decomposed.
   b. Use design patterns. Design patterns are architecture patterns that have been proven by past projects with known advantages and disadvantages. The use of design patterns greatly improves the probability of success. In fact, most of the architecture examples presented in this chapter offer design patterns that can be utilized by the user.
   c. Use design tactics and principles.
3. Flow requirement down to subsystems or components. This step can again be done in many different ways with or without using an architecture model. When an architecture model is used, IDEF0 and UML/SysML-based approaches can be used by elaborating how the system black-box capabilities are achieved by the collaboration of the subsystem or component capabilities.
4. Evaluate and improve the architecture by ensuring that the functional capabilities are supported by the architecture and compare the quality attribute responses between different candidate architectures or individual architecture decisions.
5. For each of the subsystems or components, Steps 2 through 4 can be repeated to hierarchically elaborate the architecture to a level that the lowest level of component is readily available or readily developed. It is important to note that this "readiness" and the "depth" of the architecture elaboration vary with the specific purposes and scenarios and the concerns to be addressed.

6.2.6 Example: Aircraft Flight Control System Architecture

Typical aircraft FCS follows the common patterns discussed earlier both in terms of the sensor/computer/actuator/UI pattern and in terms of the layered architecture.

For aircraft FCSs, the sensors typically include the rotation rate sensors such as gyros, accelerometers, and most importantly, static and dynamic pressure sensors at various locations of the aircraft to produce the input for "air data."

The actuators are typically the control surfaces and their mechanical, hydraulic, or electrical driving systems.

The flight computers are often 3-for-1 or 4-for-1 redundant due to the flight critical nature of the FCS. The flight control software typically handles the inner control loops for stability augmentation and outer-loops and pilot command shaping to allow handling and control of the aircraft by the pilot. In addition, autopilot capabilities are also provided by the flight control software.

The "UI" provides the capability for the pilot to control the aircraft. The interfaces are typically the stick and pedal as well as the displays and switches on the pilot control panel.

Figure 6.3 provides a top-level diagram of the FCS.

6.2.7 Example: Spacecraft Control System Architecture

Very similarly, typical spacecraft attitude control system follows the common patterns discussed earlier both in terms of the sensor/computer/actuator/UI pattern and in terms of the layered architecture. Typical spacecraft control system is shown in Figure 6.4.

Typical spacecraft FCS follows the common pattern discussed earlier. The typical sensors are

- Inertial reference units (IRUs) for spacecraft rotational rate sensing
- Star-tracker for spacecraft absolute attitude sensing
- Sun sensors for spacecraft attitude sensing during safing
- Temperature, current sensors for thermal control, and electrical power system (EPS) sensing.

The typical actuators are

- Thrusters for orbit raising, station-keeping, momentum management, and large maneuver attitude control
- Reaction wheel for precision attitude control
- Magnetic torque rod for momentum management.

The flight computer interfaces with the sensor and actuator devices and provides control to the spacecraft. The block diagram provides some top level ideas of the functional capability of the flight software. The "UI" is through the telemetry and command system.

6.2.8 Example: IMA for Boeing 787

Due to the many benefits of IMA, both Boeing 787 and Airbus 380 have adopted IMA-based avionics architecture. Boeing 787's IMA is based on what is called Common Core System (CCS) which was developed by Smith Industry (now General Electric).

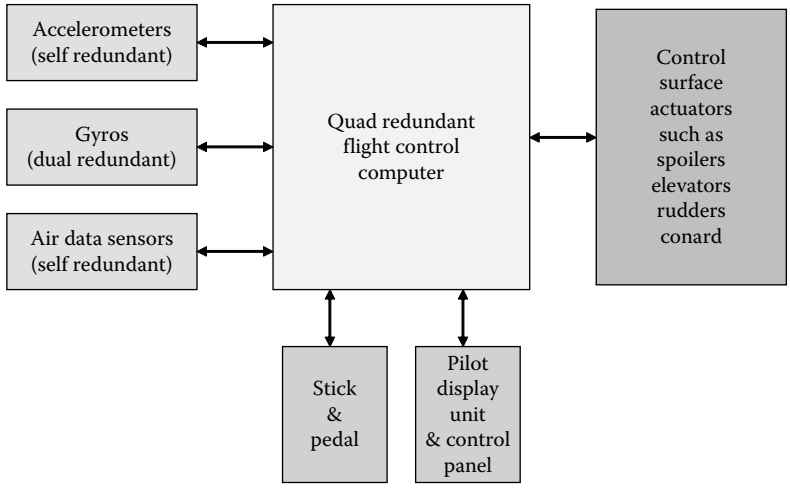

FIGURE 6.3 High-level block diagram of aircraft FCS.

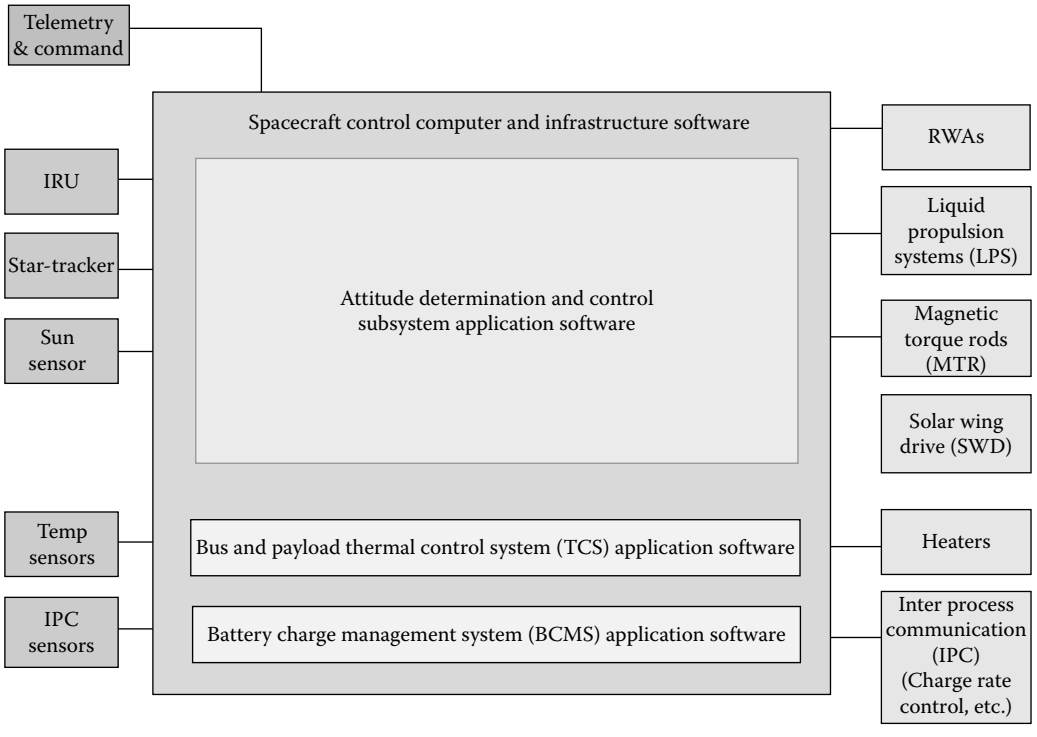

FIGURE 6.4 Typical spacecraft control system.

CCS is the 787's central nerve system. The CCS is a computing platform which includes dual common computing resource (CCR) cabinets that host processing and power control modules and network switches. Application specific modules (ASMs) can be installed in the cabinets.

The CCS platform runs an ARINC 653 partitioned operating environment with an Avionics Full Duplex Switched Ethernet (AFDX) network backbone. The CCS provides shared system platform resources to host airplane functional systems such as avionics, environmental control, electrical, mechanical, hydraulic, auxiliary power unit, cabin services, flight controls, health management, fuel, payloads, and propulsion.

The CCS has a common data network (CDN) which includes network switches inside the CCR cabinets and externally mounted throughout the aircraft. CDN is a fiber optic Ethernet that connects all the systems that need to communicate with the CCS and with each other.

The CCS is designed to utilize remote data concentrators (RDC) to consolidate inputs from aircraft systems and aircraft sensors which include analog and digital interfaces, such as ARINC 429 and controller area network (CAN) bus.

The CCS replaces multiple computers and hosts up to 80 avionics and utility functions. More than 100 different line replaceable units (LRUs) were eliminated.

6.3 Software Architecture of Aerospace Real-Time Systems

6.3.1 Overview

Software for modern aerospace systems has been one of the major cost and schedule drivers for many aircraft and spacecraft programs. Software architecture design is one of the key activities that has a significant impact on the development and maintenance of the aerospace systems, both in terms of technical performance and in terms of cost and schedule.

Modern aerospace system architecture is driven by two types of requirements:

- The functional/performance/usage requirements.
- The operation/development/maintenance quality attributes.

The functional/performance/usage requirements drive the capabilities to be supported by the architecture. The quality attributes drives the selection of different architecture tactics and design patterns to achieve "quality attribute response."

Two of the key tactics that have a major impact on the architecture design are two generic "design patterns": the layered architecture and the component-based architecture. These "design patterns" lead to quality attribute responses that are generally desired such as maintainability, evolvability, and ease of development.

In the subsequent subsections, we provide a detailed discussion about component-based architecture and the layered architecture and discuss the various layers in a typical aerospace real-time control system.

6.3.2 Layered Architecture

Layered architecture is a particular way of managing the dependency and consequently the complexities within the system. Layered Architecture is not restricted to software. The basic idea applies to both systems and software. In a layered architecture, the system/software is divided into groups of components with each group called a "layer."

A strictly layered architecture requires that the layers form a serial dependency stack. In other words, if the layers are numbered from bottom to top to be layer 0, layer 1, . . . , and layer N, then Layer K can only depend on Layer K–1.

Relaxed layered architecture allows the upper layers to depend on all the lower layers. In reality, the architect and developer can define a specific layered architecture with dependency rules between the strictly layered architecture and relaxed layered architecture to provide the best compromise between complexity management and performance and efficiency.

The restricted dependency in a layered architecture means that a change of a layer's interface can only impact the immediate layer above in a strictly layered architecture or layers above for the relaxed layered architecture.

Layers are typically selected according to "separation of concern" principle: upper layers taking care of higher level and more abstract concerns and lower layers taking care of lower level and more specific concerns.

Figure 6.5 shows a typical example of the layered architecture of aerospace real-time system. In this example, the layers are organized to address different level of concerns.

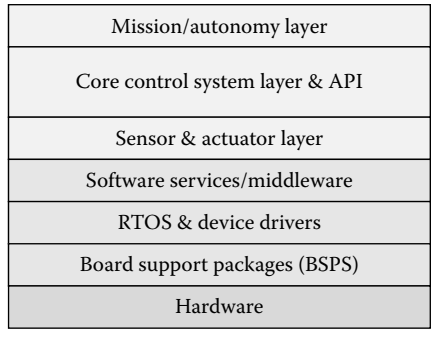

FIGURE 6.5 Layers of typical aerospace real-time software.

The top layer, "mission and system autonomy layer," is concerned with achieving the mission objectives by operating the system using the core control system's abstract capabilities (interfaces). This layer is not concerned with how the specific operations are achieved.

The next layer down is the core control system layer which is concerned with how to provide the abstracted capabilities using control algorithms. To the upper layer this layer is abstracted by its API.

The next layer down is the sensor and actuator layer which handles the specific details of individual sensor and actuator. To the upper layer, the abstract sensor and actuator interfaces are device independent.

The above three layers are called the "application layer." Below the application layer is the "software service/middleware layer," which provides services such as dispatching, communication, and data management.

Below the middleware there exist the real-time operating system and the device driver layer.

The operating system (OS) is shielded from the hardware by the board support package layer.

The lower three layers are sometimes called the "infrastructure layer."

6.3.3 Component-Based Architecture

Software component is a software element which is encapsulated with well-defined services and interfaces, that can be reused, is not specific to the context where the component is used, and can be independently deployed. Typically, a component is constructed using smaller components.

Component-based architecture decomposes the engineered systems into a set of components that interact with each other using a structured approach which is determined by the architecture.

Software components often take the form of objects or collections of objects (such as a C++ or Java Class) in some binary or textual form, adhering to some interface description language [such as IDL used by Common Object Request Broker Architecture (COBRA)] so that the component may exist autonomously from other components in a computer.

Component-based architecture facilitates planned reuse of software (product-line approach). It also allows independent vendors to be in the business of producing components that can be marketed to different high-level integrators.

It is important to recognize that "large grain" components, that is, components that integrate significant portion of total system capability, allows new systems to be developed in a very rapid fashion and leads to significant cost and scheduling savings.

Figure 6.6 shows the high-level static architecture of a typical spacecraft control system which is composed of "large grain" components of attitude determination and control (ADCS), thermal control system (TCS), and EPS. The ADCS is in turn composed of the next level of "large grain" components that are the "orbit and ephemeris subsystem," the "steering subsystem (SteeringSubsys)," the "attitude determination subsystem (ADSubsys)," the "attitude and momentum control subsystem (AMCSubsys)," the "solar wing control subsystem (SWCSubsys)," and the "common gimbal control subsystem (CGCSubsys)."

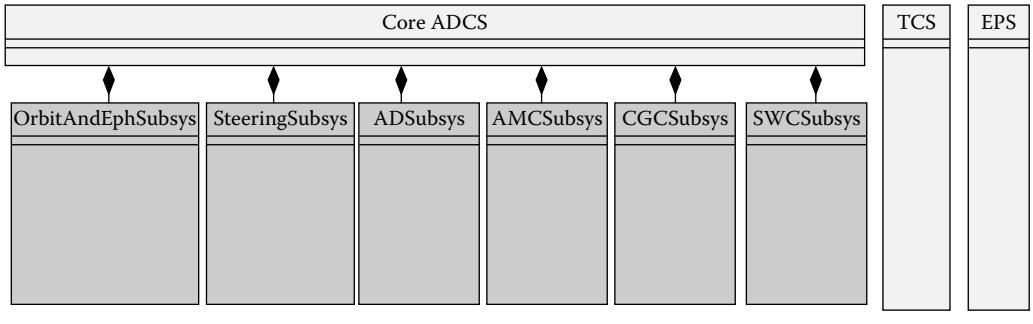

FIGURE 6.6 High-level static architecture of a typical spacecraft control system.

6.3.4 The Infrastructure Layers

The infrastructure layers typically consists of three layers, the middleware layer, the Real Time Operating System (RTOS)+Device Driver Layer and the Board Support Package (BSP) Layer.

The infrastructure software shields the computer hardware and computes the input and output device specifics from the application software (it does not shield, however, the high-level characteristics of sensors and actuators) by providing the application layers with abstract services that do not have to change when the computer hardware, the I/O devices, or the infrastructure software itself change for other reasons.

Typically, the services provided by the infrastructure software are

- Multitasking scheduling/dispatching services
- Timing services
- Communication services
- Data management services.

These services are typically provided through the middleware which augments the generic real-time operating system capabilities to provide services that the application software can use with minimum efforts.

Almost all the real-time operating systems used by real-time control systems comply or conform to the Portable Operating System Interfaces (POSIX) standard. Using a POSIX compliant or conformant RTOS means that the RTOS can be easily exchanged out to be replaced by another vendor's products if needed without causing a problem to the middleware and the application software.

For aerospace control systems, as mentioned before, ARINC 653 defines a standard for partitioned real-time operating system capabilities to the application software. In an ARINC653 compliant system, the throughput (time) and memory are partitioned into independent resources that appear to be an independent virtual computer. Failures/faults that happen to a particular partition do not propagate into other partitions. ARINC653 allows independent application software to run on the same computer. To some extent, these applications can be independently qualified and certified. Figure 6.7 illustrates this concept.

6.3.5 Application Layers

The application layers implement the functionality of the real-time system on the top of the platform provided by the infrastructure layers.

Typical aerospace application software has at least three layers: The sensor and actuator layer, the core algorithm layer, and the operation and autonomy layer.

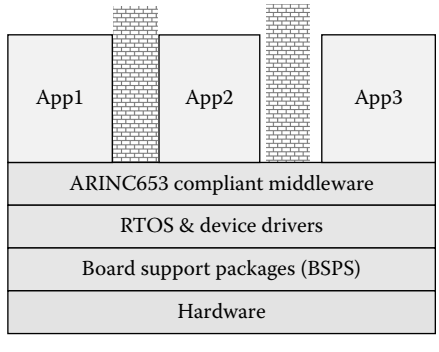

FIGURE 6.7 Concept of independent applications running on the same hardware through ARINC653 compliant partitioned real-time operating system.

The sensor and actuator abstraction layer on the bottom provides an abstraction of the sensors and actuators. Although the sensor and actuator layer itself depends on specifics of the sensors and actuators, it is typically a design goal that the abstraction provided by this layer is independent of the specifics of the devices and consequently the change of sensors and actuators does not impact on the layers above.

The core algorithm layer implements the detailed algorithms for the aerospace system. It is typically a design goal to make this layer independent of the hardware devices and implementation focus on the abstract algorithms.

The operation and autonomy layer provides the operational and autonomy capabilities. These capabilities are concerned with operating the system using the capabilities provided by the core algorithm layer.

It is a good practice for each of the layers to be built using components.

6.3.6 Example: Aircraft FCS Application Layer Software Architecture

As we discussed before, the flight computer and the infrastructure software form what we call the "platform" which is very similar from system to system. It is the application software and the sensor and actuator hardware that makes the real-time aerospace system unique. Consequently, in this example as well as other examples that follow, we pay much more attention to the architecture of the application layer software.

It is also worthwhile to mention that there are many ways to describe and view the architecture of software. In this chapter, we use diagrams with which control system engineers are more familiar with.

Figure 6.8 shows a high-level architecture of fictitious flight control software.

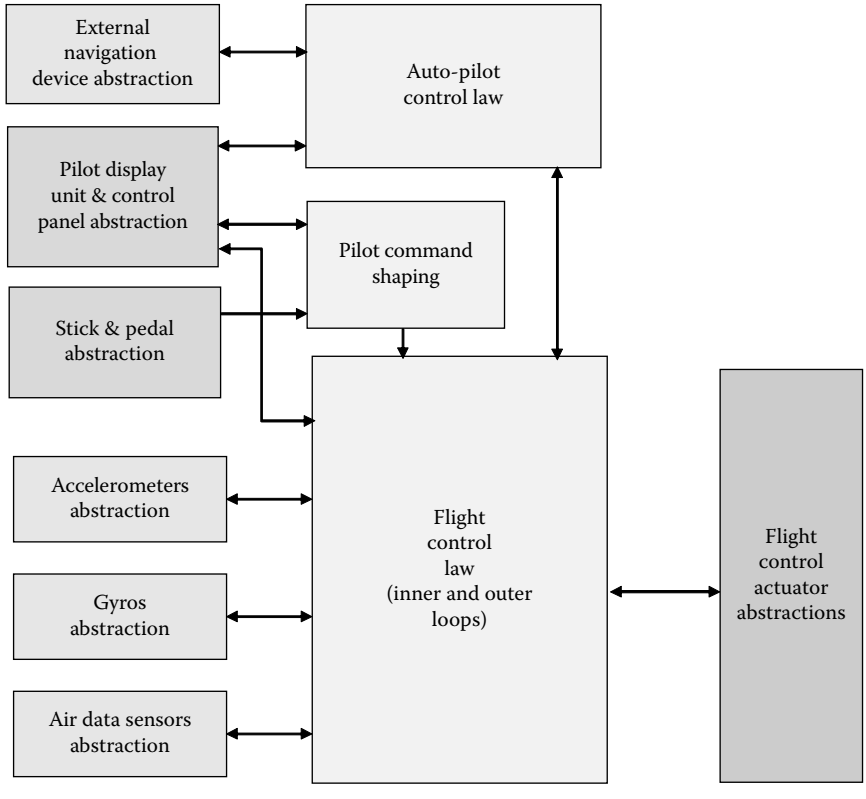

FIGURE 6.8 Fictitious flight control application Software Architecture.

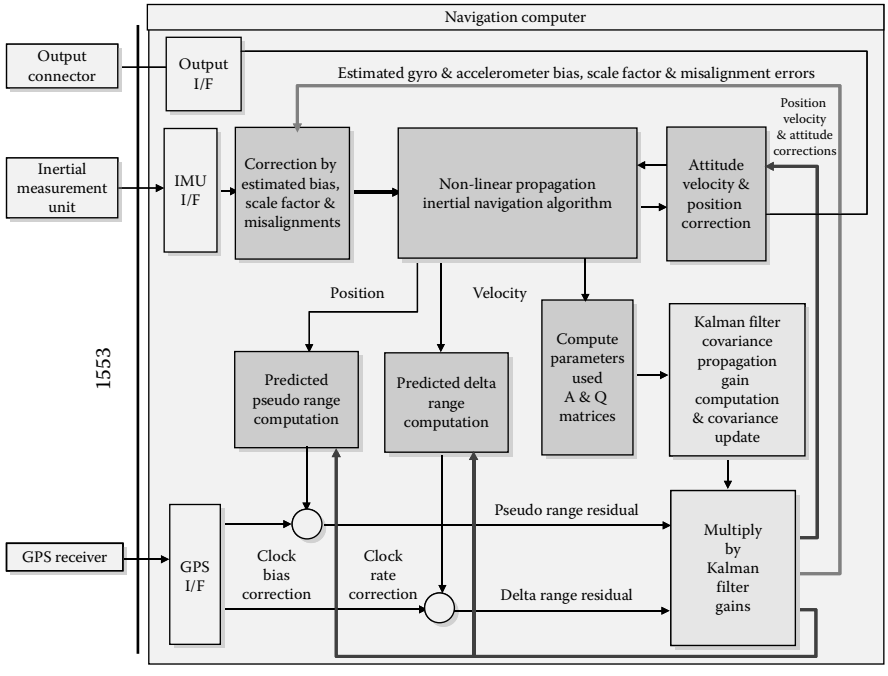

FIGURE 6.9 Typical architecture of integrated GPS/INS system.

6.3.7 Example: GPS-Aided Aircraft Navigation System Application Layer System Architecture

GPS/INS (Inertial Navigation System) is not a complete control system but rather it is a system that provides the capability of an "estimator" in an "estimator/controller" system architecture. Consequently, the GPS/INS system has sensors but does not have actuators.

In this particular example, the Inertial Measurement Unit (IMU), GPS receiver, and the output connector, all interface with the flight computer through MIL-STD-1553 serial data bus. This approach simplifies the hardware architecture.

The functional block diagram presented in Figure 6.9 shows the capabilities and the mechanism of GPS/INS application flight software which in this case is hosted on the dedicated navigation computer hardware and low-level infrastructure software. The same application software can be hosted in an IMA-based system.

6.3.8 Example: Spacecraft Control System Application Software Architecture

Figure 6.10 shows an example of "large grain" component-based architecture for a spacecraft control system.

6.4 Real-Time Aerospace System Quality and Development Processes Standard

6.4.1 Overview

There are a number of software process standards that are followed in the aerospace industry. The choice of standard is driven by the market the system is to address.

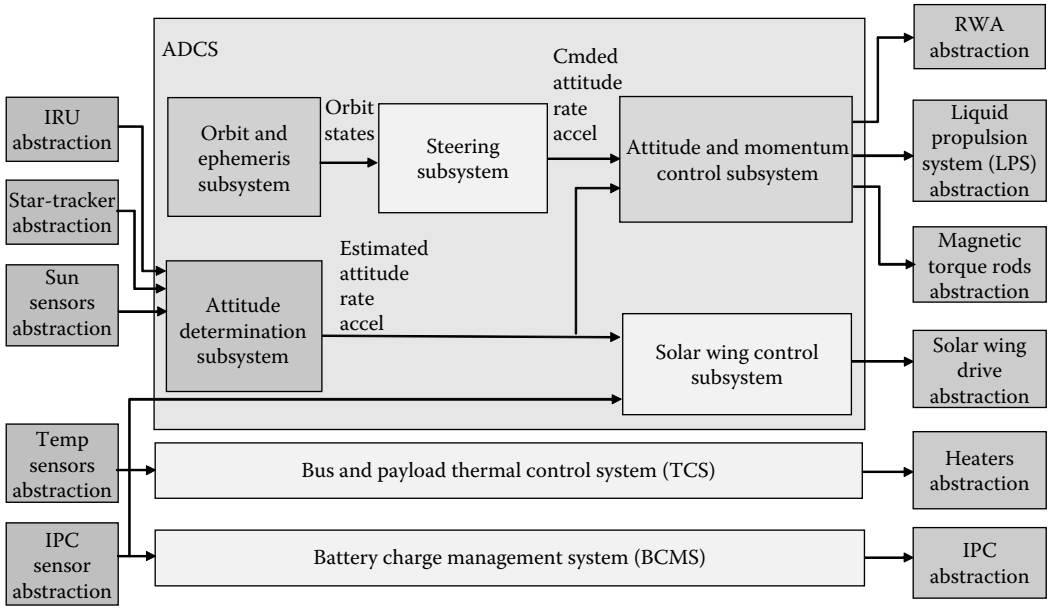

FIGURE 6.10 A typical large grain component-based spacecraft control software.

For military applications, there has been a history of MIL-STD-2167, MIL-STD-498, and IEEE 12207. These processes typically do not provide sufficient flexibility in terms of adopting modern software development methodology and consequently is being used less and less than in the 1980s and 1990s.

For commercial aviation market, due to the need of FAA certification, virtually all aircraft-related software development follow DO-178B standard created by Radio Technical Commission for Aeronautics (RTCA). DO-178 is objective driven and consequently provides flexibility for different software development methodologies to be used.

Capability Maturity Model Integration (CMM/CMMI), developed and advocated by Carnegie Mellon University's Software Engineering Institute (SEI), is widely used across various sectors of industry. CMMI is not only applicable to software engineering, but applies also to system engineering.

It is important to mention that, due to large cost and schedule overruns, or even complete failures of a large number of high-profile aerospace software development efforts, the industry is moving more and more toward a risk-driven, iterative development approach rather than the traditional waterfall approach. IBM's RUP and the open source version of a similar process called unified process (UP) and Open Unified Process (OpenUP) are among the popular ones that are adopted in the industry.

The iterative development processes such as RUP are allowable under DO-178B as well as under CMMI. It will be difficult, although not impossible, to adopt an iterative approach for MIL-STD-2167 type of processes.

6.4.2 DO-178B Standard

DO-178B, Software Considerations in Airborne Systems and Equipment Certification provides guidance for software development. DO-178B is published by RTCA, and was jointly developed by RTCA and the European Organization for Civil Aviation Equipment (EUROCAE).

DO178-B has become a *de facto* standard and the FAA's Advisory Circular AC20-115B established DO178-B as the accepted means of certifying all new aviation software.

TABLE 6.1 Summary of the Definitions of the Design Assurance Levels (DALs)

DAL	Failure Consequence	Description of Failure Consequence
A	Catastrophic	May cause a crash
B	Hazardous	Large impact on performance/safety/crew operation; serious/fatal injuries among passengers
C	Major	Major impact on performance/safety/crew operation; leads to passenger discomfort
D	Minor	Noticeable impact. Leads to passenger inconvenience
E	No effect	No impact on performance and safety

DO178-B is primarily concerned with the development processes. As a result, certification to DO178-B requires delivery of multiple supporting documents and records. The quantity of items needed for DO178-B certification, and the amount of information that they must contain, is determined by the level of certification.

The Design Assurance Levels (DALs) are used to classify the criticality of the aerospace system software and different level of test and validation objectives with different levels of independence are defined for each of the levels.

DO-178B is objective driven and consequently there is a lot of flexibility in terms of the software development life-cycle process. Both the traditional waterfall process and the more modern iterative process can be allowed under DO-178. It does, however, provides the following key activities of the development effort in terms of the objectives, what has been done and what documents are produced:

- Planning
- Development
- Verification
- Configuration management
- Quality assurance
- Certification liaison.

The following is a list of DO-178B required documents and records. Not all are required at all certification levels.

DO-178B Documents related to planning:

- Plan for Software Aspects of Certification (PSAC)
- Software Development Plan (SDP)
- Software Verification Plan (SVP)
- Software Configuration Management Plan (SCMP)
- Software Quality Assurance Plan (SQAP)
- Software Requirements Standards (SRS)
- Software Design Standards (SDS)
- Software Code Standards (SCS).

DO-178B Documents related to development:

- Software Requirements Data (SRD)
- Software Design Description (SDD)
- The actual software code and images.

DO-178B Documents/records related to verification:

- Software Verification Cases and Procedures (SVCP)

- Software Configuration Item (SCI)
- Software Accomplishment Summary (SAS)
- Software Verification Results (SVR)
- Problem Reports.

DO-178B Documents/records related to configuration management:

- Software Configuration Item (SCI)
- Software Life-Cycle Environment Configuration Item (SECI)
- Software Configuration Management Records
- Software Quality Assurance Records (SQAR).

DO-178 Documents/records related to quality assurance:

- Software Quality Assurance Records (SQAR)
- Software Conformity Review (SCR)
- Software Accomplishment Summary (SAS).

6.4.3 Mil-STD-2167/Mil-STD-498/IEEE 12207 Standards

Military-Standard-498 (MIL-STD-498) was a U.S. military standard with the purpose of software development and documentation. It was released in 1994, in replacement of DOD-STD-2167A, and a few other related standards. Mil-STD-498 was intended to be interim standard and was canceled in 1998 and then replaced by J-STD-016 /IEEE 12207. However, these standards were developed by the same key technical personnel and follow the same philosophy, and the "essential contents" are all the same.

These standards provide a rigorous guideline as to how software shall be developed with quality ensured. The process included guidelines on requirement analysis, architecture and detailed design, code, unit test, integration test, qualification test, configuration management, and maintenance. These standards specify the following documents to be produced:

- Software Development Plan (SDP)
- Software Test Plan (STP)
- Software Installation Plan (SIP)
- Software Transition Plan (STrP)
- Operational Concept Description (OCD)
- System/Subsystem Specification (SSS)
- Software Requirements Specification (SRS)
- Interface Requirements Specification (IRS)
- System/Subsystem Design Description (SSDD)
- Software Design Description (SDD)
- Interface Design Description (IDD)
- Database Design Description (DBDD)
- Software Test Description (STD)
- Software Test Report (STR)
- Software Product Specification (SPS)
- Software Version Description (SVD)
- Software User Manual (SUM)
- Software Input/Output Manual (SIOM)
- Software Center Operator Manual (SCOM)
- Computer Operation Manual (COM)
- Computer Programming Manual (CPM)
- Firmware Support Manual (FSM)

6.4.4 Capability Maturity Model Integration

Capability Maturity Model Integration (CMMI) is a process improvement approach that provides companies and organizations with the essential elements of effective processes. CMMI is the successor of the Software CMM. The software CMM was developed from 1987 through 1997. CMMI Version 1.1 was released in 2002 followed by version 1.2 in 2006. The goal of the CMMI project is to improve the usability of maturity models by integrating many different models into one framework. It was created by members of industry, government and the SEI. The main sponsors included the Office of the Secretary of Defense (OSD) and the National Defense Industrial Association.

CMMI's best practices are published in documents called models. A process model is a structured collection of practices that describe the characteristics of effective processes. The practices are those proven by experience to be effective. A process model is used to help to ensure stable, capable and mature processes and is used as a guide to improve the processes.

CMMI model is not a process but rather describes the characteristics of the processes. Consequently, very different processes and system and software development methodologies can be consistent with practices identified by CMMI. For example, both waterfall development processes and iterative development processes can be consistent with CMMI practices.

There are currently two areas of interest covered by CMMI models: development and acquisition. Only "development" is discussed in this chapter.

CMMI v1.2 (CMMI-DEV) model provides best practices in the following 22 process areas:

- Causal Analysis and Resolution
- Configuration Management
- Decision Analysis and Resolution
- Integrated Project Management
- Measurement and Analysis
- Organizational Innovation and Deployment
- Organizational Process Definition
- Organizational Process Focus
- Organizational Process Performance
- Organizational Training
- Project Monitoring and Control
- Project Planning
- Process and Product Quality Assurance
- Product Integration
- Quantitative Project Management
- Requirements Management
- Requirements Development
- Risk Management
- Supplier Agreement Management
- Technical Solution
- Validation
- Verification

The maturity of a company/organization's process and practices can be appraised and rated at one of the following five levels:

- Level 1: Initial
- Level 2: Repeatable
- Level 3: Defined
- Level 4: Managed
- Level 5: Optimized

6.4.5 Waterfall versus Iterative Development Process

It is important to note that for system and software development, there are two very different ways of development. The first way is often referred to as "waterfall process" which features the sequential execution of system/software development tasks. A car-product-line is an example of waterfall process. In a waterfall process, a working product is produced only at the end of the process. The second way is often referred to as "iterative process" which features "growing" or "maturing" the product through iterations. At each iteration, there is a working product which has partial or immature capabilities. A good example of iterative development will be the growth of life.

Waterfall process is very efficient when the requirement and design is mature and it is simply a matter of implementation or mass production. The disadvantage of waterfall process is that it does not tolerate changes. A change in the requirement or design can cause major cost and schedule issues in winter-fall processes. Consequently, waterfall process will be a good approach for mass production of a well-designed product. The waterfall approach, however, may not be a good process to mature the design of a product.

Iterative development, on the other hand, grows the product from a very immature prototype to a fully functional and fully qualified product through iterations. In each iteration, one will do a bit of everything including requirement analysis, architecture design, detailed design, implementation, integration, and test. Iterative development is appropriate when the requirement and design is highly likely to change during the development. Iterative development is efficient if the product developed is not for mass-production. (Note that software is never for mass-production, of course we ignored the production in the trivial case of making copies or distribute through a Web site.)

One of the key benefits of iterative development effort is that it exposes and reduces risks early and often and consequently in most cases, iterative development process is better than the traditional waterfall process for software development.

Among many variations of iterative development process, the IBM RUP represents one of the most successful processes using the iterative development approach.

Figure 6.11 illustrates the RUP.

As illustrated in the figure, the life cycle is divided into four sequential phases (waterfall in nature) and each phase consists of a number of iterations (iterative in nature). For each iteration, one

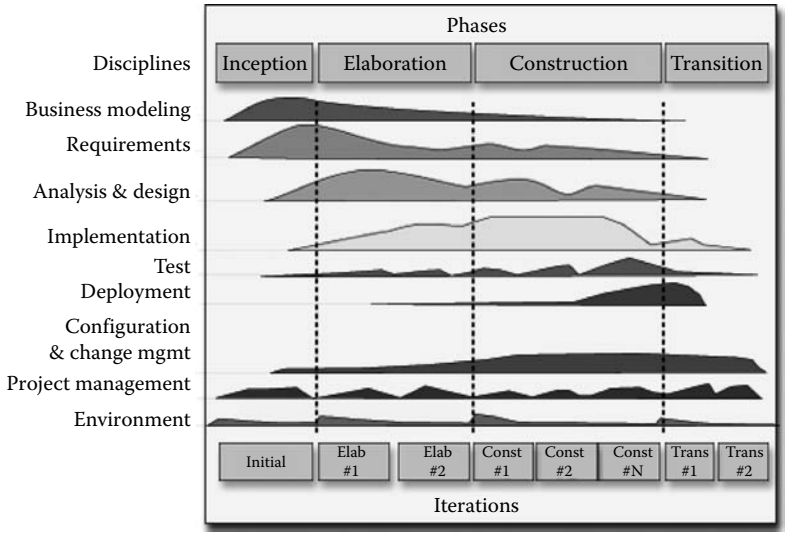

FIGURE 6.11 The rational unified process.

does a little bit of business modeling, requirements, analysis and design, implementation, test, and deployment.

The purpose of inception is to reduce the business risk. The purpose of elaboration is to reduce the architecture risk. The purpose of the construction phase is to construct the product and reduce the implementation risk, and the transition phase is to reduce the deployment risk.

6.5 Simulation, Test, and Verification/Validation Approaches

6.5.1 Overview

For a real-time control system development, simulation plays a very important role in the life cycle of control system development. The simulation is used to provide validation of the system concept prior to the development efforts. It can also be used to provide the sizing, cost, and schedule estimate during the "inception phase" of the efforts.

During the (traditional) preliminary design and critical design phases, (similar to elaboration phase for RUP), the simulation is then used to verify (1) that the requirement of the system is consistent with the "use cases" or "usage scenarios," (2) the architecture of the system supports all the required capabilities, and (3) the detailed design works and meets the requirements.

During implementation phase, or the "construction phase" in RUP-like process, the "software-in-the-loop" (SIL) simulation can be used as a software development platform to integrate and test the implemented system/software. SIL simulation allows most of the application software bugs to be detected and removed early and easily in a desktop environment.

Beyond the fidelity provided by "SIL" simulation, a "processor-in-the-loop" (PIL) simulation will run the "truth models" on a general purpose computer and the flight software on a target processor. PIL simulation is typically performed in real-time and consequently allows the risks associated with processor and real-time execution retired early in an easy and convenient environment.

The highest fidelity simulation is provided by the "hardware-in-the-loop" (HIL) simulation environment where a test version of the flight computer is integrated with I/O hardware driven by real or simulated hardware devices. HIL provides the realistic environment for the flight software to be tested. "Human-in-the-loop" simulation is a special type of HIL simulation where the human interface hardware devices are included in the simulation and consequently allow one to interact with the system.

With the exception of the simulation for the concept development, it is usually more productive if the commonality of various simulations mentioned above, including the simulations for SIL test, simulation for PIL test, and simulation for HIL test, are maximized and are natural extension of lower-fidelity simulation.

6.5.2 Simulation for Concept Development and Verification & Validation (V&V)

During concept development of a design of an aerospace control system, simulation plays an important role of verifying that the concept is feasible and that the objectives of the system can be achieved. The simulation also helps to develop an early estimate of the scope and size of the effort and as well as the development of an estimate of the cost and schedule.

Simulation for concept development typically requires that the simulation be simple and fast and could be easily tailored to analyze different scenarios in a short time. Consequently, simulations for this purpose are typically coded in languages such as MATLAB and Simulink or in the form of a spreadsheet. It will be too much overhead at this phase to try to use a simulation that has much higher fidelity with the exception that the risks for the projects are identified to be "implementation, cost, and schedule "risks instead of "technology and design" risks.

If the risks are on the "implementation side," early versions of higher-fidelity simulations such as SIL, PIL, and even HIL may be necessary.

6.5.3 Flight Software in the Loop Simulation and V&V

Once a detailed design is developed, it becomes necessary to develop and maintain a high-fidelity simulation for the purposes of design validation, performance analysis, and implementation of risk reduction.

Although it is a trade, most of the time, it is productive for the high-fidelity simulation to be a "flight software in the loop simulation," that is, SIL simulation. SIL simulation has the advantage of simultaneously validating system architecture and system performance, the algorithm design, and the software implementation.

Modern SIL simulations are typically available on the developer's desktop. SIL simulations allow the designers, analysts, and implementers to quickly implement and validate design changes in an efficient manner and consequently are very important factors for efficient software and system development processes. This is especially true for iterative development process where early and often test and fast turn-around are essential for efficient execution of iterations.

Typical SILs run faster than real-time to allow efficient testing and debugging. SIL can also be configured to run real-time when risks associated with real-time process and multitasking are to be reduced early in the development.

6.5.4 Processor in the Loop Simulation and V&V

Processor in the loop simulation (PIL) provides the next level of fidelity of simulation for the real-time aerospace systems. Some PIL platforms just have the target processor that runs the flight software and the simulation truth models that run on a general purpose processor and they communicate with one another through standard interfaces such as Ethernet and Transmission Control Protocol and the Internet Protocol (TCP/IP). Some PILs are of high fidelity in such a way that all the hardware interfaces appear to be real to the processor and in this case, hardware-related low-level software functions can be integrated and tested in this environment.

Typically, PIL platforms are more expensive, less available, and take more effort to run test cases. Consequently, PILs should be used less frequently than SILs. In fact, most of the debugging and testing should take place in the SIL environment.

A PIL allows risks associated with the specific target processor to be reduced by testing and debugging in the PIL environment. With the advancement of modern infrastructure software, such a risk rarely happens to the application software. A PIL is very useful for infrastructure software risks reduction.

6.5.5 Hardware-in-the-Loop V&V

Hardware in the loop simulations provide the highest fidelity simulation where engineering development units will be connected together to conduct integration test.

An HIL platform typically only runs real-time and it uses the engineering development model of flight computers, sensors, and actuators to perform tests where hardware interfaces are important.

In an HIL platform, depending on what interfaces or integration issues are considered as risks, not all hardware will be included in the simulation, only those devices deemed risky for the successful development of system are included.

Clearly, an HIL has the most fidelity and it is also the most expensive and the least available. Consequently, an HIL is only used to perform test and validation that cannot be done elsewhere such as hardware/software interface and interactions with sensors and actuators as well as tests that are designed to test the entire system such as test cases used as part of formal qualification test (FQT).

6.6 Integrated System and Software Engineering and Model-Driven and Model-Based Development

Unlike software development in other industries, for an aerospace industry, it has always been separated into two sets of activities. The first set includes definition and validation of the algorithms, the development of a simulation that validates the algorithm design and system performance and the development of an algorithm design document to communicate the design to engineers who will be writing the code. The second set of activities includes software coding of the algorithms testing, integration, and qualification of the code.

This traditional approach has suffered for many years from a fundamental flaw: the domain knowledge is separate from the code, test, and integration activities. Consequently, the second sets of activities become very inefficient and are often the area of risk for the projects.

Indeed, software cost and schedule are identified as key risks in almost all aerospace real-time system development projects. While there are many reasons for this fact, the separation of domain expertise from software coding, debugging, and testing is one of the key problems in the traditional aerospace software development process.

This problem is becoming more and more apparent due to the following reasons:

- The increase of the complexity of the algorithms has made it necessary for the architecture of these algorithms and the software to be done by individuals who have the domain knowledge. Combined control system domain expertise and system and software architecture expertise have proven to be critical for efficient and successful development of complex, performance demanding, safety-critical aerospace systems and software.
- The increase of the complexity has made it necessary for the person who writes and debugs the software that implements the algorithm to have sufficient domain knowledge to be efficient.
- The modern iterative development approach (such as IBM's RUP, which has proven to be a superior approach to the traditional waterfall process for software development, demands the capability for fast iterations, which consists of tightly coupled requirement, architecture, detailed design, implementation, integration, and test activities. The traditional "system-engineering-throw-over-wall-to-software-engineering" approach simply does not fit the iterative development approach.

Integrated Algorithm, Simulation, and Software Development (IASSD) is an approach that integrates the two sets of activities mentioned earlier into a single set of activities to be performed by an integrated team of engineers who develop the algorithms, simulation, and flight software at the same time. IASSD is typically performed in an iterative development process such as RUP.

There are a number of approaches IASSD can be practiced:

IASSD can be done by integrating the domain/software team developing the algorithms, simulation and flight software using the traditional "hand-code" approach. For these purposes, domain experts are trained with software development skills and software engineers are trained with domain knowledge.

IASSD can be done using a technique called autocode. When autocode is used, the algorithms, simulation, and flight software are defined and implemented in modern graphical languages such as Simulink and Matrix-X, which are commonly used by control engineers. The flight code is then automatically generated from the higher-level graphical description of the design. This approach is often referred to as "model based design."

Autocode can also be done from an UML-based architecture model and this approach is often associated with "model-driven design."

Regardless of the approach taken, a critical factor in successful IASSD-based development is the development and the use of "flight software in the loop" simulations throughout the life cycle. For the

hand-coded-based development approach, the hand-coded flight software is included in the simulation. For the model-based approach, the flight software in the loop simulation is really just the graphical model-based simulation itself.

6.7 Software Reuse and Software Product Lines for Aerospace Systems

As mentioned before, cost and schedule problems for aerospace real-time software systems are persistent across the industry. Better process, better approach, better tools, and better designs provide better productivity and consequently reduce the cost and schedule.

It is important to realize no productivity improvement saves more cost and schedule than eliminating certain development completely. Software Reuse and Software Product Line are ways to reduce cost and schedule along this line.

It is also important to note that reuse means much more than just the reuse of code. The concept, the requirement, the architecture, the architecture model, the design document, the code, the test cases, and so on can all be reused.

Software reuse is a broader term than SPL. Software reuse includes any form of reuse. Product-line approach, however, only refers to "planned" and "managed reuse" in a "prescribed way." Because of the planning and development of reusable assets in a product-line approach, the SPL approach often achieves significantly better results than traditional "plain" reuse.

Over the years, software reuse has evolved from the reuse of functions or subroutines, to the reuse of modules/classes/objects that contain data and functions, to components that provide significant amount of capabilities for a software system.

In terms of the size of the reused elements, it has grown from the "small grain functions/objects" to "large grain components." As we discussed in the architecture section, "large grain components" allows rapid assembly, integration, and test of software system using just a few large components.

An SPL approach goes further from the advancement of the reuse strategy into fully strategic reuse. That means developing many software system products with some common managed set of features from a common set of core assets in a prescribed way.

The establishment of an SPL consequently is more complicated than the establishment of a code-library.

By carefully studying successful and unsuccessful examples of SPL practices in the industry, SEI of CMU has developed a fairly mature framework for software product line practices. The current version of SPL Frame Work is FrameWork 5.0.

This framework identifies three essential activities including "core asset development," "product development," and "management."

This frame work identifies and describes 29 practice areas (PAs) that fall in three categories: "software engineering," "technical management," and "organizational management."

The software engineering PAs are

1. Architecture definition
2. Architecture evaluation
3. Component development
4. Mining existing assets
5. Requirement engineering
6. Software system integration
7. Testing
8. Understanding relevant domains
9. Using externally available software.

The technical management PAs are

10. Configuration management
11. Make/buy/mine/commission analysis
12. Measurement and tracking
13. Process discipline
14. Scoping
15. Technical planning
16. Technical risk management
17. Tool support.

The organizational management PAs are

18. Building a business case
19. Customer interface management
20. Developing an acquisition strategy
21. Funding
22. Launching and institutionalizing
23. Market analysis
24. Operations
25. Organizational planning
26. Organizational risk management
27. Structuring the organization
28. Technology forecasting
29. Training.

For aerospace systems, there are a lot of similarities and common capabilities for different systems used by aircraft, missile, launch vehicles, and spacecraft. Capabilities such as navigation, attitude determination, orbit determination, and thermal control are very similar across many different types of vehicles. For flight control, similarity exists among smaller classes of vehicles. In addition, virtually for all real-time aerospace systems, the real-time operating system can be very similar. These similarities allow reuse at various levels. By careful planning, SPLs can be established to go beyond traditionally simple reuse into strategic reuse through software product-line.

Successful deployment of product-line can lead to large-scale productivity gains; it improves time to market; it helps to maintain a market presence; it helps to sustain growth; it reduces demand for manpower; and it improves product quality and improves customer satisfaction.

Some of the key factors that contribute to successful development of SPL are

1. A compelling business case
2. Deep domain knowledge
3. A rich legacy base
4. A dedicated champion
5. Organizational cohesion
6. Courage to try new engineering approaches.

6.8 Conclusions

The advancement of technology is both enabling and demanding a much more sophisticated aerospace control systems with superior capabilities and is developed with a much reduced cost, shorter schedule, and better quality.

This trend requires the aerospace control system engineers to have integrated system engineering, system architecture, system analysis, simulation engineering, software engineering, software architecture, and system software and integration and test.

The authors hope that this chapter may be of some help to adapt the aerospace space control system engineering community toward the future.

References

1. Sharp, D., Reducing avionics software cost through component based product line development, Software Technology Conference, Salt Lake City, UT, April 1998.
2. Batory, D., Lou Coglianese, L., Mark Goodwin, M., and Shafer, S., Creating reference architectures: An example from Avionics. ACM SIGSOFT Symposium on Software Reusability, Seattle, WA, 1995.
3. Clements, P. and Bergey, J., The U.S. Army's Common Avionics Architecture System (CAAS) Product Line: A case study, Carnegie Mellon University, Software Engineering Institute, CMU/SEI-2005-TR-019, Pittsburgh, September 2005.

7

Stochastic Decision Making and Aerial Surveillance Control Strategies for Teams of Unmanned Aerial Vehicles

Raymond W. Holsapple
Air Force Research Laboratory

John J. Baker
University of Michigan

Amir J. Matlock
University of Michigan

7.1 Introduction

In many modern military operations unmanned aerial vehicles (UAVs) are relied upon for many tasks. Some are used for launching weapons at enemy forces, and some may even be weapons themselves. In addition, many UAVs are used for gathering information that may be used to make decisions about current or future missions. Even though there is no doubt that force, might, and firepower are important for military success, few would argue that the truly critical aspects are intelligence, communication, and decision making. Missions that focus on these latter purposes are referred to as Intelligence, Surveillance, and Reconnaissance (ISR) missions.

There are many challenges faced when ISR missions are performed by UAVs, and hence there is a tremendous amount of research effort that is being put into studying algorithms and technology that will increase the effectiveness of UAV ISR missions. The trend is that uses for UAVs in military operations is growing at an amazing rate. At some point in the future, we might even be able to say that most military operations (air, land, and sea) will be performed by unmanned systems that have varying levels of autonomy.

Some of the key topics in this area of research have been task assignment and path planning. A significant amount of effort and money has been put into research that focuses on these topics. As such there are volumes of literature on the subjects. Not all of the research has its focus on military issues, as UAVs provide a solution to a wide variety of both military and civilian applications. Aerial surveillance, for example, is a key technology for both civilian and military applications. Civilian uses include forest fire monitoring, wildlife tracking, oil spill detection, traffic monitoring, and search/rescue missions. Military applications are many and varied. They include both strategic and tactical uses; a few examples include target detection, target classification, target tracking, battle damage assessment, perimeter monitoring, area surveillance, and intelligence gathering.

The decision algorithms discussed in this chapter were designed for an Air Force research program known as COUNTER [1,2]. COUNTER is an acronym for Cooperative Operations in UrbaN TERrain. The main objective of COUNTER is to use a team of UAVs to investigate task assignment and path planning algorithms for use in ISR missions in urban areas. The chore of gathering information to make decisions on assignment and path planning becomes increasingly difficult in urban areas. Some of the more obvious challenges are the introduction of increased clutter, which can significantly increase false alarm rates and missed detection rates.

COUNTER uses a team of UAVs, one small (unmanned) aerial vehicle (SAV) and four micro (unmanned) aerial vehicles (MAVs). The exact sizes of the platforms need not fit any standard industry definition for these classes of UAVs as COUNTER focuses on algorithm development that is platform independent. The UAVs that satisfy the role of a MAV are not micro by any standard. The SAV loiters over the urban area at 1000–2000 ft. above ground level (AGL), while an operator views the live video feed from the SAV for objects of interest. For the algorithms discussed in this chapter, we assume that the objects of interest remain stationary. After an operator selects a collection of objects to view more closely, a task assignment algorithm assigns a tour to each MAV that is to be launched. The task assignment algorithm that performs this task is not the focus of this chapter, but is described in great detail in [1–6]. The MAVs fly at a much lower altitude (50–150 ft. AGL) allowing them to inspect the objects of interest close-up and at an acute angle, which may permit them to see into vehicles and under tarpaulins and camouflage nets. Like the SAV, each MAV is equipped with front and side facing video cameras. This video feed is relayed back to a ground control station where an operator attempts to classify the objects in real time as the MAVs inspect the collection of objects assigned to them.

Generally speaking, the operator is not asked to give a response whether or not a particular object of interest is a target or a nontarget based on his or her inspection of the video. Instead, the operator is asked whether or not he or she has seen a distinguishing feature that has been described to him or her prior to the mission. The operator may even have a sample picture of such a feature to refer to during the mission. The assumption about this feature is that it uniquely separates targets from non-targets. This assumption will be necessary and will become more obvious when we consider the stochastic controller of Section 7.2.

The inclusion of pop-up alarms as a stochastic event makes the cooperative planning problem more complex and also contributes to the unpredictability of vehicle routes. Due to the extreme complexity of the resulting optimization problem and the requirement to compute new routes quickly when pop-up alarms occur, computing an exact solution to the optimization problem is not feasible. In this chapter, we consider an aerial surveillance problem in Section 7.3. A heuristic approach will be used to compute a suitable and acceptable suboptimal solution within the allowable computation time.

The literature in UAV cooperative control is vast and suggests many ways to control a team of UAVs. Past research for aerial surveillance can be grouped into at least two categories. One approach looks

at the surveillance problem as a coverage problem. In [7], an exhaustive search algorithm similar to a lawn mower pattern is developed to search for targets. Ahmadzadeh et al. [8] consider the problem of optimizing the coverage of an area while satisfying hard constraints, such as initial and final positions. DeLima [9] considers optimizing coverage while using metrics such as dynamic coverage, heterogeneity of coverage, and energy consumption. Others [10–14] have also investigated similar techniques to optimize coverage. Another aspect of aerial surveillance focuses on control algorithms that observe areas of higher interest in a region. Girard et al. [15] and Beard et al. [16] developed control algorithms to track the perimeter of a known area of interest.

To be specific, this chapter offers strategies to solve UAV ISR problems for two main thrusts. The first problem revolves around the task of a team of UAVs using video to closely inspect potential targets. The key aspect of this problem is making decisions about whether or not to make a revisit of an object in order to gather more visual information about the object. The main problem associated with the task is the uncertainty about the benefit of that revisit considering the need to save fuel for future revisits. The second problem is one of aerial surveillance of a military base. We assume the base is divided into regions of varying levels of priority. The task is to determine appropriate vehicle routes when these regions have a dynamic priority (reward) function.

7.2 Stochastic Decision Making with Uncertainty

It is very likely that a human operator would be overwhelmed if he or she was expected to manage MAV task assignments while simultaneously attempting to detect potential target features in multiple live-video streams. Therefore, a controller was developed to assist the operator in making task assignment decisions [17]. The key feature of this approach is the inclusion of an operator error model. Stochastic dynamic programming is used to solve this problem. This type of problem is known as decision making with uncertainty. The state of our dynamic program is the amount of fuel that is allocated for revisiting objects of interest, hereafter referred to as reserve.

In addition to the reserve, the stochastic controller also makes use of the operator's response, the amount of time the operator took to make the response (operator delay), and the number of remaining objects in the MAV's tour. *A priori* probabilities are used to compute the expected value of the reward function, which is the cornerstone of the dynamic program. Some of these probabilities are based on experimental results and some are problem design choices. These probabilities characterize the true-target density, the true-target feature visibility, and the operator's decision behavior.

Revisits of potential targets are often useful because they can provide additional visual information regarding the object from a different heading. For example, a feature may only be visible from the rear aspect of the object, so if the MAV approaches from the front aspect only the operator will never see the distinguishing feature. In this case there may (or may not) be sufficient information gain to perform a revisit, depending on the values of the inputs to the controller. In the case of extremely low target density, there might be sufficient information gain to perform a revisit even in cases where the operator says that he or she has seen the distinguishing feature on the initial inspection. This is due to two things: (1) the uncertainty in the operator's responses and (2) the inherent inclination of the dynamic program to desire target verification in areas of extremely low target density.

Here we note that the reserve set aside for revisits is finite. As previously mentioned, this reserve is the state of the dynamic program that is used to solve this optimization problem. The solution of the dynamic program generates a matrix of cost thresholds that will be used in the decision process. At the moment an operator gives a response (which coincides with the moment of decision by the controller), the expected cost of a revisit is compared to the cost threshold, which was computed during the solution of the stochastic dynamic program. At this point the control decision is simply a table look up. If the expected cost is less than the cost threshold and the expected cost is feasible, then the MAV will revisit the object of interest.

In previous publications [17–21] a probabilistic method was developed to determine the reward values that are used in the cost function of the dynamic program. The probabilities were developed with the assumption that the MAV had Automatic Target Recognition (ATR). ATR is an autonomous system that detects desirable features in images; this is becoming an area of substantial research interest in the fields of computer science and robotics. Using ATR, the MAV would only defer to the operator for classification if the ATR encountered an ambiguous feature. However, in reality optical feature recognition would be problematic in this application due to many reasons: payload constraints on MAVs, potential for poor communication, and highly variable lighting conditions. Additionally, and perhaps most importantly, the MAVs used for COUNTER are not equipped with such a device. This was the motivation for a system that instead of employing an ATR, relies only on a human operator for feature recognition.

In this section we present three different expected reward functions, which will be compared. These functions will rely only upon the operator's response from viewing live video streams captured during the flyover of objects of interest instead of utilizing an ATR. The number of *a posteriori* probabilities needed for the reward functions will be developed. Then an analysis will be performed comparing the reward methods against a benchmark method and each other. Performance of each method will be discussed and a recommendation for future work is made.

7.2.1 Definition of Terms

In this formulation of the dynamic program there are three main events, and they can be treated as boolean operators: feature visibility, operator response, and target truth status (whether or not an object of interest is a true target). These events are considered boolean because only absolutes are considered. For example, the feature is either visible or not visible, the operator indicates feature or no feature, and an object is either a target or it is not. Each MAV visit can be thought of as some combination of these events. The first and second visit of a MAV will be treated as independent events, so the subscripts denoting the visit are only important when probabilities involve events from both visits. The variables describing these events are listed in Table 7.1.

7.2.2 *A Priori* Probabilities

The probability that an object of interest is a true target is assumed *a priori*:

$$P(T_t) = p, \tag{7.1}$$

$$P(T_f) = 1 - p. \tag{7.2}$$

In some scenarios p can be very low; it could be one out of thousands in extremely cluttered urban areas. For the purposes of this chapter, we have chosen values of p in the interval $[0.1, 0.2]$. If we were to let

TABLE 7.1 Notation for Probabilities in Section 7.2

Notation	Description
T	Target truth status
V	Visibility truth status for identifying feature on true-target
R	Response from operator on presence of identifying feature
θ	True-target feature visibility range length
	Subscript
1	Initial flyover of object
2	Second flyover of object
t	Boolean: True
f	Boolean: False

TABLE 7.2 Operator Confusion Matrix

	R_t	R_f
$P\left(R\middle\vert V_t \cap T_t\right)$	P_D	$1 - P_D$
$P\left(R\middle\vert V_t \cap T_f\right)$	Undefined	Undefined
$P\left(R\middle\vert V_f \cap T_t\right)$	P_{FA}	$1 - P_{FA}$
$P\left(R\middle\vert V_f \cap T_f\right)$	P_{FA}	$1 - P_{FA}$

p be extremely small, we would have to load the simulations we conducted with hundreds or thousands of objects of interest each time the simulation was executed. This would lead to an extraordinarily large amount of time spent on running simulations. Moreover, this range of values for p is the *a priori* probability range used in the COUNTER flight tests.

An operator confusion matrix developed in [17] is a way of depicting the imperfect behavior of a human operator given a collection of probabilistic events. In previous efforts [17–21], the probabilistic event corresponding to any given operator response was simply the target truth status. This truth status is of course unknown to the operator but has a probability distribution described by Equations 7.1 and 7.2. The operator confusion matrix developed for this chapter is different from the previous version because it accounts for an additional stochastic event, the feature visibility. This confusion matrix is shown in Table 7.2. Note that a feature cannot be visible on an object of interest that is not an actual target, thus in Table 7.2, $P\left(R\middle\vert V_t \cap T_f\right)$ is undefined by nature. Also in Table 7.2, a design choice was made in which $P(R_t \vert V_f \cap T_t)$ would be the probability of false alarm. The issue is that even if the operator's implication (the object is a target) is technically correct, it is based on no visual evidence and should be treated as a false alarm, which is how it is modeled in this chapter.

Here we should note a couple of things. First, P_D (probability of detection) and P_{FA} (probability of false alarm) are conditional probabilities of the operator's response given the target truth status and the feature visibility status. Second, it is assumed that P_D and P_{FA} are affected by the operator's workload. Nominally we suggest that as an operator's workload increases, the probability of detection should decrease while the probability of false alarm increases.

Now consider the possible feature visibility outcomes from a MAV flying over an object of interest. Each target is modeled as having its distinguishing feature visible only when the target is approached from a heading that falls within an assumed range of length θ. It is further assumed that $0 < \theta < \pi$. The range of visibility divided by the total range of angles the object can be viewed from is the conditional probability that a feature is visible given that it is a target. Table 7.3 lists this set of *a priori* conditional probabilities.

In the case of two visits, the system is modeled such that the MAVs will perform any second visits from the opposite heading as the initial visit. For example, if the initial approach heading is 100° and the stochastic controller determines a second visit should be performed, the heading of the revisit will be 280°. According to the model, if a feature is visible on the first pass it will not be visible on the second pass and vice versa. This is due to the constraint on θ. The conditional probabilities of feature visibility for two visits given the target truth status may now be inferred and are given in Table 7.4.

TABLE 7.3 Visibility Given Target Truth Status

	V_t	V_f
$P\left(V\middle\vert T_t\right)$	$\theta/2\pi$	$1 - \theta/2\pi$
$P\left(V\middle\vert T_f\right)$	0	1

TABLE 7.4 Visibility Given Target Truth Status

	V_{1t}, V_{2t}	V_{1t}, V_{2f}	V_{1f}, V_{2t}	V_{1f}, V_{2f}
$P\left(V_1 \cap V_2 \mid T_t\right)$	0	$\theta/2\pi$	$\theta/2\pi$	$1-\theta/\pi$
$P\left(V_1 \cap V_2 \mid T_f\right)$	0	0	0	1

7.2.3 Reward Multiplier Probabilities

The motivation of the following exercise is to determine the probability of the operator's response and target feature visibility on a second visit, given the operator's response and target feature visibility from the first visit. These probabilities will be used as gains applied against reward values so that they are weighted according to the probability that they will occur. To do this, the probabilities will be broken down into their constituent *a priori* sub probability combinations.

Before describing how that is done, we first note that we seek the following:

$$\begin{aligned} P(R_2 \cap V_2 | R_1 \cap V_1) &= \frac{P((R_2 \cap V_2) \cap (R_1 \cap V_1))}{P(R_1 \cap V_1)} \\ &= \frac{P(R_2 \cap V_2 \cap R_1 \cap V_1 \cap T_t) + P(R_2 \cap V_2 \cap R_1 \cap V_1 \cap T_f)}{P(R_1 \cap V_1 \cap T_t) + P(R_1 \cap V_1 \cap T_f)}. \end{aligned} \tag{7.3}$$

The terms in the denominator can be resolved into their *a priori* constituent parts as follows:

$$\begin{aligned} P(R_1 \cap V_1 \cap T) &= P(R_1 | V_1 \cap T) P(V_1 \cap T) \\ &= P(R_1 | V_1 \cap T) P(V_1 | T) P(T) \qquad (7.4) \\ &\equiv \widetilde{P}(T). \qquad (7.5) \end{aligned}$$

The terms in the numerator may also be resolved into *a priori* constituents. To do this we will begin with a definition.

Definition 7.1:

Two events E_1 and E_2 are conditionally independent of event E_3 if and only if

$$P(E_1 \cap E_2 | E_3) = P(E_1 | E_3) P(E_2 | E_3), \tag{7.6}$$

or equivalently

$$P(E_1 | E_2 \cap E_3) = P(E_1 | E_3). \tag{7.7}$$

■

While breaking the terms in the numerator into their constituent parts, we must assume conditional independence several times. In the equations below, conditional independence is assumed as we proceed from Equation 7.8 to 7.9 and from Equation 7.10 to 7.11. This assumption is intuitive, as it makes sense to assume that the operator's response and feature visibility pairs from the first and second visits should be conditionally independent of each other given the target truth status. The terms in the numerator are

broken down as follows:

$$
\begin{aligned}
P(R_2 \cap V_2 \cap R_1 \cap V_1 \cap T) &= P((R_2 \cap V_2) \cap (R_1 \cap V_1)|T)P(T) && (7.8)\\
&= P(R_2 \cap V_2|T)P(R_1 \cap V_1|T)P(T) && (7.9)\\
&= \frac{P(R_2 \cap V_2 \cap T)}{P(T)} \frac{P(R_1 \cap V_1 \cap T)}{P(T)} P(T)\\
&= \frac{P(R_2 \cap V_2 \cap T)P(R_1 \cap V_1 \cap T)}{P(T)}\\
&= \frac{P(R_2|V_2 \cap T)P(V_2 \cap T)P(R_1|V_1 \cap T)P(V_1 \cap T)}{P(T)}\\
&= P(R_2|V_2 \cap T)P(R_1|V_1 \cap T)\frac{P(V_2|T)P(T)}{P(T)}P(V_1|T)P(T)\\
&= P(R_1|V_1 \cap T)P(R_2|V_2 \cap T)P(V_1|T)P(V_2|T)P(T) && (7.10)\\
&= P(R_1|V_1 \cap T)P(R_2|V_2 \cap T)P(V_1 \cap V_2|T)P(T) && (7.11)\\
&\equiv \widehat{P}(T). && (7.12)
\end{aligned}
$$

Finally, we may use Equations 7.4 and 7.11 to assemble the *a priori* constituent forms of the numerator and denominator in Equation 7.3. For brevity, we will write Equation 7.3 using the equivalent definitions given by Equations 7.5 and 7.12,

$$P(R_2 \cap V_2|R_1 \cap V_1) = \frac{\widehat{P}(T_t) + \widehat{P}(T_f)}{\widetilde{P}(T_t) + \widetilde{P}(T_f)}. \tag{7.13}$$

7.2.4 Reward Probabilities

The two conditional probabilities used to compute the reward values must also be determined. These two probabilities are $P(T|R_1 \cap V_1)$ and $P(T|R_1 \cap V_1 \cap R_2 \cap V_2)$, and they are decomposed into constituents in the equations below. Using Equations 7.4 and 7.5, we have

$$
\begin{aligned}
P(T|R_1 \cap V_1) &= \frac{P(T \cap R_1 \cap V_1)}{P(R_1 \cap V_1)}\\
&= \frac{P(R_1 \cap V_1 \cap T)}{P(R_1 \cap V_1 \cap T_t) + P(R_1 \cap V_1 \cap T_f)}\\
&= \frac{\widetilde{P}(T)}{\widetilde{P}(T_t) + \widetilde{P}(T_f)}. && (7.14)
\end{aligned}
$$

Furthermore, using Equations 7.11 and 7.12, the more complicated of the two conditional probabilities becomes

$$
\begin{aligned}
P(T|R_1 \cap V_1 \cap R_2 \cap V_2) &= \frac{P(T \cap R_1 \cap V_1 \cap R_2 \cap V_2)}{P(R_1 \cap V_1 \cap R_2 \cap V_2)}\\
&= \frac{P(R_2 \cap V_2 \cap R_1 \cap V_1 \cap T)}{P(R_2 \cap V_2 \cap R_1 \cap V_1 \cap T_t) + P(R_2 \cap V_2 \cap R_1 \cap V_1 \cap T_f)}\\
&= \frac{\widehat{P}(T)}{\widehat{P}(T_t) + \widehat{P}(T_f)}. && (7.15)
\end{aligned}
$$

Since each event is represented by a boolean value, the number of equations needed to describe the system is simply 2^n, where n is the number of boolean events involved. For example, $n = 5$ in Equation 7.15, so there are 2^5 equations in the form of Equation 7.15 which are used to compute the reward value.

7.2.5 Reward Functions

Two information theory reward functions from a previous effort [17] and an additional method, where discrete reward values are assigned, will be considered and evaluated. The range of values from the rewards is essentially arbitrary but provides a basis for comparison of possible outcomes. These rewards will then be scaled respectively by Equation 7.13 from Section 7.2.3. For the purpose of brevity while describing the three methods, let $A = R_1 \cap V_1$ and $B = R_2 \cap V_2$.

Method 7.1:

We begin by defining some conditional probabilities:

$$P_{11} = P(T_t|A), \tag{7.16}$$
$$P_{12} = P(T_f|A), \tag{7.17}$$
$$P_{13} = P(T_t|A \cap B), \tag{7.18}$$
$$P_{14} = P(T_f|A \cap B). \tag{7.19}$$

Then the reward value using Method 7.1 can be expressed using Equations 7.16 through 7.19 and is given by the following:

$$\mathrm{R}_1 = \log\left(\frac{P_{13}}{P_{14}} + \frac{P_{14}}{P_{13}}\right) - \log\left(\frac{P_{11}}{P_{12}} + \frac{P_{12}}{P_{11}}\right). \tag{7.20}$$

Method 7.2:

We begin by defining some probabilities:

$$P_{21} = P(T_t \cap A), \tag{7.21}$$
$$P_{22} = P(T_f \cap A), \tag{7.22}$$
$$P_{23} = P(T_t \cap A \cap B), \tag{7.23}$$
$$P_{24} = P(T_f \cap A \cap B). \tag{7.24}$$

Then the reward value using Method 7.2 can be expressed using Equations 7.1, 7.2, 7.16 through 7.19, 7.21 through 7.24 and is given by the following:

$$\mathrm{R}_2 = \left(P_{23}\log\left(\frac{P_{13}}{p}\right) + P_{24}\log\left(\frac{P_{14}}{1-p}\right)\right) - \left(P_{21}\log\left(\frac{P_{11}}{p}\right) + P_{22}\log\left(\frac{P_{12}}{1-p}\right)\right). \tag{7.25}$$

Method 7.3:

For Method 7.3, discrete values were chosen for the 16 combinations of operator response and feature visibility for both visits. A value of zero was assigned if the outcome was impossible, such as the feature being visible on both passes. A small reward was given if the operator was incorrect on both passes but the situation was possible, or when the operator was correct on the first visit but incorrect on the second visit. Moderate rewards were assigned for the operator being incorrect on the first pass, but correct on the second. The largest reward was given when the operator was correct on both visits.

In order to determine a benchmark method to compare Methods 7.1 through 7.3 a comprehensive study using Monte Carlo simulations was done to determine the mean operator response delay. An

operator delay threshold slightly greater than the mean delay was chosen as the constant delay threshold. This was done so that the threshold would envelope a majority of the operator delay times. If the operator delay was less than the threshold, the UAV would perform a revisit.

When one considers the functions given by Equations 7.20 and 7.25, it becomes obvious that several possibly strange events might occur. The first one is due to the structure of Equation 7.20. It is possible that one or more of the probabilities P_{11}, P_{12}, P_{13}, and P_{14} may in fact be equal to zero, resulting in a reward value that is infinite or indeterminate. The second peculiar occurrence is due to the structure of both Equations 7.20 and 7.25. It is possible that appropriate values of $P_{11}, P_{12}, P_{13}, P_{14}, P_{21}, P_{22}, P_{23}$, and P_{24} may render either R_1 or R_2 to be negative. Consider briefly what this means. It means that the information gain due to a second visit of an object of interest is negative, which would imply that one has lost information by performing the revisit. These issues may be undesirable side effects of using these types of reward functions, but they are not crippling. One benefit of Method 7.3 is that there is never a negative, infinite, or indeterminate reward value.

7.2.6 Threshold Surface Plots

Analyzing the surfaces provided by the threshold function provides a preliminary indication of how the system will respond. The threshold surface is determined by the operator's response, the amount of remaining reserve, the number of objects remaining to make a decision about, and the information gain for a revisit [18–20]. The controller compares an expected revisit cost to the corresponding threshold value determined during the solution to the stochastic dynamic program. If the expected cost is less than the threshold and also if it is feasible (satisfies constraints of the system), then the MAV will perform a revisit of the current object of interest. The model used to determine the expected revisit cost is linear and is given by $J(\tau) = 2\tau + \eta$, where τ is the random operator delay and η is a fixed cost that models the UAV's turnaround time. The development of this model is described in more detail in [18–20]. Having a deep intuitive understanding of how the shape of the threshold surface impacts the response of the system is useful but not necessary. We are ultimately concerned with any saturation that occurs along the threshold value axis. Large amounts of saturation indicate a bias toward a certain stochastic controller decision.

For Method 7.1, Figure 7.1 indicates that the case where the operator responds with true results in a very saturated threshold surface. This means that when the operator responds that he or she has seen the distinguishing feature, the stochastic controller will heavily favor revisits in most instances. Whereas the case when the operator responds negatively, shown in Figure 7.2, is not predictable. For Method 7.2, Figure 7.3 indicates that if the operator responds true, then the stochastic controller will most likely not make a revisit, whereas if the operator responds false, the stochastic controller will most likely perform a revisit as shown in Figure 7.4. For Method 7.3, there is not enough saturation in Figures 7.5 and 7.6 to definitely say what trend the stochastic controller might favor. A quantitative analysis with simulations is described in Section 7.4.

7.3 Aerial Surveillance

In this section, a scenario is considered where a team of UAVs is assigned for surveillance of a military base and the surrounding area to protect against potential threats. The path planning for each vehicle is determined by solving an optimization problem, which is described below. A reward function quantifies the amount of information the UAVs have accumulated over a given time horizon T, with $R(z, t)$ denoting the reward at time instant t and z being the collection of position coordinates in $\mathbb{R}^2$ of all the UAVs in the team. The UAVs are assumed to fly at a constant altitude so that the dynamics of the UAVs evolve in a plane. The sensor footprints of the cameras are assumed to be circles that are centered directly below the UAVs, with radius r_i. When the region inside a UAV's footprint is visited, information is

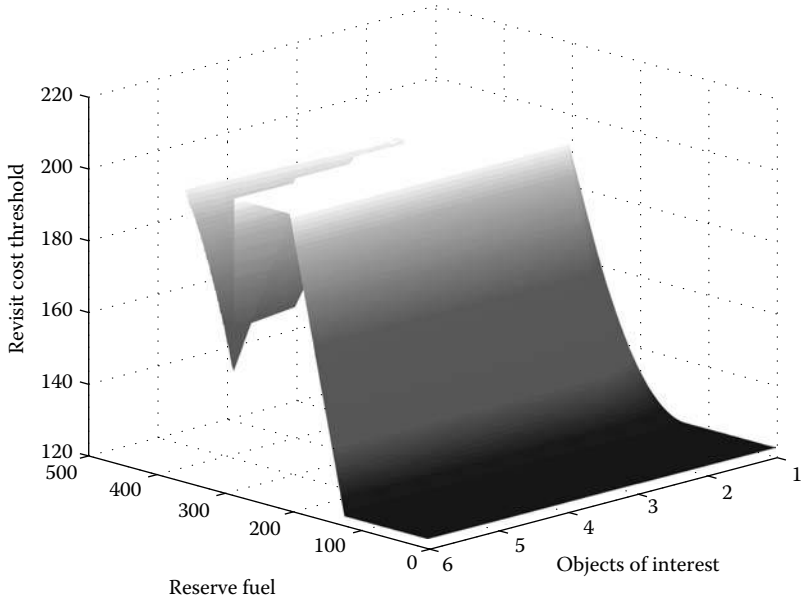

FIGURE 7.1 Method 7.1 Threshold surface plot, operator response is true.

collected. When information is collected in a given area, the reward for that area is set to zero so that the UAVs will tend not to revisit previously viewed areas. Conversely, when regions are outside the UAV's footprint, information grows. For a more detailed listing of the notation for this optimization problem, see Table 7.5.

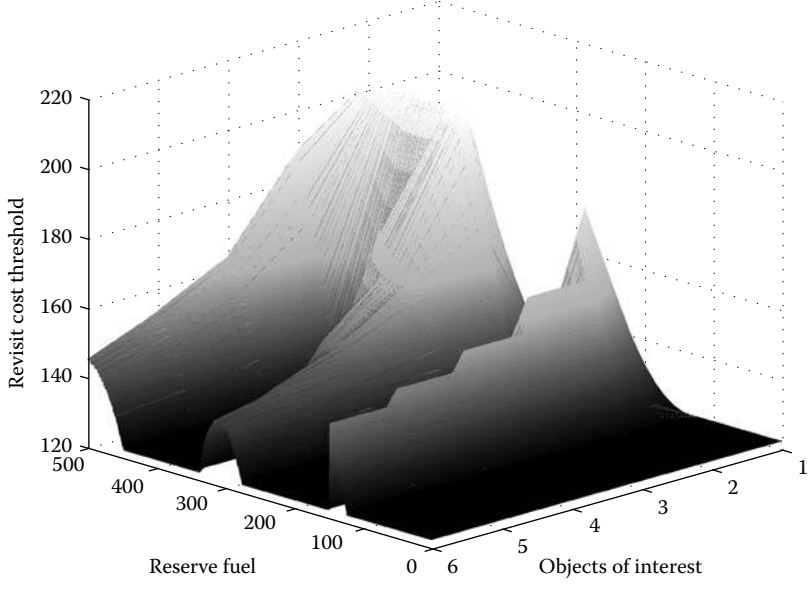

FIGURE 7.2 Method 7.1 Threshold surface plot, operator response is false.

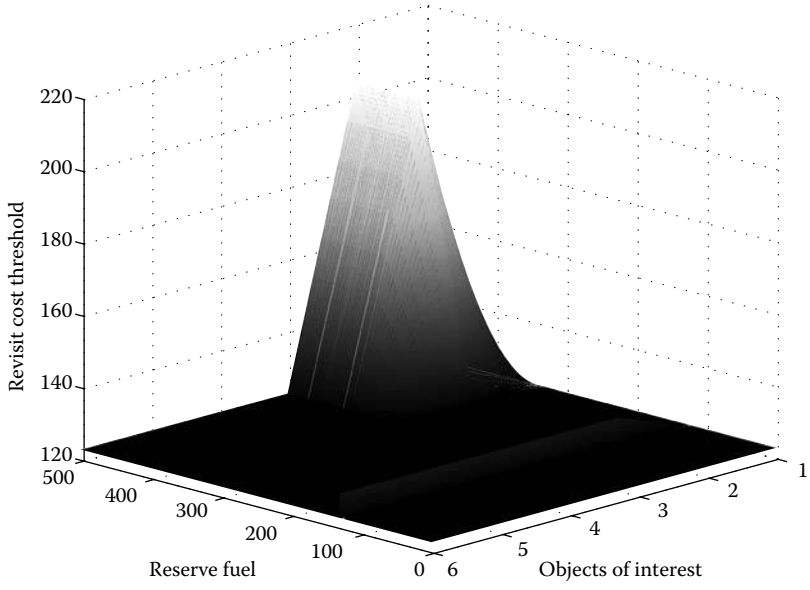

FIGURE 7.3 Method 7.2 Threshold surface plot, operator response is true.

7.3.1 Problem Formulation

First it should be noted that many variables in this section are time dependent, for example, the position and control input of the UAVs. However, explicit functional dependence on the time variable t is suppressed except when it is explicitly desired to indicate the dependence on time.

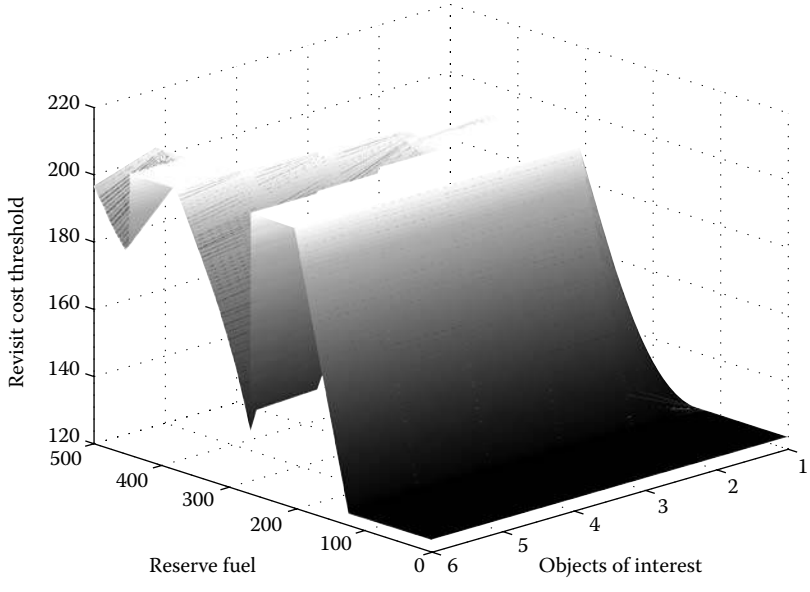

FIGURE 7.4 Method 7.2 Threshold surface plot, operator response is false.

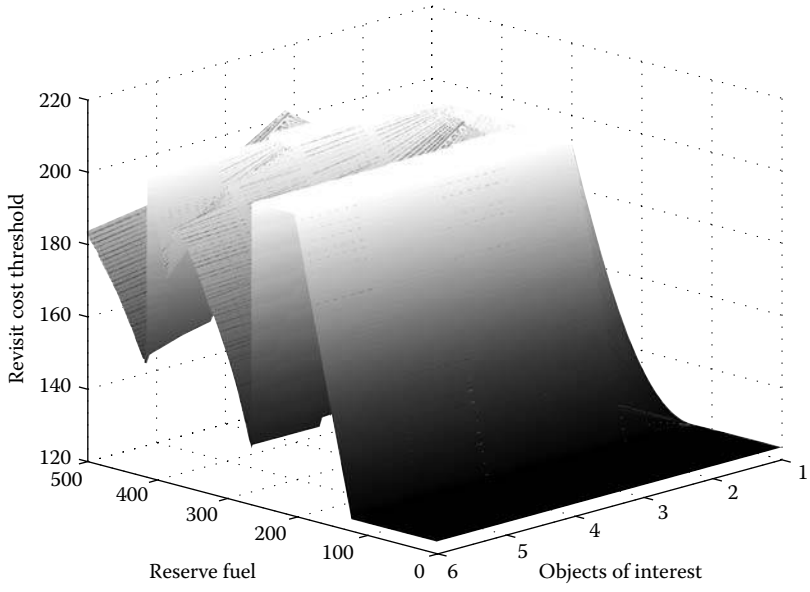

FIGURE 7.5 Method 7.3 Threshold surface plot, operator response is true.

The vehicles are dynamically constrained by the Dubins' vehicle model to have a constant speed and a maximum control input. The system dynamics are given by the following equations:

$$\dot{x}_i(t) = v_i \cos \psi_i(t), \tag{7.26}$$

$$\dot{y}_i(t) = v_i \sin \psi_i(t), \tag{7.27}$$

$$\dot{\psi}_i(t) = u_i(t). \tag{7.28}$$

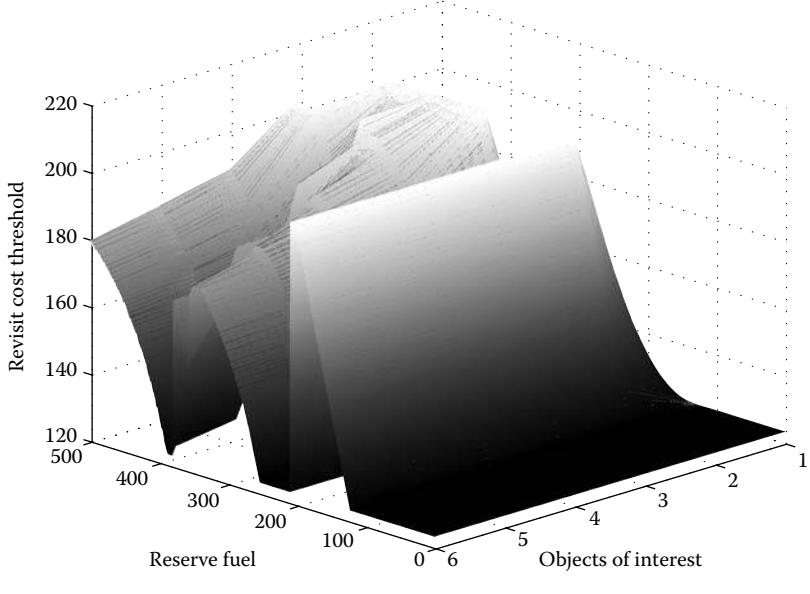

FIGURE 7.6 Method 7.3 Threshold surface plot, operator response is false.

TABLE 7.5 Notation for Optimization Formulation in Section 7.3

Notation	Description
A^*	Transpose of A
$N \in \mathbb{Z}$	Number of UAVs
$T \in \mathbb{Z}$	Time horizon
$z_i = [x_i \;\; y_i]^* \in \mathbb{R}^2$	Position coordinates of vehicle i
$\psi_i \in \mathbb{R}$	Orientation angle of vehicle i
$u_i \in \mathbb{R}$	Control input for vehicle i
$r_i \in \mathbb{R}$	UAV footprint radius of vehicle i
$v_i \in \mathbb{R}$	Constant velocity of vehicle i
$E : \mathbb{R}^3 \to \mathbb{R}$	Environment reward function
$R : \mathbb{R}^{2N+1} \to \mathbb{R}$	Instantaneous reward for all UAVs

In addition to Equations 7.26 through 7.28 the system has the following constraints:

$$\dot{\psi}_i(t) \leq \omega_i, \tag{7.29}$$

$$\bigcup_{i=1}^{N} \bigcup_{t\epsilon[0,T]} \mathfrak{B}[z_i(t), r_i] = \Gamma, \tag{7.30}$$

where ω_i is the maximum angular velocity of vehicle i, Γ is the set of points (region) that is desired to be visited, and $\mathfrak{B}[z_i(t), r_i]$ is defined in Equation 7.31.

In addition to the definitions in Table 7.5, we define $z(t) = [z_1^*(t) \;\; z_2^*(t) \ldots z_N^*(t)]^* \in \mathbb{R}^{2N}$ to be the position coordinates of all N UAVs. The position of each UAV is expressly determined by the control input given to the UAV; we ignore deviations that may be caused by noise of the input signal or any other stochastic events that may affect the position of the UAV. Then, similar to $z(t)$, we define the control inputs of all the UAVs to be $u(t) = [u_1(t) \;\; u_2(t) \ldots u_N(t)]^* \in \mathbb{R}^N$. Moreover, let $\mathcal{U}$ be the set of all feasible inputs, that is, the set of all inputs that allow the system of UAVs to fly feasible paths.

It is also convenient to make the following definitions:

$$\mathfrak{B}[z_i, r_i] = \left\{ \hat{z} \in \mathbb{R}^2 \mid d(\hat{z}, z_i) \leq r_i \right\}, \tag{7.31}$$

$$\widetilde{R}(z_i, t) = \int_{\mathfrak{B}[z_i, r_i]} E(\hat{z}, t) \; dA, \tag{7.32}$$

$$R(z, t) = \sum_{i=1}^{N} \widetilde{R}(z_i, t). \tag{7.33}$$

Equation 7.31 is the closed ball of radius r_i centered at z_i. This function is used to describe the footprint of vehicle i; the function d is simply the Euclidean distance function. Equation 7.32 describes the computation of the local reward information for a vehicle whose footprint is centered at z_i. In this equation $\hat{z}$ is the variable the integration is computed over; it ranges over all points in the set $\mathfrak{B}[z_i, r_i]$. E is the environment reward function that describes the value of visiting a specific point in the domain at time t; in Section 7.4.2 we describe how E explicitly changes with respect to time. The domain might be divided into regions of different but constant reward value initially, and the rewards all grow at the same small constant rate. Another example might have an asset of high value, in which case E might be a quadratic function near this asset and taper off into a low constant plateau further away from the asset. Equation 7.33 is the total reward for all N UAVs at time t. In addition, for notational convenience that will become apparent in

Section 7.4.2 we define

$$v = [v_1 \;\; v_2 \;\; \ldots \;\; v_N]^*, \tag{7.34}$$

$$r = [r_1 \;\; r_2 \;\; \ldots \;\; r_N]^*, \tag{7.35}$$

$$\omega = [\omega_1 \;\; \omega_2 \;\; \ldots \;\; \omega_N]^*. \tag{7.36}$$

We also define $\gamma : [0, T] \to \mathbb{R}^{2N}$ to be a rectifiable path defined as such: for every $t \in [0, T]$, let $\gamma(t) = z(t)$. Furthermore, we shall assume that both u and z are continuously differentiable, that is, $u \in C^1(\mathcal{U})$ and $z \in C^1\left(\mathbb{R}^{2N}\right)$. Finally, our optimization problem may be stated. We want to compute

$$\underset{u \in \mathcal{U}}{\operatorname{argmax}} \int_0^T R(z(t), t)\dot{z}(t)\, dt, \tag{7.37}$$

or equivalently

$$\underset{u \in \mathcal{U}}{\operatorname{argmax}} \int_\gamma R(\gamma(t), t)\, d\gamma(t). \tag{7.38}$$

This optimization problem is not solved analytically in this chapter. Instead, a heuristic optimization technique is employed to approximate the solution to this problem and compute the control inputs for the team of UAVs. The optimization technique used is reviewed in Section 7.3.2.

7.3.2 Review of Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary optimization heuristic that is based upon the social behavior of birds looking for optimal food sources. The algorithm was developed by Russell Eberhart and James Kennedy. It was developed after several attempts to simulate the movement of birds in a flock. Example movements are flocking synchronously, changing directions suddenly, scattering and regrouping [22]. Since then, PSO has been implemented to optimize nonlinear and piecewise continuous functions. PSO is a nongradient optimization heuristic that can be used to search nonconvex spaces. One of the reasons for its growing notoriety is because of its easy implementation compared to other evolutionary algorithms and the quality of its results.

PSO is characterized by a number of particles, $\mathcal{N}_p$, where each has the length of the search space. The position vectors, $\mathcal{X}_i$, of the particles represent a solution to the optimization problem. Each particle also has a velocity, $\mathcal{V}_i$, of size equal to its particle length. The particles have memory of their own best position, $\mathcal{X}_i^b$, as well as the global best position of all the particles $\mathcal{X}^{gb}$. There are examples in the literature that break the particle into neighborhoods, the difference being that the particles only remember $\mathcal{X}_i^b$ and the best in its local neighborhood $\mathcal{X}^{lb}$, which can consist of several "close" particles. Each particle iteratively adjusts its velocity based on $\mathcal{X}_i^b$, $\mathcal{X}^{gb}$ and a weighting, α, of its previous velocity. The new velocity is then added to its current position and the new position offers a new solution to the optimization problem. By tweaking the constants c_1 and c_2 in Equation 7.39, the balance of each particle's exploration versus exploitation can be altered. Typical values for the constants are $c_1 = c_2 = 1.49$ and $\alpha = 0.72$ [23,24]. To start the PSO technique, a random position and velocity is given to each particle. To evaluate the solution, particles' positions are then placed into the function to be evaluated. The PSO algorithm is iterative and the evolutionary equations of particle i are given by the following:

$$\mathcal{V}_i^{k+1} = \alpha\mathcal{V}_i^k + c_1\mu_1\left(\mathcal{X}_i^b - \mathcal{X}_i^k\right) + c_2\mu_2\left(\mathcal{X}^{gb} - \mathcal{X}_i^k\right), \tag{7.39}$$

$$\mathcal{X}_i^{k+1} = \mathcal{X}_i^k + \mathcal{V}_i^{k+1}, \tag{7.40}$$

where μ_1 and μ_2 are random variables with an assumed distribution.

7.3.3 Application of PSO to the Surveillance Problem

In order to use PSO to solve the problem, the time horizon is discretized into T discreet intervals. During each of the intervals we assume that the UAVs have one of three control inputs: turn left, turn right, or go straight. This is captured mathematically by enforcing a constant discrete constraint on u_i. For each $i = 1, \ldots, N$ we want

$$u_i \in \{-\omega_i, 0, \omega_i\}, \quad \omega_i > 0, \tag{7.41}$$

where ω_i is the maximum turning rate of vehicle i. To show how we enforce this constraint we first define the control input sequence for each vehicle as follows: for each $i = 1, \ldots, N$

$$U_i = \left[u_i^1 \;\; u_i^2 \;\; \ldots \;\; u_i^{T-1}\right]^*. \tag{7.42}$$

As the optimization problem is solved using PSO, each iteration will yield values for the u_i^k, however, we must ensure that the constraint given in 7.41 is satisfied. First for each $i = 1, \ldots, N$ define

$$U_i^M = \max_{k=1,\ldots,T-1} \left\{\left|u_i^k\right|\right\}. \tag{7.43}$$

Then a temporary control sequence, $\widetilde{U}_i$, is computed componentwise for each of the N vehicles. For each $i = 1, \ldots, N$ and for each $k = 1, \ldots, T-1$, let

$$\widetilde{U}_i^k = 3\left(\frac{\left|u_i^k\right|}{U_i^M}\right). \tag{7.44}$$

Finally, the actual control sequence is redefined componentwise as follows: For each $i = 1, \ldots, N$ and for each $k = 1, \ldots, T-1$

$$U_i^k = \begin{cases} -\omega_i, & \text{if } \widetilde{U}_i^k \leq 1 \\ 0, & \text{if } 1 < \widetilde{U}_i^k \leq 2 \\ \omega_i, & \text{if } \widetilde{U}_i^k > 2 \end{cases}. \tag{7.45}$$

Now we define the control input sequence $\overline{U} \in \mathbb{R}^{N(T-1)}$, which contains the control input sequence for all N UAVs:

$$\overline{U} = \left[U_1^* \;\; U_2^* \;\; \ldots \;\; U_N^*\right]^*. \tag{7.46}$$

$\overline{U}$ is set up so that the first $T-1$ elements correspond to the control input for the first vehicle; elements T through $2(T-1)$ correspond to the second vehicle and so on. For each vehicle, the first element of U_i is the first control input, and the control inputs are sequentially implemented.

Now that the control sequence for all the UAVs have been defined, the system can be numerically integrated to approximate the states of the vehicles. The sequence of the state vectors is defined exactly like the sequence of the control inputs. The state sequence of each vehicle is defined as follows: for each $i = 1, \ldots, N$

$$X_i = \left[x_i^1 \;\; x_i^2 \;\; \ldots \;\; x_i^T\right], \tag{7.47}$$

$$Y_i = \left[y_i^1 \;\; y_i^2 \;\; \ldots \;\; y_i^T\right], \tag{7.48}$$

$$\Psi_i = \left[\psi_i^1 \;\; \psi_i^2 \;\; \ldots \;\; \psi_i^T\right]. \tag{7.49}$$

Then the state sequence arrays for all vehicles is given by the following:

$$X = [X_1 \;\; X_2 \;\; \ldots \;\; X_N], \tag{7.50}$$

$$Y = [Y_1 \;\; Y_2 \;\; \ldots \;\; Y_N], \tag{7.51}$$

$$\Psi = [\Psi_1 \;\; \Psi_2 \;\; \ldots \;\; \Psi_N]. \tag{7.52}$$

The first element of each X_i, Y_i, Ψ_i corresponds to the initial condition of vehicle i, while the last element of each is the final state of the vehicle. The second element of each X_i, Y_i, Ψ_i corresponds to the states of vehicle i after the control input u_i^1 has been implemented, and so on until the final state is calculated. This is the reason that each vehicle will go through T states in the simulation, but there will only be $T-1$ control inputs. In other words, there are $T-1$ state transitions in the discretized system. A second-order Runge-Kutta method was used for the numerical integration.

Once the vehicle states are known, the reward for the UAV being at that location can be calculated from Equations 7.31 through 7.33. The total reward for the system at the final time T is calculated by summing the rewards for each vehicle at each state in the discretized state array. This entire process has given us only one value of the reward function we are attempting to maximize in Equation 7.38. The PSO algorithm is used to determine new control inputs iteratively, so that a new total reward at time T may be computed. This process continues iteratively until a preselected stopping criterion has been met.

7.4 Simulation and Results

7.4.1 Comparison of Reward Functions for Stochastic Decision Making

Simulations were conducted to test the various reward methods. Each method underwent 100 trials, each trial being 1200 simulation seconds long, with 20 objects of interest and four UAVs. The location, orientation and target truth status of the objects of interest were randomized for each trial. A log was kept of the stochastic controller's decisions throughout all of the trials. The data collected from the trial runs was used for comparison between the various reward methods.

To compare the different methods, a rating system was devised. For all cases where the stochastic controller had a UAV revisit an object of interest, it added a point if either of the following occurred:

1. The operator response was true on one visit, false on the other and the target truth status was true,
2. The operator response was false on both visits and the target truth status was false.

First note that in event 1 above, the orientation of the distinguishing feature and the order of the operator's responses was considered before adding the point; it was not added "blindly." Events 1 and 2 are the best-case scenarios where everything the operator indicates coincides with reality. The points tallied for the case where target truth status is true and the case where it was false was kept separate because they have a different probability of occurrence. The overall points that tally for each case is then divided by the probability of that case to determine a normalized point system. The points for each target truth status case can then be added together to determine an overall score for that method. Table 7.6 summarizes the results.

From Table 7.6, it can be seen that Method 7.2 outperforms the other methods for the case where the target truth status is false. This behavior likely results from the fact that Method 7.2 has a strong preference, due to threshold function saturation, to opt for a revisit when the operator responds "did not

TABLE 7.6 Simulation Results (Target Density: $p = 0.1$)

	Benchmark	Method 7.1	Method 7.2	Method 7.3
T_t mean score (S_t)	0.190	0.610	0.500	0.760
T_t standard deviation	0.419	0.803	0.674	0.842
T_f mean score (S_f)	0.410	0.070	2.080	0.520
T_f standard deviation	0.911	0.432	2.977	1.453
Adjusted S_t $\left(\widetilde{S}_t = S_t/p\right)$	1.900	6.100	5.000	7.600
Adjusted S_f $\left(\widetilde{S}_f = S_f/(1-p)\right)$	0.456	0.078	2.311	0.578
Total score $\left(S = \widetilde{S}_t + \widetilde{S}_f\right)$	2.356	6.178	7.311	8.178

see feature," in combination with the fact that an operator will tend to give this particular response on both visits more often than not simply due to target density. Method 7.3 outperforms the other methods for the case where the target truth status is true, and its overall score indicates that it is the best of the three methods. This could be due to a lack of significant saturation in both threshold functions seen in Figures 7.5 and 7.6. Although it has no strong preference for either response, it seems to perform well on average for both, whereas Methods 7.1 and 7.2 tend to favor a particular response.

7.4.2 Defensive Surveillance Examples

For the simulations used in this section, the following parameters were held constant:

$$
\begin{aligned}
N &= 3 \text{ (number of UAVs)},\\
T &= 20 \text{ (time horizon)},\\
\mathcal{N}_p &= 20 \text{ (number of particles)},\\
v &= [30\ \ 30\ \ 30]^* \text{ (constant velocities of the UAVs)},\\
r &= [3\ \ 3\ \ 3]^* \text{ (radii of the UAVs' footprints)},\\
\omega &= \left[\frac{\pi}{2}\ \ \frac{\pi}{2}\ \ \frac{\pi}{2}\right]^* \text{ (maximum angular velocities of the UAVs)},\\
\rho &= \frac{1}{20} \text{ (growth rate of the environmental reward functions)}.
\end{aligned}
$$

Three examples are described in this section and simulated using MATLAB®. Three different environmental reward functions are described below. These examples do not consider random pop-up alarms within the base. The stopping criterion for the PSO algorithm was simply to stop after a fixed amount of time steps.

The reward functions were chosen to be continuous, positive semidefinite functions. As mentioned in Section 7.3, the environmental reward functions grow at all points z in the domain where there is no UAV that has z in its footprint. It was important to make sure that the rate at which the reward functions grow was not extremely rapid so that the reward function does not "blow up," and also to ensure the UAVs do not linger in the areas where the reward function is largest. In order to describe the environmental reward functions thoroughly, we begin by explicitly writing the value of the each of the three functions at all points z in the domain at $t = 0$. This is shown in Equations 7.53 through 7.55:

$$E_1(z,0) = E_1([x\ \ y]^*,0) = \max\left\{|50-x|,|50-y|\right\}, \tag{7.53}$$

$$E_2(z,0) = E_2([x\ \ y]^*,0) = 10\left(\sin\frac{x}{7}+\sin\frac{y}{7}+2\right), \tag{7.54}$$

$$E_3(z,0) = E_3([x\ \ y]^*,0) = 50-\max\left\{|50-x|,|50-y|\right\}. \tag{7.55}$$

The growth rate of the three environmental reward functions is given by the following differential equation: For each $k \in \{1,2,3\}$ we let

$$\frac{\partial E_k(z,t)}{\partial t} = \rho E_k(z,t). \tag{7.56}$$

Since ρ is constant and we have initial values for all three functions, the solutions to the partial differential equations described by Equations 7.53 through 7.56 are easily obtained. They are given by the following equations:

$$E_1(z,t) = E_1([x\ \ y]^*,t) = e^{\rho t}\max\left\{|50-x|,|50-y|\right\}, \tag{7.57}$$

$$E_2(z,t) = E_2([x\ \ y]^*,t) = 10\left(\sin\frac{x}{7}+\sin\frac{y}{7}+2\right)e^{\rho t}, \tag{7.58}$$

$$E_3(z,t) = E_3([x\ \ y]^*,t) = e^{\rho t}\left[50-\max\left\{|50-x|,|50-y|\right\}\right]. \tag{7.59}$$

E_1 is for perimeter surveillance; E_2 is for lake surveillance; and E_3 is for protecting a center asset.

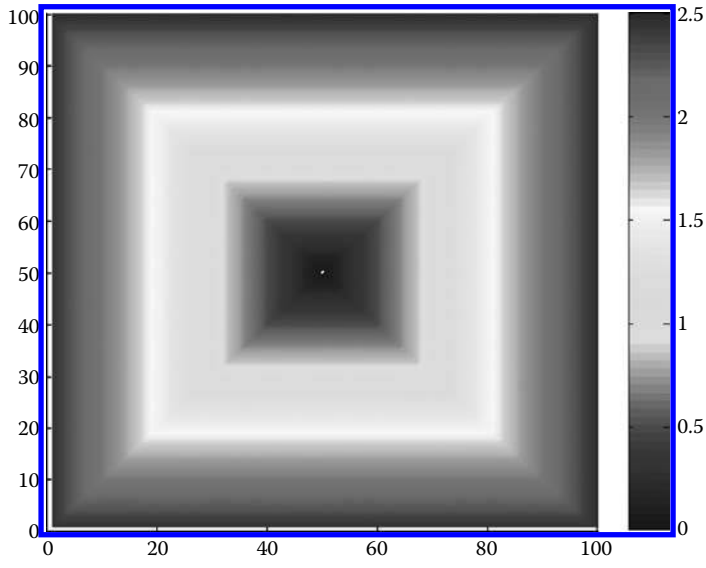

FIGURE 7.7 Contour of $\left.\frac{\partial E_1(z,t)}{\partial t}\right|_{t=0} = \rho E_1(z,0)$ for perimeter surveillance.

7.4.2.1 Perimeter Surveillance

Figure 7.7 shows $\left.\frac{\partial E_1(z,t)}{\partial t}\right|_{t=0} = \rho E_1(z,0)$, the growth rate of E_1 evaluated at $t = 0$. It is proportional to the distance away from the center, as expected for perimeter surveillance. In this example, the UAVs would like to stay near the perimeter of the base to prevent threats from reaching the interior. Figure 7.8 shows the number of times a specific area was within the footprint of any UAV. A quantitative analysis using a well-defined metric was not done, but one may visually inspect and compare Figures 7.7 and 7.8. In the limit, the idea is for the two figures to look identical. Obviously, the simulation runs only for a

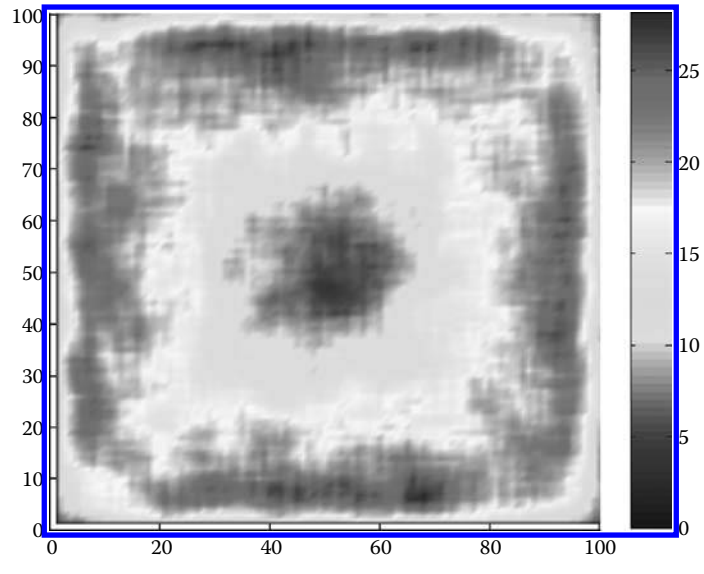

FIGURE 7.8 Number of visits per 2000 steps for perimeter surveillance.

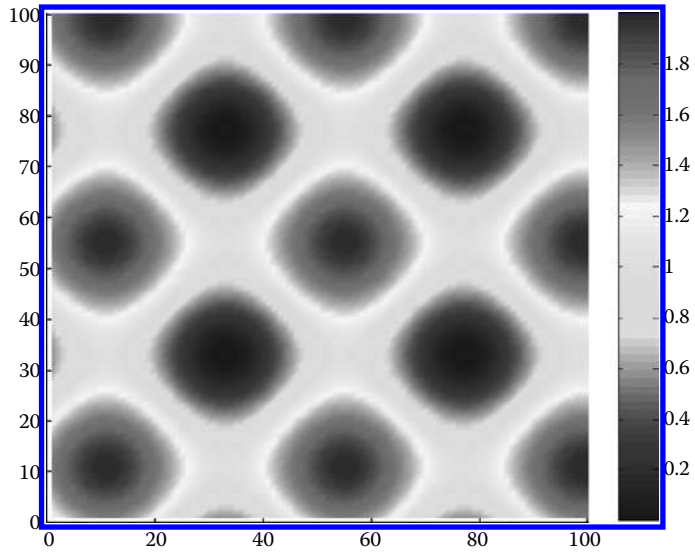

FIGURE 7.9 Contour of $\left.\dfrac{\partial E_2(z,t)}{\partial t}\right|_{t=0} = \rho E_2(z,0)$ for lake surveillance.

finite time, however, the results are very promising. The UAVs highly concentrated their time to the outer edges of the base, as desired in perimeter surveillance.

7.4.2.2 Lake Surveillance

Figure 7.9 shows $\left.\dfrac{\partial E_2(z,t)}{\partial t}\right|_{t=0} = \rho E_2(z,0)$, the growth rate of E_2 evaluated at $t = 0$. This example is a surveillance scenario where there are regions of very little interest such as small lakes held within the base (dark gray regions centered at (30,30), (80,30), (30,80), and (80,80)). The probability of a target being in

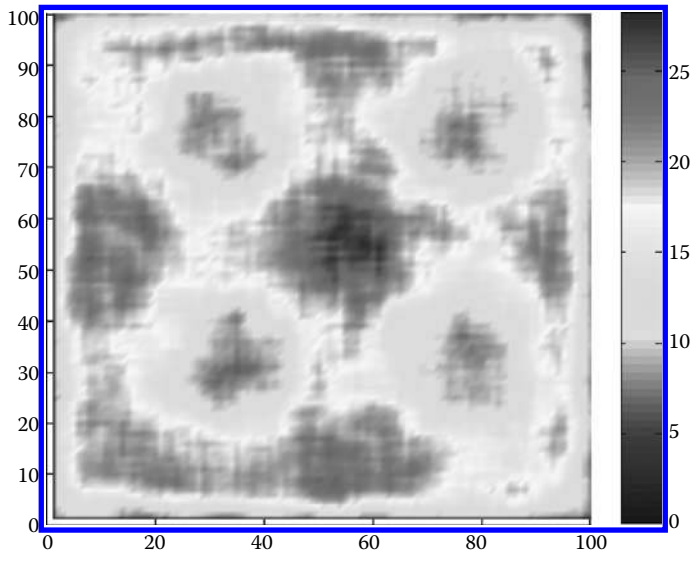

FIGURE 7.10 Number of visits per 2000 steps for lake surveillance.

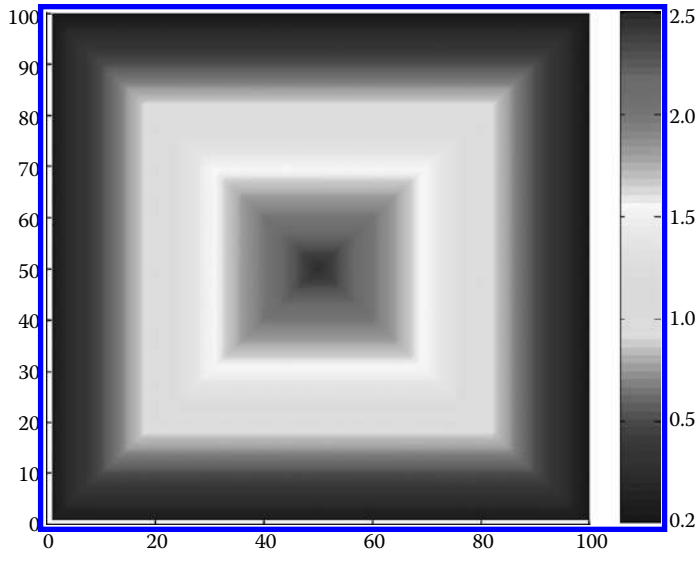

FIGURE 7.11 Contour $\left.\frac{\partial E_3(z,t)}{\partial t}\right|_{t=0} = \rho E_3(z,0)$ for center asset surveillance.

these regions is very low, while regions outside the lakes are of greater interest. The goal in the scenario is to spend almost no time viewing regions over the lakes, but instead visit the hot spots such as roads. Figure 7.10 shows that the UAVs donate the majority of their time to the more important regions outside of the lakes, where the environment has the highest reward.

7.4.2.3 Center Asset

Figure 7.11 shows $\left.\frac{\partial E_3(z,t)}{\partial t}\right|_{t=0} = \rho E_3(z,0)$, the growth rate of E_3 evaluated at $t = 0$. This scenario considers a situation where there is a stationary object at the center of the base that is considered to be

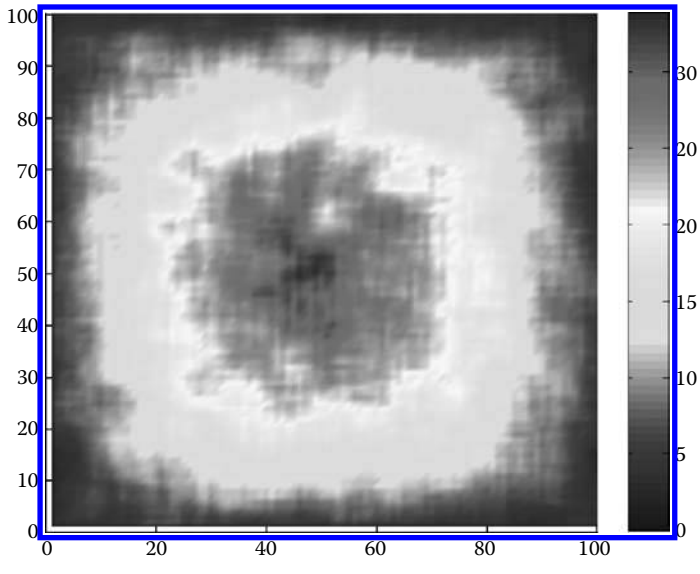

FIGURE 7.12 Number of visits per 2000 steps for center asset surveillance.

a high value asset. As such, protection of this asset is of great importance. Therefore, surveillance of the base should concentrate on this center asset. Figure 7.12 shows the results from using PSO. One can easily see that the inner most region is viewed much more than the outer regions of the base. Again the results are very promising.

7.5 Conclusion

The performance, and perhaps effectiveness, of the stochastic controller depends strongly on the reward function. To evaluate the performance of a variety of reward methods, hundreds of simulation trials were performed and the resulting data were analyzed by a scoring algorithm. The results of the scoring algorithm seem to indicate that the quality of a particular reward function is related to the amount of saturation appearing in the threshold surfaces for that method. Although threshold surface saturation is not necessarily bad, it is not optimal. The method where saturation was mostly avoided was the method that performed the best. It was also shown that a discrete reward method can outperform an information theoretical method.

Essentially, the reward function outputs two values, expected rewards for the cases where the operator responds true or false. It may not be necessary to have a function dedicated to the determination of these values based on probabilities in the scenario. Instead, the two expected reward values could be determined using an optimization method (reinforcement learning perhaps) that adjusts the values over a series of simulations.

The defensive aerial surveillance algorithm in this chapter can easily be implemented and used for surveillance in both military and civilian applications. The results shown in Section 7.4 show that PSO is a viable approach to constructing trajectories for difficult cooperative surveillance problems. We have demonstrated that the PSO method combines both exploration and exploitation of important areas for surveillance. The algorithm at hand can be used to calculate trajectories online and within reasonable time. Furthermore, the PSO algorithm generates unpredictable paths online, which makes it difficult for adversaries to predict the paths of the UAVs.

The goal of this chapter was to present two novel approaches to two different and very important topics in the ever growing research field of UAVs performing military ISR missions. This is a field that has a multitude of challenges, some that have been solved but most that have not. UAV platform design is one area that is being heavily researched by numerous commercial and government research groups. However, the basics of autopilot design and aerodynamics for most classes of UAVs has been thoroughly researched and is a topic that is fairly well understood. This is not true for UAVs that are truly on the micro scale, and there is plenty of research on maturing that technology as well. It appears that the chief problem area for UAVs performing military ISR missions is something that is generally referred to as outer-loop control. Things like adaptive task assignment and path planning, autonomous network formation, mixed-initiative (human and computer) control of assets, machine learning, and decision making with uncertainty are topics that are most important to make the next leap of autonomy that will be essential for military dominance in the near future. One might liken the challenges to achieve this as being a solid polyhedron with many (possibly hundreds) faces. In this chapter we have but scratched the surface of one of the faces, let alone delve deep into the polyhedron to achieve a true understanding of the topic.

References

1. D. Gross, S. Rasmussen, P. Chandler, and G. Feitshans. Cooperative operations in UrbaN TERrain (COUNTER). In *Proceedings of the SPIE*, Vol. 6249 of *Defense Transformation and Network-Centric Systems*, Orlando, FL, May 2006.
2. N. Jodeh, M. Mears, and D. Gross. Overview of the cooperative operations in UrbaN TERrain (COUNTER) program. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Honolulu, HI, August 2008.

3. S. Rasmussen and T. Shima. Tree search algorithm for assigning cooperating UAVs to multiple tasks. *International Journal of Robust and Nonlinear Control*, 18(2), 135–153, 2008.
4. T. Shima and S. Rasmussen, editors. *Unmanned Aerial Vehicles, Cooperative Decision and Control: Challenges and Practical Approaches*, 1st ed. Advances in Design and Control. SIAM, Philadelphia, PA, December 2008.
5. T. Shima, S. Rasmussen, and D. Gross. Assigning micro UAVs to task tours in an urban terrain. *IEEE Transactions on Control Systems Technology*, 15(4), 601–612, 2007.
6. T. Shima, S. Rasmussen, A. Sparks, and K. Passino. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers and Operations Research*, 33(11), 3252–3269, 2006.
7. D. Enns, D. Bugajski, and S. Pratt. Guidance and control for cooperative search. In *Proceedings of the American Control Conference*, Anchorage, AK, May 2002.
8. A. Ahmadzadeh, A. Jadbabaie, V. Kumar, and G. Pappas. Stable multi-particle systems and application in multi-vehicle path planning and coverage. In *Proceedings of the IEEE Conference on Decision and Control*, New Orleans, LA, December 2007.
9. P. DeLima and D. Pack. Toward developing an optimal cooperative search algorithm for multiple unmanned aerial vehicles. In *Proceedings of the International Symposium on Collaborative Technologies and Systems*, Irvine, CA, May 2008.
10. A. Ahmadzadeh, A. Jadbabaie, V. Kumar, and G. Pappas. Cooperative coverage using receding horizon control. In *Proceedings of the European Control Conference*, Kos, Greece, July 2007.
11. C. Cassandras and W. Li. A receding horizon approach for solving some cooperative control problems. In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.
12. M. Flint, M. Polycarpou, and E. Fernandez-Gaucherand. Cooperative control for multiple autonomous UAVs searching for targets. In *Proceedings of the IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.
13. P. Hokayem, D. Stipanovic, and M. Spong. On persistent coverage control. In *Proceedings of the IEEE Conference on Decision and Control*, New Orleans, LA, December 2007.
14. J. Ousingsawat and M. Earl. Modified lawn-mower search pattern for areas comprised of weighted regions. In *Proceedings of the American Control Conference*, New York, NY, July 2007.
15. A. Girard, A. Howell, and J. Hedrick. Border patrol and surveillance missions using multiple unmanned air vehilces. In *Proceedings of the IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.
16. R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. *Proceedings of the IEEE*, 94(7), 1306–1324, July 2006.
17. M. Pachter, P. Chandler, and S. Darbha. Optimal control of an ATR module equipped MAV-human operator team. In *Cooperative Networks: Control and Optimization — Proceedings of the 6th International Conference on Cooperative Control and Optimization*, Edward Elgar Publishing, Northampton, MA, 2006.
18. A. Girard, S. Darbha, M. Pachter, and P. Chandler. Stochastic dynamic programming for uncertainty handling in UAV operations. In *Proceedings of the American Control Conference*, New York, NY, July 2007.
19. A. Girard, M. Pachter, and P. Chandler. Optimal decision rules and human operator models for UAV operations. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, Keystone, CO, August 2006.
20. M. Pachter, P. Chandler, and S. Darbha. Optimal sequential inspections. In *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, December 2006.
21. M. Pachter, P. Chandler, and S. Darbha. Optimal MAV operations in an uncertain environment. *International Journal of Robust and Nonlinear Control*, 18(2), 248–262, January 2008.
22. J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE Conference on Neural Networks*, Perth, Australia, November 1995.
23. I. Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85(6), 317–325, March 2003.
24. F. van den Bergh and A. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8), 937–971, April 2006.

8

Control Allocation

Michael W. Oppenheimer
Air Force Research Laboratory

David B. Doman
Air Force Research Laboratory

Michael A. Bolender
Air Force Research Laboratory

8.1 Introduction

Over the past few decades, much emphasis has been placed on over-actuated systems for air vehicles. Over-actuating an air vehicle provides a certain amount of redundancy for the flight control system, thus potentially allowing for recovery from off-nominal conditions. Due to this redundancy, control allocation algorithms are typically utilized to compute a unique solution to the over-actuated problem. Control allocators compute the commands that are applied to the actuators so that a prescribed set of forces or moments are generated by the control effectors. Usually, control allocation problems are formulated as optimization problems so that all of the available degrees of freedom can be utilized and, when sufficient control power exists, secondary objectives can be achieved.

A conventional aircraft utilizes an elevator for pitch control, ailerons for roll control, and a rudder for yaw control. As aircraft designs have advanced, more control effectors (some unconventional) have been placed on the vehicles. In some cases, certain control effectors may be able to exert significant influence upon multiple axes. When a system is equipped with more effectors than controlled variables, the system may be over-actuated. The allocation, blending, or mixing of these control effectors to achieve some desired objectives constitute the control allocation problem.

Due to over-actuation and the influence of control surfaces on multiple controlled variables, it can be difficult to determine an appropriate method of how to translate a controlled variable command into a control surface command. Some air vehicle concepts have been designed with 10 or more control effectors and only three controlled variables. As the number of control effectors increases, the determination of *ad hoc* control allocation schemes becomes more difficult and the need for systematic control allocation

algorithms increases. In addition, rate and position limits of the control effectors must be considered in order to achieve a realistic solution. Not only is the mixing of control surface effects critical, but it is also desirable to enable the aircraft to recover from off-nominal conditions, such as a failed control surface, when physically possible. Reconfigurable controllers can adjust control system parameters to adapt to off-nominal conditions [1–5]. In reconfigurable control systems, a control allocation algorithm can be used to perform automatic redistribution of the control power requests among a large number of control effectors, while still obeying the rate and position limits of the actuators.

In the most general case, ignoring rate and position limits of the control effectors, the control allocation problem is to find the control effector vector, $\boldsymbol{\delta} \in \mathbb{R}^n$, such that

$$\mathbf{f}(\boldsymbol{\delta}) = \mathbf{d}_{des} \tag{8.1}$$

where $\mathbf{d}_{des} \in \mathbb{R}^m$ is a vector of desired quantities and $\mathbf{f}(\boldsymbol{\delta}) \in \mathbb{R}^m$ is a vector of linear and/or nonlinear functions of the controls. A simple example of a control allocation problem is to find δ_1 and δ_2 such that

$$3\delta_1 + \delta_2 = 2 \tag{8.2}$$

In this case, $d_{des} = 2$ and $f(\boldsymbol{\delta}) = 3\delta_1 + \delta_2$ are scalars and $\boldsymbol{\delta} = [\delta_1 \;\; \delta_2]^T$. Equation 8.2 is linear in the control vector and can be written in matrix form as

$$\mathbf{B}\boldsymbol{\delta} = d_{des} \tag{8.3}$$

where

$$\mathbf{B} = \begin{bmatrix} 3 & 1 \end{bmatrix} \tag{8.4}$$

An example of a nonlinear control allocation problem is

$$3\delta_1^2 + \delta_2^3 = 2 \tag{8.5}$$

Essentially, the control allocation problem without rate and position limits is the standard mathematical problem of simultaneously solving a set of equations in which there are more unknowns than equations, that is, $n > m$ in Equation 8.1. In a linear framework, this is equivalent to the linear algebra problem of finding a solution to $\mathbf{Ax} = \mathbf{b}$ when there are more columns in $\mathbf{A}$ than rows.

There are three main goals of a control allocation algorithm.

- Determine a unique solution to Equation 8.1 when multiple solutions exist.
- Obey physical limitations of the control effectors, namely, rate and position limits.
- Determine a "best" configuration of control settings when no solution exists.

Reconsidering Equation 8.2, it is obvious that many solutions exist to this problem. For example, a few solutions to $3\delta_1 + \delta_2 = 2$ are

- $\delta_1 = 0$ and $\delta_2 = 2$
- $\delta_1 = \dfrac{2}{3}$ and $\delta_2 = 0$
- $\delta_1 = -3$ and $\delta_2 = 11$

Figure 8.1 illustrates a simple two-dimensional problem of finding $\boldsymbol{\delta}$ such that $3\delta_1 + \delta_2 = d_{des}$ with constraints on the values of the elements of $\boldsymbol{\delta}$. The square in Figure 8.1 defines a set of position limits for the actuators. In this case, both δ_1 and δ_2 are limited to $-1 \le \delta_1 \le 1$ and $-1 \le \delta_2 \le 1$. Of the three solutions given above, only the second one satisfies the limits on the control effector positions. Including these limits yields the *constrained* control allocation problem:

$$\begin{aligned} &\text{Find } \delta_1, \delta_2 \text{ such that } d_{des} = 3\delta_1 + \delta_2 \\ &\text{subject to} -1 \le \delta_1 \le 1, \;\; -1 \le \delta_2 \le 1 \end{aligned} \tag{8.6}$$

Four lines are drawn in Figure 8.1 corresponding to different values of d_{des}. Each line represents the equation $3\delta_1 + \delta_2 = d_{des}$ with $d_{des} = 2, 3, 4, 5$. The solution is the intersection of the hyperspaces of the

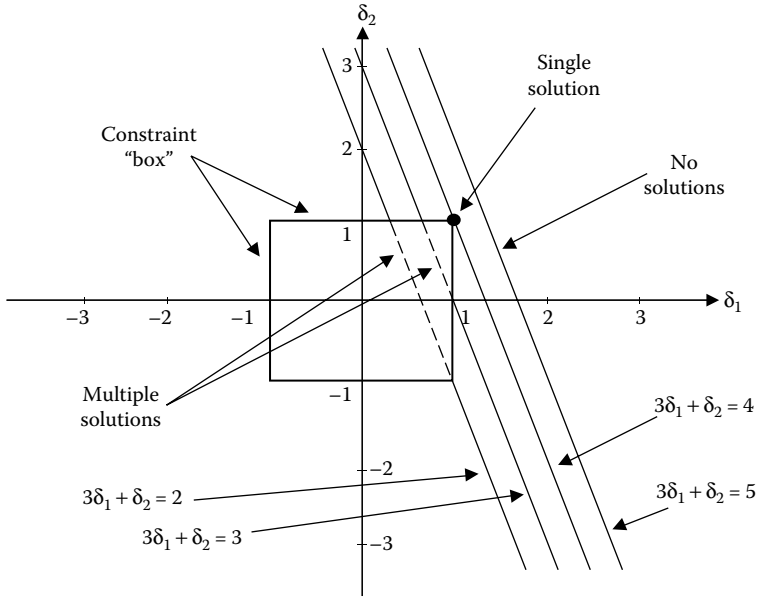

FIGURE 8.1 Example of control allocation.

constraints and the equation $d_{des} = 3\delta_1 + \delta_2$. For $d_{des} = 2$ or 3, multiple solutions exist, as shown by the dashed lines. For $d_{des} = 4$, only one solution exists and it is the point where $\delta_1 = \delta_2 = 1$, while for $d_{des} = 5$, no solutions exist. When only one solution exists, simply select that solution. When multiple solutions exist, a method to pick a single solution is necessary. When no solution exists, a method to minimize the error, between the desired and attainable quantities, is required. This illustrates the constrained control allocation problem.

This chapter describes techniques that can be used to solve unconstrained and constrained control allocation problems. The constraints place limits on the rate and position of control effectors so that the physical limitations of the actuation device are not violated. Some of the simplest control allocation techniques are explicit ganging, pseudo control, pseudo inverse, and daisy chaining. Unfortunately, each of these techniques suffer from a difficulty in guaranteeing that rate and position limits will not be violated and can be difficult to apply because of the need to derive a control mixing law *a priori*. A constrained control allocation method, called direct allocation [6], finds the control vector that results in the best approximation of the desired vector in a given direction. Unconstrained least squares control allocation methods, that account for rate and position limits, through the use of penalty functions, have also been developed [7]. One of the first instances of linear programming (LP)-based control allocators was from Paradiso [8,9]. In this work, Paradiso developed a selection procedure for determining actuator positions, for limited authority systems, that was based on LP. More recently, the constrained control allocation problem has been posed as a constrained optimization problem [10]. In this work, the control allocation problem was split into two subproblems. The first was an error minimization problem, which attempted to find a control vector such that the control effector induced moments or accelerations matched the desired moments or accelerations. If multiple solutions exist to the error minimization problem, the second problem attempts to find a unique solution by driving the control vector to a preference vector that optimizes a secondary objective. The linear constrained control allocation problem has also been extended to an affine problem [11] to account for nonlinearities in the moment–deflection curves. Quadratic programming has also been used to solve constrained control allocation problems [12]. An excellent paper discussing control allocation, by Bodson [13], provides a glimpse into numerous control allocation techniques.

8.2 Historical Perspective

Control allocation is a general term that describes a process that is used to determine how to employ a number of control effectors to achieve a desired reaction from a system. Control allocation methodologies have been used in the aircraft and automobile fields since the inception of the vehicles themselves.

Early aircraft were controlled by a pilot who manipulated the stick and rudder pedals to actuate the ailerons, elevators, and rudders. The stick and rudder pedals were mechanically connected to the aircraft control surfaces by cables and pulleys. The number of control surfaces on a typical early monoplane was five. The control surface suite consisted of two ailerons, one rudder, and two elevators. A stick that controlled the elevator and ailerons had two degrees of freedom, fore-aft and side-to-side. By moving the stick, the pilot could control four surfaces. Mechanical connections that constrained the movements of these surfaces resulted in a control allocation strategy called "ganging." Ganging is a technique in which multiple control surfaces are constrained to move together. In the case of left and right ailerons, a common ganging strategy was to constrain the surfaces to deflect differentially. To roll the vehicle right wing down, the right aileron deflected trailing edge up and the left aileron moved trailing edge down, each deflecting by the same amount. Figure 8.2 shows the mechanical connections that were typically used to gang ailerons. As the pilot moved the stick left or right, the cables and pulleys constrained the ailerons to deflect differentially.

For the elevators, the two surfaces were constrained to move symmetrically, that is, both surfaces move trailing edge up or down by the same amount. In this way, the two ailerons and two elevators were constrained to move as one effective aileron and one effective elevator. This ganging strategy has the effect of reducing the dimension of the control space from five original control surfaces to three effective control surfaces.

The purpose of ganging is to simplify the allocation of control tasks to multiple effectors. For an aircraft, one objective is to maintain control over the three rotational degrees of freedom, namely, pitch, roll, and yaw. By ganging the control surfaces as described earlier, the number of effective controls (3) is the same as the number of rotational degrees of freedom to control (3). Thus, it is straightforward to allocate control tasks among the three effective control surfaces. The effective aileron controls the roll axis, the effective elevator controls the pitch axis, and the rudder controls the yaw axis. This technique of ganging is suitable under nominal conditions, however, in certain cases, such as when a control surface fails or is damaged, it may not take full advantage of all available control effectiveness.

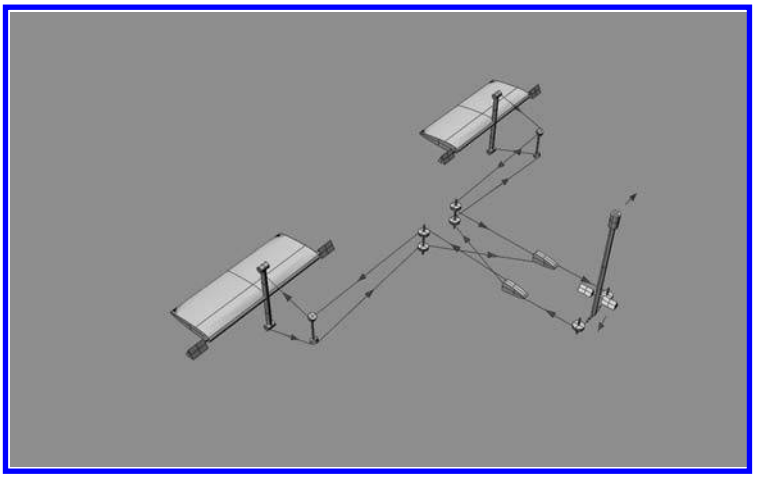

FIGURE 8.2 Mechanical connections to gang ailerons.

Another example of ganging can be found in traditional automobiles. The driver controls the direction of travel by turning the steering wheel, which mechanically causes the two front wheels to turn together at the same angle. When the driver applies a force to the brake pedal, forces are applied to drums or rotors on all four wheels in order to slow down their rotation. Braking systems are constructed such that the forces applied at the front and rear brakes are ganged to take advantage of the greater braking forces that can be achieved by the front wheels. On a motorcycle, the operator has the ability to independently control the front and rear braking forces. As a result, a skilled rider can use this extra freedom to achieve feats that would be impossible with a ganged system.

Historically, designers have intelligently designed control effector ganging strategies to operate well under nominal conditions; however, it is possible to blend control effectors in other ways to improve performance under certain situations. Mechanically ganged effectors may lead to undesirable consequences during a failure, due to the restrictions of forcing multiple surfaces to act as one.

The point of this discussion is that there are potentially many ways of blending control effectors together in order to achieve a desired vehicle response. Furthermore, when designing modern control systems for vehicles that are not constrained by mechanical connections, the designer can blend control effectors together in ways that reduce or eliminate the effects of failures or achieve new levels of performance in different situations.

Vehicle designers have now included many control effectors to add redundancy, some of which can exert control over multiple controlled variables. An example of a control surface that can affect more than one controlled variable is a twin rudder on an air vehicle, that is canted into a "V" tail configuration. Each of these surfaces can typically exert control over the rolling, pitching, and yawing motion. The use of many control effectors, some of which may be "unconventional" control surfaces, makes it difficult to determine a suitable ganging scheme. In these cases, a control allocation algorithm can be used to systematically determine the control settings necessary to produce a desired response.

8.3 Linear Control Allocation

The linear control allocation problem is defined as follows: find the control vector, $\boldsymbol{\delta} \in \mathbb{R}^n$, such that

$$\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des} \tag{8.7}$$

subject to

$$\begin{aligned} \boldsymbol{\delta}_{min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{max} \\ |\dot{\boldsymbol{\delta}}| \leq \dot{\boldsymbol{\delta}}_{max} \end{aligned} \tag{8.8}$$

where $\mathbf{B} \in \mathbb{R}^{m \times n}$ is the control effectiveness matrix, the lower and upper position limits are defined by $\boldsymbol{\delta}_{min} \in \mathbb{R}^n$ and $\boldsymbol{\delta}_{max} \in \mathbb{R}^n$, respectively, $\dot{\boldsymbol{\delta}} \in \mathbb{R}^n$ are the control rates, $\dot{\boldsymbol{\delta}}_{max} \in \mathbb{R}^n$ are the maximum control rates, $\mathbf{d}_{des} \in \mathbb{R}^m$ are desired quantities or controlled variables, n is the number of control effectors, and m is the number of controlled variables. Typically, for aircraft inner-loop control laws, $\mathbf{d}_{des} \in \mathbb{R}^3$, corresponding to the three rotational degrees of freedom. Equation 8.8 describes the position and rate limits for the control effectors. In a digital computer implementation, the rate limits can be converted into effective position limits. The combined limits, $\underline{\boldsymbol{\delta}} \in \mathbb{R}^n$, $\overline{\boldsymbol{\delta}} \in \mathbb{R}^n$, become the most restrictive of the rate or position limits and are specified as

$$\begin{aligned} \overline{\boldsymbol{\delta}} &= \min\left(\boldsymbol{\delta}_{max}, \boldsymbol{\delta} + \Delta t \dot{\boldsymbol{\delta}}_{max}\right) \\ \underline{\boldsymbol{\delta}} &= \max\left(\boldsymbol{\delta}_{min}, \boldsymbol{\delta} - \Delta t \dot{\boldsymbol{\delta}}_{max}\right) \end{aligned} \tag{8.9}$$

where Δt is the sampling interval and the elements of $\boldsymbol{\delta}$ describe the current positions of each control effector. The constraints in Equation 8.8 then become

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \overline{\boldsymbol{\delta}} \tag{8.10}$$

Note that the $\boldsymbol{\delta}$ in Equation 8.7 is what is being computed while the $\boldsymbol{\delta}$ in Equation 8.8 is the current location of the control effectors.

A necessary condition for a system to be over-actuated is that the number of columns of $\mathbf{B}$ must be greater than the number of rows of $\mathbf{B}$, that is, $n > m$. A necessary and sufficient condition for over-actuation is that the number of linearly independent columns of $\mathbf{B}$ must be greater than the number of rows of $\mathbf{B}$. For aircraft inner-loop control laws, the control effectiveness matrix typically is of the form:

$$\mathbf{B} = \begin{bmatrix} \dfrac{\partial L}{\partial \delta_1} & \dfrac{\partial L}{\partial \delta_2} & \cdots & \dfrac{\partial L}{\partial \delta_n} \\ \dfrac{\partial M}{\partial \delta_1} & \dfrac{\partial M}{\partial \delta_2} & \cdots & \dfrac{\partial M}{\partial \delta_n} \\ \dfrac{\partial N}{\partial \delta_1} & \dfrac{\partial N}{\partial \delta_2} & \cdots & \dfrac{\partial N}{\partial \delta_n} \end{bmatrix} \tag{8.11}$$

where L, M, and N are the rolling, pitching, and yawing moments, respectively.

Equations 8.7, 8.10, and 8.9 define the linear control allocation problem. The objective is to determine methods that allow computation of the control effector vector $\boldsymbol{\delta}$, while possibly taking into account effector rate and position limits. The following discussion provides an introduction to methods that can be utilized to either reduce the dimension of the over-actuated system to the point that a square allocation problem results ($n = m$) or to directly solve the over-actuated linear control allocation problem.

8.3.1 Unconstrained Linear Control Allocation

In unconstrained linear control allocation, position and rate limits of the control effectors are ignored. Here, the objective is to find $\boldsymbol{\delta}$ such that $\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$.

8.3.1.1 Matrix Inverse

A special situation arises when the control effectiveness matrix, $\mathbf{B}$, is square and invertible. A square $\mathbf{B}$ matrix implies that the number of control effectors equals the number of controlled variables, that is, $\boldsymbol{\delta} \in \mathbb{R}^n$ and $\mathbf{d}_{des} \in \mathbb{R}^n$. If $\mathbf{B}$ is square and invertible (full rank), then the control allocation solution is a standard inverse:

$$\boldsymbol{\delta} = \mathbf{B}^{-1}\mathbf{d}_{des} \tag{8.12}$$

If the entries in $\mathbf{B}$ are constant, that is, each entry in Equation 8.11 does not change when the operating condition changes, then $\mathbf{B}$ and $\mathbf{B}^{-1}$ can be computed *a priori*. The user must determine, for the given application, whether the entries in $\mathbf{B}$ are sufficiently constant to be able to accurately compute $\mathbf{B}^{-1}$ offline. If they are not, then $\mathbf{B}^{-1}$ can be computed online.

For over-actuated systems, $\mathbf{B}$ is not square. However, a number of techniques exist that can reduce the control space dimension. Upon reducing the control dimension, it is possible to transform a nonsquare control allocation problem into a square allocation problem, such that it may be solved with Equation 8.12. The next section discusses one such technique.

8.3.1.2 Explicit Ganging

In this approach, an *a priori* method is used to combine or gang the effectors to reduce the number of effective control devices. Historically, ganging was done with cables, pulleys, or other mechanical means. On modern fly-by-wire aircraft, ganging is performed in software. The goal is to find a matrix $\mathbf{G} \in \mathbb{R}^{n \times p}$, where $p \leq n$, that relates a set of pseudo controls, $\boldsymbol{\delta}_{pseudo} \in \mathbb{R}^p$, to the actual controls, $\boldsymbol{\delta} \in \mathbb{R}^n$, such that

$$\boldsymbol{\delta} = \mathbf{G}\boldsymbol{\delta}_{pseudo} \tag{8.13}$$

For example, consider a vehicle that has left and right ailerons for roll control, δ_{a_L} and δ_{a_R}, a rudder for yaw control, δ_r, and an elevator for pitch control, δ_e. *A priori*, a ganging law can be constructed

to produce a single roll control device. One possibility is to let

$$\delta_a = 0.5\left(\delta_{a_L} - \delta_{a_R}\right) \tag{8.14}$$

where δ_a is the single effective roll control device. Therefore, the full ganging law becomes

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_{a_L} \\ \delta_{a_R} \\ \delta_e \\ \delta_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix} = \mathbf{G}\boldsymbol{\delta}_{pseudo} \tag{8.15}$$

Here, the reason for the term pseudo controls becomes clear because one of the elements of $\boldsymbol{\delta}_{pseudo}$, namely, δ_a, is not a physical control, instead, it is a linear combination of two physical control effectors, δ_{a_L} and δ_{a_R}. Then, the control allocation problem is to find $\boldsymbol{\delta}_{pseudo}$, such that

$$\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des} \Rightarrow \mathbf{BG}\boldsymbol{\delta}_{pseudo} = \mathbf{d}_{des} \tag{8.16}$$

Solving this allocation problem for $\boldsymbol{\delta}_{pseudo}$ and using Equation 8.15 yields the physical control effector commands. An explicit ganging strategy is determined offline and is typically used when it is obvious how to combine redundant control effectors. If the entries in **B** can be modeled as constants, then **BG** can be computed offline.

It is important to point out that this method can be used to reduce the control space dimension of an over-actuated system. As previously mentioned, aircraft inner-loop control laws typically contain three objective functions, namely that the moments produced by the controls are equal to a set of desired moments ($\mathbf{d}_{des}$). In the explicit ganging example above, $\boldsymbol{\delta} \in \mathbb{R}^4$ and if $\mathbf{d}_{des} \in \mathbb{R}^3$, then $\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$ is a nonsquare control allocation problem. After employing the explicit ganging methodology, $\boldsymbol{\delta}_{pseudo} \in \mathbb{R}^3$ so $\mathbf{BG}\boldsymbol{\delta}_{pseudo} = \mathbf{d}_{des}$ is a square control allocation problem. Hence, in the above example, the dimension of the control space is reduced from 4 to 3.

8.3.1.3 Pseudo Inverse

The pseudo inverse method is an optimization technique that requires a pseudo inversion of the generally nonsquare **B** matrix. The pseudo inverse solution is the minimum 2-norm solution to the control allocation problem and can be formulated as follows:

$$\min_{\boldsymbol{\delta}} J = \min_{\boldsymbol{\delta}} \frac{1}{2}(\boldsymbol{\delta} + \mathbf{c})^{\mathbf{T}}\mathbf{W}(\boldsymbol{\delta} + \mathbf{c}) \tag{8.17}$$

subject to

$$\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des} \tag{8.18}$$

where $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a weighting matrix and $\mathbf{c} \in \mathbb{R}^n$ is an offset vector used to represent an off-nominal condition with one or more control effectors. To solve this problem, first form the Hamiltonian (H) such that

$$H = \frac{1}{2}\boldsymbol{\delta}^T\mathbf{W}\boldsymbol{\delta} + \frac{1}{2}\mathbf{c}^T\mathbf{W}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T\mathbf{W}\mathbf{c} + \frac{1}{2}\mathbf{c}^T\mathbf{W}\mathbf{c} + \boldsymbol{\xi}(\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}) \tag{8.19}$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$ is an as yet undetermined Lagrange multiplier. Taking the partial derivatives of H with respect to $\boldsymbol{\delta}$ and $\boldsymbol{\xi}$, setting these expressions equal to zero, and rearranging [14], gives the final result:

$$\boldsymbol{\delta} = -\mathbf{c} + \mathbf{W}^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T)^{-1}[\mathbf{d}_{des} + \mathbf{B}\mathbf{c}] = -\mathbf{c} + \mathbf{B}^{\sharp}[\mathbf{d}_{des} + \mathbf{B}\mathbf{c}] \tag{8.20}$$

where $\mathbf{B}^{\sharp}$ is the weighted pseudo inverse of **B**. It should be noted that if an effector is offset (locked and unable to move), two items must be taken into account, position offset ($-\mathbf{c}$), and the moments generated

by the offset (**Bc**). For the position offset, the negative of the locked position is placed in the corresponding entry of the **c** vector. For example, assume that there are four controls and that the third control effector is locked at +5 deg due to a failure. Then, $\mathbf{c} = \begin{bmatrix} 0 & 0 & -5 & 0 \end{bmatrix}^T$.

The weighting matrix, **W**, can be selected to incorporate the position limits of the control effectors. For example, a diagonal element of **W** can be selected to be a function of the corresponding component of **δ**, so that the weighting function approaches ∞ as the control approaches a physical limit. There are no guarantees that commands to the control effectors will not exceed the position limits; however, in practice the method is effective in constraining the positions of the controls. This method can be useful in generating preference vectors for more complex optimization based methods for the purpose of robustness analysis. Like the previous techniques, if **B** is constant, then $\mathbf{B}^\sharp$ can be computed off-line.

8.3.1.4 Pseudo Control

When it is not obvious how to gang the control surfaces, the pseudo control method [5] can be used. This method begins by performing a singular value decomposition [15] (SVD) on **B** such that

$$\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{8.21}$$

Next, the **B** matrix is partitioned such that

$$\mathbf{B} = \begin{bmatrix}\mathbf{U}_{01} & \mathbf{U}_{02} & \mathbf{U}_1\end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_{01} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{02} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{01}^T \\ \mathbf{V}_{02}^T \\ \mathbf{V}_1^T \end{bmatrix} \tag{8.22}$$

where $\boldsymbol{\Sigma}_{01}$ contains the largest singular values of the dimension desired. Now, **B** is approximated using only the largest singular values, that is,

$$\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \approx \mathbf{U}_{01}\boldsymbol{\Sigma}_{01}\mathbf{V}_{01}^T \tag{8.23}$$

Then,

$$\boldsymbol{\delta} = \mathbf{G}\boldsymbol{\delta}_{pseudo} \Rightarrow \mathbf{B}\boldsymbol{\delta} = \mathbf{B}\mathbf{G}\boldsymbol{\delta}_{pseudo} = \widetilde{\mathbf{B}}\boldsymbol{\delta}_{pseudo} \tag{8.24}$$

Letting

$$\begin{aligned} \widetilde{\mathbf{B}} &= \mathbf{U}_{01}\boldsymbol{\Sigma}_{01} \\ \mathbf{G} &= \mathbf{V}_{01} \end{aligned} \tag{8.25}$$

then

$$\mathbf{B}\mathbf{G} \cong \left(\mathbf{U}_{01}\boldsymbol{\Sigma}_{01}\mathbf{V}_{01}^T\right)\left(\mathbf{V}_{01}\right) = \widetilde{\mathbf{B}} \tag{8.26}$$

As with the explicit ganging method, the control allocation goal is to find $\boldsymbol{\delta}_{pseudo}$ such that

$$\widetilde{\mathbf{B}}\boldsymbol{\delta}_{pseudo} = \mathbf{d}_{des} \tag{8.27}$$

After finding $\boldsymbol{\delta}_{pseudo}$, **δ** can be determined using Equation 8.24. The effect is that the surfaces with the most control power in a given axis will be assigned to that axis. Errors that result from the now unmodeled terms, $\mathbf{U}_{02}\boldsymbol{\Sigma}_{02}\mathbf{V}_{02}^T$, will have to be mitigated through feedback. Additionally, if the entries in **B** are constant, the SVD of **B** can be computed off-line and the partitioning of **B** can be determined *a priori*.

8.3.2 Constrained Linear Control Allocation

The only difference between unconstrained and constrained linear control allocation is the inclusion of position and rate limits. In constrained linear control allocation, the goal is to find **δ** such that $\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$ subject to $\boldsymbol{\delta}_{min} \le \boldsymbol{\delta} \le \boldsymbol{\delta}_{max}$, $|\dot{\boldsymbol{\delta}}| \le \dot{\boldsymbol{\delta}}_{max}$.

8.3.2.1 Redistributed Pseudo Inverse

The redistributed pseudo inverse works in a fashion similar to the pseudo inverse, with the addition of position and rate limits. For the redistributed pseudo inverse, the process is iterative and position saturated control effectors are removed from subsequent pseudo inverse solutions. The first step is to solve the control allocation problem using the pseudo inverse solution in Equation 8.20, with **c** initially a vector of all zeros. If no controls exceed their minimum or maximum position limits (recall, rate limits are converted to effective position limits), then the process stops and the solution from Equation 8.20 is used. However, if one or more controls saturate, the problem is solved again, this time zeroing out the columns of the **B** matrix corresponding to the saturated controls and placing the negative of the saturated values in the vector **c**. If any additional control effectors saturate, the corresponding columns of **B** are zeroed out and the problem is solved again. This process continues until all control effectors saturate or a solution is found that does not violate the constraints. One must be cautious here because when saturation occurs, there are two **B** matrices in Equation 8.20, one for the pseudo inverse solution, $\mathbf{W}^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T)^{-1}$, and one for the offset or saturated contribution, **Bc**. When zeroing out a column corresponding to a saturated effector, only the pseudo inverse **B** matrix is modified, while the **Bc** term uses the original **B** matrix.

Consider the following example of a redistributed pseudo inverse control allocation problem: Let

$$\mathbf{B} = \begin{bmatrix} 2 & -2 & -2 & -1 \\ 1 & 1 & -3 & -2 \\ 2 & -2 & -1 & -1 \end{bmatrix} \quad \mathbf{d}_{des} = \begin{bmatrix} 0.5 & 1 & 1 \end{bmatrix}^T \tag{8.28}$$

$$-0.75 \le \delta_1 \le 0.75 \quad -0.75 \le \delta_2 \le 0.75 \quad -0.75 \le \delta_3 \le 0.75 \quad -0.75 \le \delta_4 \le 0.75 \tag{8.29}$$

In this example, there are three controlled variables (3 rows in **B**) and four control effectors (4 columns in **B**). Each control has the same lower and upper position limits and the weighting matrix is $\mathbf{W} = \mathbf{I}$. The first step in the redistributed pseudo inverse solution is to compute $\boldsymbol{\delta}$ using Equation 8.20. The results are

$$\boldsymbol{\delta} = \mathbf{B}^{\sharp}\mathbf{d}_{des} = \begin{bmatrix} 2 & -2 & -2 & -1 \\ 1 & 1 & -3 & -2 \\ 2 & -2 & -1 & -1 \end{bmatrix}^{\sharp} \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.55 \\ 0.23 \\ 0.5 \\ -0.86 \end{bmatrix} \tag{8.30}$$

where $\mathbf{c} = \mathbf{0}$. Since δ_4 exceeds its negative limit, set $\delta_4 = -0.75$ and zero out the fourth column of the **B** matrix. Now, calculate the next pseudo inverse solution using

$$\begin{aligned} \boldsymbol{\delta} &= -\mathbf{c} + \mathbf{B}_{red}^{\sharp}\left[\mathbf{d}_{des} + \mathbf{B}\mathbf{c}\right] \\ &= -\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.75 \end{bmatrix} + \begin{bmatrix} 2 & -2 & -2 & 0 \\ 1 & 1 & -3 & 0 \\ 2 & -2 & -1 & 0 \end{bmatrix}^{\sharp} \left\{ \begin{bmatrix} 0.5 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 2 & -2 & -2 & -1 \\ 1 & 1 & -3 & -2 \\ 2 & -2 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.75 \end{bmatrix} \right\} \\ &= \begin{bmatrix} 0.6875 & 0.3125 & 0.5 & -0.75 \end{bmatrix}^T \end{aligned} \tag{8.31}$$

where $\mathbf{B}_{red}$ is the reduced **B** matrix formed by zeroing out the column corresponding to the saturated control effector. The results show that δ_4 is now at its most negative limit, as expected, and no other control effectors exceed their limit, hence the redistributed pseudo inverse calculations are complete. A check of this result gives

$$\mathbf{B}\begin{bmatrix} 0.6875 & 0.3125 & 0.5 & -0.75 \end{bmatrix}^T = \begin{bmatrix} 0.5 & 1 & 1 \end{bmatrix}^T = \mathbf{d}_{des} \tag{8.32}$$

Thus, the calculated control settings do provide the desired commands. In this example, only two iterations of pseudo inverse calculations were performed. If the control effector setting computed in Equation 8.31

had one or more controls exceeding their position limits, then the process of zeroing out columns in the **B** matrix would continue until all controls are at their limits (either positive or negative) or a feasible solution is found (one where remaining controls do not exceed position limits). As with the pseudo inverse, if the entries in **B** can be approximated as constants, $\mathbf{B}^{\sharp}$ can be computed off-line. Likewise, all possible $\mathbf{B}^{\sharp}_{red}$ matrices can be computed off-line.

8.3.2.2 Daisy Chaining

The daisy chain approach assumes a hierarchy of control effectors. In this method, when one control or a group of controls saturates, there is an error between the commands and the values produced by the control effectors. The daisy chain method would then utilize another control to attempt to reduce the error. Figure 8.3 shows an example of daisy chain allocation. In this example, the goal is to produce a desired pitch acceleration, given by $\dot{q}_{des}$. There are three controls that can produce pitching moment, a elevator (δ_e), a bodyflap (δ_{bf}), and a canard (δ_c). This example solves one equation in the set $\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$. Here, the single equation becomes

$$\begin{bmatrix} M_{\delta_e} & M_{\delta_{bf}} & M_{\delta_c} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_{bf} \\ \delta_c \end{bmatrix} = \dot{q}_{des} \tag{8.33}$$

subject to $\delta_{e_{min}} \leq \delta_e \leq \delta_{e_{max}}$, $\delta_{bf_{min}} \leq \delta_{bf} \leq \delta_{bf_{max}}$, and $\delta_{c_{min}} \leq \delta_c \leq \delta_{c_{max}}$. The control effectiveness entries in the **B** matrix are M_{δ_e}, $M_{\delta_{bf}}$, and M_{δ_c}.

The daisy chain procedure works as follows: the primary control effector, δ_e in this case, is commanded to produce the desired acceleration ($\delta_e = \dot{q}_{des}/M_{\delta_e}$) and the command is constrained by the position and rate limits. If the elevator can produce this acceleration, then the bodyflap and canard are not utilized and the algorithm terminates. However, if there is a moment deficiency between the acceleration that the elevator produces and the desired acceleration, the control effector that is second in line, in this case the bodyflap, is commanded to produce an acceleration equivalent to the acceleration deficiency ($\delta_{bf} = (\dot{q}_{des} - M_{\delta_e}\delta_e)/M_{\delta_{bf}}$). The bodyflap is then constrained by its position and rate limits. If the bodyflap can produce the required acceleration, the canard is not needed and the algorithm terminates. However, if the bodyflap cannot produce the required acceleration, then the canard is commanded to produce the difference between the commanded acceleration and the accelerations produced by the elevator and bodyflap ($\delta_c = (\dot{q}_{des} - M_{\delta_e}\delta_e - M_{\delta_{bf}}\delta_{bf})/M_{\delta_c}$). The canard is then constrained by its position

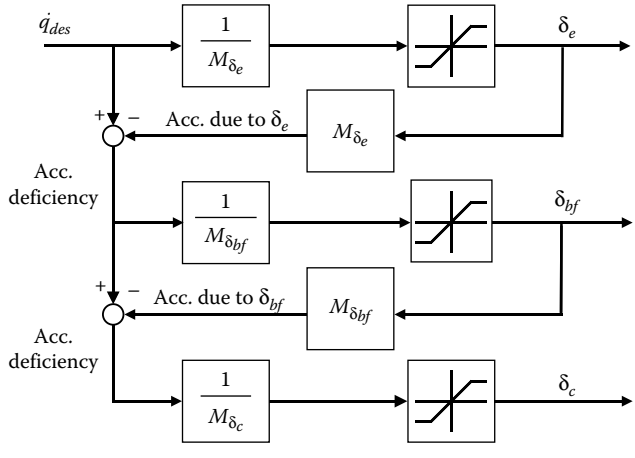

FIGURE 8.3 Example of daisy chain allocation.

and rate limits. Because the canard is the last control in the chain, the process ends, even if an acceleration deficiency still exists.

In the example, three control effectors were used; however, this process can be used for any number of control effectors. A drawback is that one must set up the hierarchy *a priori*. A second drawback is that this method of allocation is most useful when control effectors have authority in only one axis. For control effectors with authority in multiple axes, the daisy chain method could be used, but it is likely that conflicts would exist from the commands for different axes.

8.3.3 Direct Allocation

The direct allocation method, by Durham [6], is a constrained control allocation method aimed at finding a real number, ρ, and a vector, $\boldsymbol{\delta}_1$, such that

$$\mathbf{B}\boldsymbol{\delta}_1 = \rho \mathbf{d}_{des} \tag{8.34}$$

and

$$\boldsymbol{\delta}_{min} \leq \boldsymbol{\delta} \leq \boldsymbol{\delta}_{max} \tag{8.35}$$

If $\rho > 1$, then $\boldsymbol{\delta} = \boldsymbol{\delta}_1/\rho$. If $\rho \leq 1$, then $\boldsymbol{\delta} = \boldsymbol{\delta}_1$. In order to use this method, an attainable moment set (AMS) [16] must be established. The AMS is of the dimension of $\mathbf{d}_{des}$ and for linear systems it consists of a convex hull within moment or controlled variable space defined by planar surfaces that correspond to one or more of the control effectors set at a position limit. Physically, ρ represents how much a control power demand must be scaled in order to touch the boundary of the AMS. When $\rho \leq 1$, the moment demand lies within the AMS and the allocator supplies the demand. When $\rho > 1$, the control power demand exceeds supply and the demand is scaled back to touch the boundary of the AMS, while preserving the direction of $\mathbf{d}_{des}$. Algorithms exist for generating the AMS for the three moment problem [16].

8.3.4 Linear and Quadratic Programming Optimization Methods

8.3.4.1 Error and Control Minimization

The objective of the error minimization problem is to find a vector $\boldsymbol{\delta}$, given $\mathbf{B}$ and $\mathbf{d}_{des}$, such that

$$J = \|\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}\|_p \tag{8.36}$$

is minimized, subject to

$$\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \overline{\boldsymbol{\delta}} \tag{8.37}$$

The norm depends on the type of algorithm used to perform the minimization. When used with LP solvers, the error minimization problem is specified as

$$\min_{\boldsymbol{\delta}} J = \|\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}\|_1 \tag{8.38}$$

subject to the constraints in Equation 8.37. This can be transformed into the standard LP problem [10]

$$\min_{\boldsymbol{\delta}_s} J = \begin{bmatrix} 0 & \dots & 0 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\delta}_s \end{bmatrix} \tag{8.39}$$

subject to

$$\begin{bmatrix} \boldsymbol{\delta}_s \\ -\boldsymbol{\delta} \\ \boldsymbol{\delta} \\ -\mathbf{B}\boldsymbol{\delta} + \boldsymbol{\delta}_s \\ \mathbf{B}\boldsymbol{\delta} + \boldsymbol{\delta}_s \end{bmatrix} \geq \begin{bmatrix} \mathbf{0} \\ -\overline{\boldsymbol{\delta}} \\ \underline{\boldsymbol{\delta}} \\ -\mathbf{d}_{des} \\ \mathbf{d}_{des} \end{bmatrix} \tag{8.40}$$

where $\boldsymbol{\delta}_s \in \mathbb{R}^m$ is a vector of slack variables. Note that the slack variables must be positive, but are otherwise unconstrained. Individually, the slack variables represent how much control power demand

exceeds supply in any given axis. If $J = 0$, then the control law command is feasible, otherwise it is infeasible and the control effectors cannot meet the demands. Weights may be added to the control deficiency problem to provide greater flexibility. In this case, the error minimization problem becomes

$$\min_{\boldsymbol{\delta}_s} J = \begin{bmatrix} 0 & \dots & 0 & \mathbf{W}_d^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\delta}_s \end{bmatrix} \tag{8.41}$$

subject to Equation 8.40. In Equation 8.41, $\mathbf{W}_d^T \in \mathbb{R}^{m\times 1}$ is the weighting vector. The weights allow one to penalize control power deficiencies in some axes greater than others, for example, one could penalize the axis with the least open-loop stability. Either of the problems specified in Equation 8.39 or 8.41 can be solved with a LP solver. The solutions to these problems minimizes the unweighted (Equation 8.39) or weighted (Equation 8.41) 1-norm of the distance between $\mathbf{B}\boldsymbol{\delta}$ and $\mathbf{d}_{des}$.

The control minimization problem is a secondary optimization. If there exists sufficient control authority to satisfy Equation 8.38 such that $J = 0$, then multiple solutions may exist and a secondary objective may be achieved. The ability to do this is a direct result of the over-actuated system, in that multiple solutions to the problem may exist and one solution may be preferred over another. The control minimization is posed as follows:

$$\min_{\boldsymbol{\delta}_s} J = \begin{bmatrix} 0 & \dots & 0 & \mathbf{W}_u^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\delta}_s \end{bmatrix} \tag{8.42}$$

subject to

$$\begin{bmatrix} \boldsymbol{\delta}_s \\ -\boldsymbol{\delta} \\ \boldsymbol{\delta} \\ -\boldsymbol{\delta} + \boldsymbol{\delta}_s \\ \boldsymbol{\delta} + \boldsymbol{\delta}_s \end{bmatrix} \geq \begin{bmatrix} \mathbf{0} \\ -\overline{\boldsymbol{\delta}} \\ \underline{\boldsymbol{\delta}} \\ -\boldsymbol{\delta}_p \\ \boldsymbol{\delta}_p \end{bmatrix} \qquad \mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des} \tag{8.43}$$

where $\mathbf{W}_u^T \in \mathbb{R}^{n\times 1}$, $\boldsymbol{\delta}_s \in \mathbb{R}^n$, and $\boldsymbol{\delta}_p \in \mathbb{R}^n$ is the preferred control effector position vector. The first requirement is that the demand be satisfied (error minimization problem), followed by selecting the control effector positions that satisfy the demand and minimize a secondary objective (control minimization problem). Many secondary objectives could be specified, for example, minimum control deflection, minimum wing loading, minimum drag, minimum actuator power, and so on. A few of these secondary objectives are discussed in the next sections.

8.3.4.2 Minimum Control Deflection

The minimum control deflection is one of the simplest objectives and for aircraft applications. It roughly approximates the objective of minimizing drag and actuator deflections. For minimum control deflection, $\mathbf{W}_u^T$ and $\boldsymbol{\delta}_p$ in Equations 8.42 and 8.43 become

$$\mathbf{W}_u^T = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \quad \boldsymbol{\delta}_p = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}^T \tag{8.44}$$

Here, the objective is to drive the control effectors to their zero positions and to drive each one toward zero with equal weight.

8.3.4.3 Minimum Wing Loading

In this case, a rough approximation is used to achieve this objective. Outboard aerodynamic surfaces tend to produce higher wing root bending moments as compared to inboard surfaces. To achieve an

approximation to minimum wing loading, set

$$\boldsymbol{\delta}_p = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}^T \tag{8.45}$$

and set the components of $\mathbf{W}_u^T$ corresponding to outboard surfaces to larger weights than those corresponding to inboard surfaces. For example, let

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_{e_{in_R}} & \delta_{e_{in_L}} & \delta_{e_{out_R}} & \delta_{e_{out_L}} \end{bmatrix} \tag{8.46}$$

where $\delta_{e_{in_R}}$ is the right inboard elevon, $\delta_{e_{in_L}}$ is the left inboard elevon, $\delta_{e_{out_R}}$ is the right outboard elevon, and $\delta_{e_{out_L}}$ is the left outboard elevon. In this case, the weighting vector, to achieve minimum wing loading, could be set to

$$\mathbf{W}_u^T = \begin{bmatrix} 1 & 1 & 1000 & 1000 \end{bmatrix} \tag{8.47}$$

so that more penalty is applied to deflection of the outboard surfaces.

To obtain the actual minimum wing loading or minimum drag solutions, an aerodynamic model would be required to determine $\mathbf{W}_u^T$ and $\boldsymbol{\delta}_p$. This onboard model would be interrogated to determine the forces and moments due to each control effector and the preference vector would be selected to achieve minimum wing loading or drag.

8.3.4.4 Minimum 2-Norm for Robustness Analysis

One of the drawbacks of optimization based LP methods is that it is impossible to represent this allocator in a form that is amenable for use with conventional robustness analysis tools. In the pseudo inverse development, a direct relationship exists between inputs ($\mathbf{d}_{des}$) and outputs ($\boldsymbol{\delta}$). When $\mathbf{c} = \mathbf{0}$ and $\mathbf{W} = \mathbf{I}$, Equation 8.20 becomes

$$\boldsymbol{\delta} = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{d}_{des} = \mathbf{B}^\sharp \mathbf{d}_{des} \tag{8.48}$$

In this case, the control allocator is simply a gain matrix given by $\mathbf{B}^\sharp$. This is useful for stability or robustness analysis because a model of the control allocator, namely $\mathbf{B}^\sharp$, has been determined. When LP control allocators are utilized, no such model exists and the relationship between inputs and outputs is more complicated. In this case, determining the stability or robustness of closed loop systems is difficult. Fortunately, if no control effectors exceed their rate and position limits, then equating the preference vector in Equation 8.43 to the 2-norm (or pseudo inverse) solution, given by Equation 8.48, provides a method to model the allocator. The solution to the control allocation problem will be the 2-norm solution and hence, the LP control allocator block can be replaced by $\mathbf{B}^\sharp$ for use in a stability or robustness analysis. When control effectors exceed rate or position limits, this model is no longer valid. Note that this technique can also be used in the mixed optimization problem described in the next section.

8.3.4.5 Mixed Optimization

The mixed optimization problem [13] combines the control and error minimization problems into a single cost function. The mixed optimization problem is posed as follows: given $\mathbf{B}$, $\mathbf{d}_{des}$, and a preferred control vector $\boldsymbol{\delta}_p$, find a vector $\boldsymbol{\delta}$ such that

$$J = \|\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}\| + \nu\|\boldsymbol{\delta} - \boldsymbol{\delta}_p\| \tag{8.49}$$

is minimized, subject to $\underline{\boldsymbol{\delta}} \le \boldsymbol{\delta} \le \overline{\boldsymbol{\delta}}$ where $\nu \in \mathbb{R}^1$ is a factor which weights the relative importance of the error and control minimization problems. If ν is small (which is typically the case), priority is given to error minimization over control minimization.

8.3.4.6 Conversion of Mixed Optimization Problem to a LP Problem

Both Buffington [10] and Bodson [13] discuss the conversion of the mixed optimization problem to a LP problem. Buffington [10] performed the conversion using slack variables. While this approach is mathematically correct, it increases the number of inequality constraints, thus leading to a more computationally expensive linear program. Here, the work of Bodson [13] is used to perform the conversion.

A standard LP problem is to find a vector $\boldsymbol{\delta}$ such that

$$J = \mathbf{m}^T \boldsymbol{\delta} \tag{8.50}$$

is minimized, subject to

$$\begin{aligned} \mathbf{0} \leq \boldsymbol{\delta} \leq \mathbf{h} \\ \mathbf{A}\boldsymbol{\delta} = \mathbf{b} \end{aligned} \tag{8.51}$$

The details of the conversion are covered in Bodson [13] and only the results are presented here. Defining the vector $\mathbf{x}^T = (\mathbf{e}^+ \ \mathbf{e}^- \ \boldsymbol{\delta}^+ \ \boldsymbol{\delta}^-)$, the LP problem is

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{I} & -\mathbf{I} & -\mathbf{B} & \mathbf{B} \end{bmatrix} \\ \mathbf{b} &= \mathbf{B}\boldsymbol{\delta}_p - \mathbf{d}_{des} \\ \mathbf{m}^T &= \begin{bmatrix} 1 & \dots & 1 & \nu & \dots & \nu \end{bmatrix} \\ \mathbf{h}^T &= \begin{bmatrix} \mathbf{e}_{max} & \mathbf{e}_{max} & (\boldsymbol{\delta}_{max} - \boldsymbol{\delta}_p) & (\boldsymbol{\delta}_p - \boldsymbol{\delta}_{min}) \end{bmatrix} \end{aligned} \tag{8.52}$$

where $\mathbf{e} = \mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}$ is the error, $\mathbf{e}_{max}$ is some upper bound on the achievable error (e.g., the 1-norm of the error),

$$\boldsymbol{\delta}^+ = \begin{cases} \boldsymbol{\delta} - \boldsymbol{\delta}_p & \text{if } \boldsymbol{\delta} - \boldsymbol{\delta}_p > 0 \\ \mathbf{0} & \text{if } \boldsymbol{\delta} - \boldsymbol{\delta}_p \leq 0 \end{cases} \qquad \boldsymbol{\delta}^- = \begin{cases} -\boldsymbol{\delta}_p + \boldsymbol{\delta} & \text{if } \boldsymbol{\delta}_p - \boldsymbol{\delta} > 0 \\ \mathbf{0} & \text{if } \boldsymbol{\delta}_p - \boldsymbol{\delta} \leq 0 \end{cases} \tag{8.53}$$

and

$$\mathbf{e}^+ = \begin{cases} \mathbf{e} & \text{if } \mathbf{e} > 0 \\ \mathbf{0} & \text{if } \mathbf{e} \leq 0 \end{cases} \qquad \mathbf{e}^- = \begin{cases} -\mathbf{e} & \text{if } -\mathbf{e} > 0 \\ \mathbf{0} & \text{if } -\mathbf{e} \leq 0 \end{cases} \tag{8.54}$$

This problem can then be solved for the control deflection vector $\boldsymbol{\delta}$ using a standard LP solver.

8.3.5 Solving the LP Problem

A method used to solve LP problems is the simplex algorithm [17]. This algorithm has a highly desirable quality in that it is guaranteed to find an optimal solution in a finite period of time. Generally speaking, the simplex method moves from one basic feasible solution of the constraint set to another, in such a way that the value of the objective function is continually decreased until a minimum is reached. The number of iterations of the simplex algorithm to find a solution is, assuming no basic feasible solution is repeated, at most $\frac{n!}{m![(n-m)!]}$ where m is the number of controlled variables (3 for aircraft rotational motion) and $n = 2p + 6$ where p is the number of control effectors. Anticycling [18] can be used to ensure that the same basic feasible solutions are not encountered more than once.

8.3.6 Quadratic Programming

In the optimization problems previously discussed, the l_1 norm was used, except for the pseudo inverse problems, where the l_2 norm was used. In a quadratic formulation [7,13] of the control allocation problem, the l_2 norm is used and the objective is to find a control vector, $\boldsymbol{\delta}$, that minimizes

$$J = \|\mathbf{B}\boldsymbol{\delta} - \mathbf{d}_{des}\|_2^2 \tag{8.55}$$

The solution to this is given by the pseudo inverse solution, Equation 8.20, with $\mathbf{c} = \mathbf{0}$ and $\mathbf{W} = \mathbf{I}$. If the computed control effector vector lies within the constraints, then the algorithm stops. However, if one

or more controls exceeds a limit, the method continues, but in a fashion different than the redistributed pseudo inverse. The solution to Equation 8.55, subject to the equality constraint

$$\|\boldsymbol{\delta}\|_2^2 = p \tag{8.56}$$

is computed. For $\boldsymbol{\delta} \in \mathbb{R}^2$, the constraint in Equation 8.56 is an ellipse where p is chosen just large enough that the ellipse encloses the rectangle formed by the typical box constraints $\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \overline{\boldsymbol{\delta}}$. The solution to Equation 8.55 with equality constraints (Equation 8.56) is

$$\boldsymbol{\delta} = \mathbf{B}^T \left[\mathbf{B}\mathbf{B}^T + \lambda \mathbf{I}\right]^{-1} \mathbf{d}_{des} \tag{8.57}$$

This result looks similar to the pseudo inverse solution; however, in this case, λ is a Lagrange multiplier that can be found by solving

$$\gamma(\lambda) = \|\mathbf{B}^T \left[\mathbf{B}\mathbf{B}^T + \lambda \mathbf{I}\right]^{-1} \mathbf{d}_{des}\|_2^2 = p \tag{8.58}$$

This equation can be solved with a few iterations of the bisection method [7] and the control vector, $\boldsymbol{\delta}$, is then clipped to satisfy the constraints.

If the constraints are not active and $\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$ can be satisfied exactly, then the linear and quadratic programming approaches will yield the same results when the number of controls (n) equals the number of axes to control (m). Even when $m \neq n$, the results should be the same when the constraints are not active. The results for the redundant actuator cases will most likely be different when the constraints are active. The difference between these two approaches is difficult to quantify on a general level and is best suited to quantification on an application specific problem. Hence, application of both approaches is required to determine if one outperforms the other. Page and Steinberg [19] have compared numerous control allocation approaches. Their data is useful in determining a control allocation technique to utilize for a specific application.

8.4 Control Interactions

As the number of control effectors placed on a vehicle increases, the likelihood of the occurrence of control effector interactions increases. A control interaction is when the use of one control effector alters the force and moment producing capabilities of another. Typically, the effect of one control effector on another is ignored in the control allocation problem. In many applications, these interaction effects are small as compared to the forces and moments generated by each control effector acting individually. However, cases exist where the interactions should not be ignored. In aircraft applications, the aerodynamic surfaces are used to modulate forces and moments. Instances where control surfaces lie in close proximity to other surfaces or cases where one control surface lies downstream of another are examples of situations where the effectiveness of some surfaces are influenced by the deflection of other surfaces. In automotive applications, there exists significant coupling between the yaw and axial acceleration effectiveness of the steering angle and front wheel braking forces [20]. Another example can be found on space launch vehicles that can use a combination of differential throttles and gimballed nozzles for attitude control. The effectiveness of gimballed nozzles are influenced by the engine thrust. Hence, a control allocation method that can include the coupling effects of multiple control effectors, is desirable.

As an example of control interactions, assume the pitching moment is the sum of the base or wing-body pitching moment, the incremental pitching moments produced by each control effector taken one at a time, and the incremental pitching moments caused by any interactions between control surfaces. Then,

the pitching moment coefficient becomes

$$C_m = C_{m_{Base}}(\alpha, \beta, M) + \sum_{i=1}^{n} \Delta C_{m_{\delta_i}}(\alpha, \beta, M, \delta_i) + \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} \Delta C_{m_{\delta_i, \delta_j}}(\alpha, \beta, M, \delta_i, \delta_j) \tag{8.59}$$

where α is the angle of attack, β is the angle of sideslip, M is the Mach number, and δ_i is the position of the ith control effector. The last term in Equation 8.59 accounts for effector interactions. That is, a combined deflection cause forces and moments in addition to the forces and moments produced by the individual controls. Interactions do not fit into a linear control allocation scheme since, by definition, they are nonseparable nonlinear functions of two control deflections (a separable function of multiple variables is a function that can be specified as a sum of functions of a single variable). In a number of important cases, the interactions can be described by a bilinearity of the form $\Delta C_m(\delta_i, \delta_j) = (\partial^2 C_m / \partial \delta_i \partial \delta_j)\ \delta_i \delta_j$. In such cases, one can pose a control allocation problem similar to that in the strict linear case, except that the control effectiveness matrix, $\mathbf{B}$, is replaced by a control dependent matrix, $\mathbf{A}(\boldsymbol{\delta})$ [21]. The control allocation problem becomes

$$\min_{\boldsymbol{\delta}} \|\mathbf{A}(\boldsymbol{\delta})\boldsymbol{\delta} - \mathbf{d}_{des}\|_1 + \nu \|\mathbf{W}_{\boldsymbol{\delta}}(\boldsymbol{\delta} - \boldsymbol{\delta}_p)\|_1 \tag{8.60}$$

subject to $\underline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \overline{\boldsymbol{\delta}}$. For a three moment aircraft application, $\mathbf{A}(\boldsymbol{\delta})$ in Equation 8.60 is given by

$$\mathbf{A}(\boldsymbol{\delta}) \triangleq \left\{ \frac{1}{2} \begin{bmatrix} \boldsymbol{\delta}^T \mathbf{Q}_L \\ \boldsymbol{\delta}^T \mathbf{Q}_M \\ \boldsymbol{\delta}^T \mathbf{Q}_N \end{bmatrix} + \mathbf{B} \right\} \tag{8.61}$$

The elements of the matrices $\mathbf{Q}_L, \mathbf{Q}_M, \mathbf{Q}_N$ represent the sensitivity of the rolling, pitching, and yawing moments due to the combined actuation of two control surfaces. More specifically, $\mathbf{Q}_L$ (L is the rolling moment) can be expressed as

$$\mathbf{Q}_L = \begin{bmatrix} \frac{\partial^2 L}{\partial \delta_1^2} & \frac{\partial^2 L}{\partial \delta_1 \delta_2} & \cdots & \frac{\partial^2 L}{\partial \delta_1 \delta_n} \\ \frac{\partial^2 L}{\partial \delta_2 \delta_1} & \frac{\partial^2 L}{\partial \delta_2^2} & \cdots & \frac{\partial^2 L}{\partial \delta_2 \delta_n} \\ & \vdots & & \\ \frac{\partial^2 L}{\partial \delta_n \delta_1} & \frac{\partial^2 L}{\partial \delta_n \delta_2} & \cdots & \frac{\partial^2 L}{\partial \delta_n^2} \end{bmatrix} \tag{8.62}$$

The matrices $\mathbf{Q}_M$ and $\mathbf{Q}_N$ are similar to Equation 8.62. For the case where bilinear interaction terms are the only nonlinearities of interest, the main diagonal terms will be zero. However, this form can also accommodate cases where forces or moments are separable quadratic functions of individual control deflections, for example, individual aileron or flap contributions to yawing moment at low angles of attack [22].

Because the matrix $\mathbf{A}(\boldsymbol{\delta})$ itself is a function of $\boldsymbol{\delta}$, the control allocation problem is, strictly speaking, nonlinear. Rather than directly applying nonlinear programming techniques, this problem can be solved sequentially by solving a series of LP subproblems (each with guaranteed convergence properties) in order to progressively improve approximations to the solution of the original nonlinear programming problem. Define the control deflection vector, $\boldsymbol{\delta}_k$, as the most recent solution to an LP subproblem that is computed as part of the iterative procedure. Now pose the following LP subproblem whose solution yields an updated estimate, $\boldsymbol{\delta}_{k+1}$, of the control deflection vector that solves the original

nonlinear programming problem:

$$\min_{\boldsymbol{\delta}_{s1},\boldsymbol{\delta}_{s2}} \begin{bmatrix} 0 & 0 & \dots & 0 & w_1 & w_2 & \dots & w_n & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_{k+1} \\ \boldsymbol{\delta}_{s1} \\ \boldsymbol{\delta}_{s2} \end{bmatrix} \tag{8.63}$$

subject to

$$\begin{bmatrix} \boldsymbol{\delta}_{s1} \\ \boldsymbol{\delta}_{s2} \\ \boldsymbol{\delta}_{k+1} \\ -\boldsymbol{\delta}_{k+1} \\ -\mathbf{A}(\boldsymbol{\delta}_k)\boldsymbol{\delta}_{k+1} + \boldsymbol{\delta}_{s2} \\ \mathbf{A}(\boldsymbol{\delta}_k)\boldsymbol{\delta}_{k+1} + \boldsymbol{\delta}_{s2} \\ -\boldsymbol{\delta}_{k+1} + \boldsymbol{\delta}_{s1} \\ \boldsymbol{\delta}_{k+1} + \boldsymbol{\delta}_{s1} \end{bmatrix} \geq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \underline{\boldsymbol{\delta}} \\ -\overline{\boldsymbol{\delta}} \\ -\mathbf{d}_{des} \\ \mathbf{d}_{des} \\ -\boldsymbol{\delta}_p \\ \boldsymbol{\delta}_p \end{bmatrix} \tag{8.64}$$

where $\boldsymbol{\delta}_{s1} \in \mathbb{R}^n, \boldsymbol{\delta}_{s2} \in \mathbb{R}^m$ are slack variables. The LP subproblems are solved until either the flight control system requires a solution at the end of a control update frame or until the following convergence criteria is satisfied:

$$\frac{|\boldsymbol{\delta}_{k+1} - \boldsymbol{\delta}_k|}{|\boldsymbol{\delta}_k|} \leq \text{tol} \tag{8.65}$$

which indicates that further iterations will not result in significant improvements in the solution estimate. The variable "tol" in Equation 8.65 is a user-defined tolerance for convergence.

8.5 Effect of Actuator Dynamics on the Performance of Constrained Control Allocation Algorithms

A review of the constrained control allocation literature shows that the coupling effects that result from combining constrained control allocators and actuator dynamics has largely been ignored, although some recent research has been performed on directly including actuator dynamics in the control allocation problem [12,23]. The underlying assumption of most previous work is that actuators respond instantaneously to commands. This assumption may at first seem justified because in practice actuator dynamics are typically much faster than the rigid body modes that are to be controlled. However, interactions between a constrained control allocator and actuator dynamics can result in a system whose performance falls well short of its potential. These interactions can significantly reduce the effective rate limits of the system while at the same time yielding control effector positions which are different than the commanded positions [24].

As an example of some of the problems that can be encountered in this situation, a simulation was run with a LP based control allocation algorithm mixing four control effectors to obtain a desired set of moments, $\mathbf{d}_{des} \in \mathbb{R}^3$. In this simulation, the actuator dynamics for each control effector were set to $\delta(s)/\delta_{cmd}(s) = 5/(s+5)$ and the commanded effector positions, as computed by the control allocator, were used to initialize the allocator at the next timestep. To illustrate the effects of constrained control allocator and actuator dynamics interactions, consider Figure 8.4. The objective is to make the accelerations produced by the control effectors ($\mathbf{B}\boldsymbol{\delta}$) equal to the commanded accelerations ($\mathbf{d}_{des}$). When there are no actuator dynamics, the desired result is achieved, namely $\mathbf{B}\boldsymbol{\delta} = \mathbf{d}_{des}$. When actuator dynamics are included, the results are as shown in Figure 8.4, where it is obvious that $\mathbf{B}\boldsymbol{\delta} \neq \mathbf{d}_{des}$. A method will now be

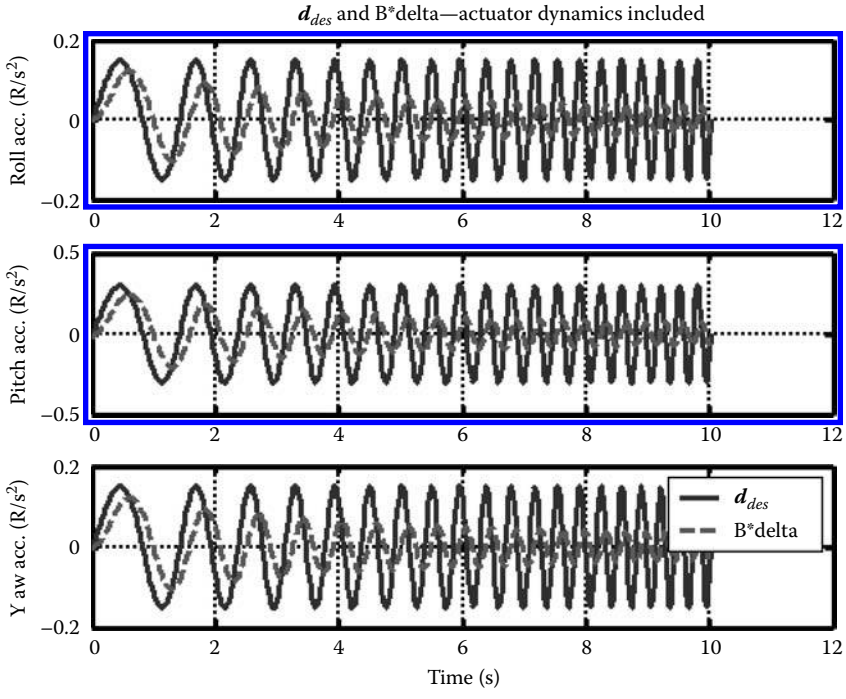

FIGURE 8.4 Acceleration commands and accelerations produced by the controls—actuator dynamics present.

introduced that compensates for the effects of actuator dynamics, so that even when actuator dynamics are present, the control effector produced accelerations equal the commanded accelerations.

The method developed postprocesses the output of a control allocation algorithm to compensate for actuator dynamics. In reaching this objective, it is of great importance to consider the real-world applicability of the approach. In order to obtain an algorithm that can operate in real time on a typical flight computer, it is pertinent not to add to the complexity of the constrained control allocation algorithm. Although the results in Härkegård [12] and Venkataraman et al. [23] overcome some of the issues raised here, it is not certain that either method can guarantee convergence in one flight control system update for both the saturated and unsaturated cases. One method which can accomplish the goal is to postprocess the output of the control allocation algorithm to amplify the commands to the actuators so that at the end of a sampling interval, the actual actuator positions are equivalent to the desired actuator positions. This is the approach taken here and it yields a simple, but effective, means of compensating for actuator dynamics.

Bolling [24] has shown that the interaction between first-order actuator dynamics and constrained control allocation algorithms can be eliminated by overdriving the actuators, that is, commanding more from the actuator so that its actual output is the desired value. In the following sections, the details of the interaction between constrained control allocators and first-order actuator dynamics are developed and a procedure to reduce the adverse affects is provided.

Figure 8.5 shows the system to be analyzed. Inputs to the control allocation algorithm consist of a vector of commands, $\mathbf{d}_{des} \in \mathbb{R}^n$, and a vector containing the current control surface deflections, $\boldsymbol{\delta} \in \mathbb{R}^m$. The output of the control allocator is the commanded control surface deflection vector, $\boldsymbol{\delta}_{cmd} \in \mathbb{R}^m$. The actuator dynamics respond to $\boldsymbol{\delta}_{cmd}$ to produce the actual control deflections, $\boldsymbol{\delta}$. The individual actuators are assumed to have uncoupled dynamics, hardware rate limits of $\dot{\boldsymbol{\delta}}_{max}$, and position limits $\boldsymbol{\delta}_{min}$, $\boldsymbol{\delta}_{max}$. As before, rate limits are taken into account by converting them into effective position limits at the end of the next sampling period (see Equation 8.9) and constraining the effector commands with respect to the

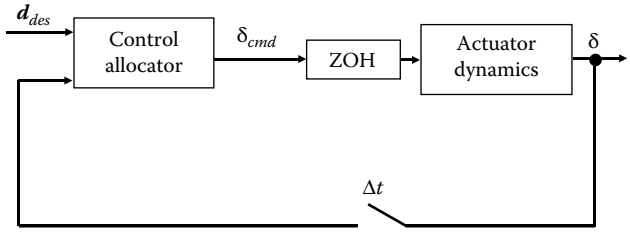

FIGURE 8.5 Control allocator and actuator interconnection.

most restrictive rate or position limits, that is,

$$\overline{\boldsymbol{\delta}} = \min(\boldsymbol{\delta}_{max}, \boldsymbol{\delta} + \dot{\boldsymbol{\delta}}_{max_{CA}} \Delta t) \; : \; \underline{\boldsymbol{\delta}} = \max(\boldsymbol{\delta}_{min}, \boldsymbol{\delta} - \dot{\boldsymbol{\delta}}_{max_{CA}} \Delta t) \tag{8.66}$$

The difference between Equations 8.66 and 8.9 is that in Equation 8.9, the rate limit is the hardware limit, $\dot{\boldsymbol{\delta}}_{max}$, whereas here, the rate limit is $\dot{\boldsymbol{\delta}}_{max_{CA}}$, the software limit. In typical implementations, each element of the vector of rate limits provided to the control allocation algorithm, $\dot{\boldsymbol{\delta}}_{max_{CA}}$, is taken to be the true hardware rate limit of the corresponding actuator. Hence, the software rate limit is equal to the hardware rate limit or $\dot{\boldsymbol{\delta}}_{max_{CA}} = \dot{\boldsymbol{\delta}}_{max}$; however, as was shown by Bolling [24], the effective rate limit of a scalar system composed of a constrained control allocation algorithm and first-order actuator dynamics of the form

$$\frac{\delta(s)}{\delta_{cmd}(s)} = \frac{a}{s+a} \tag{8.67}$$

becomes

$$\dot{\delta}_{max_{EFF}} = \Gamma \dot{\delta}_{max_{CA}} \tag{8.68}$$

where $\Gamma \triangleq 1 - e^{-a\Delta t}$ and $\dot{\delta}_{max_{EFF}}$ is the effective rate limit. For example, when $a = 20\,\text{rad/s}$, $\dot{\delta}_{max_{CA}} = 60^\circ/\text{s}$, and the flight control system operates at 50 Hz ($\Delta t = 0.02$ s), the effective rate limit is $19.78^\circ/\text{s}$. This result is well short of the ideal $60^\circ/\text{s}$ value.

A few comments are in order regarding Figure 8.5 and the analysis described above. First, note that the instantaneous position limit given by Equation 8.66 makes use of a sampled vector of actuator position measurements. This is in contrast to using the previous value of the actuator command vector, $\boldsymbol{\delta}_{cmd}$, as is often done in simulation. The motivation for using actuator measurements is that when actuator dynamics, disturbances, and uncertainties are taken into account, the actuator command vector and the true actuator positions will differ. This difference can cause a control allocator to generate inappropriate actuator commands that do not deliver the desired moments or accelerations. Thus, providing the measured actuator position vector to the control allocator has the advantage of reducing uncertainty in the actuator position.

In the same manner in which rate limits are effectively attenuated by actuator dynamics, so are the magnitudes of the commanded changes to control effector positions. Referring to Figure 8.5, the desired situation would be for $\boldsymbol{\delta} = \boldsymbol{\delta}_{cmd}$. However, actuator dynamics alter the command signals so that, in general, $\boldsymbol{\delta} \neq \boldsymbol{\delta}_{cmd}$. For actuators with high bandwidths relative to the rigid-body modes, this is not a serious concern. However, situations exist where the actuator dynamics are not sufficiently fast and need to be taken into account. Therefore, a method that compensates for both magnitude and rate limit attenuation will be described.

The actuator command signal must be modified such that

$$\tilde{\delta}_{cmd}(t_k) = M \Delta \delta_{cmd_{CA}}(t_k) + \delta(t_k) = \frac{1}{\Gamma_s} \Delta \delta_{cmd_{CA}}(t_k) + \delta(t_k) \tag{8.69}$$

where $M = \frac{1}{\Gamma_s}$, $\Gamma_s = 1 - e^{-a_{nom}\Delta t}$ is the software scaling factor and a_{nom} is the nominal bandwidth of the first-order actuators. The commanded incremental change in actuator position over one timestep is

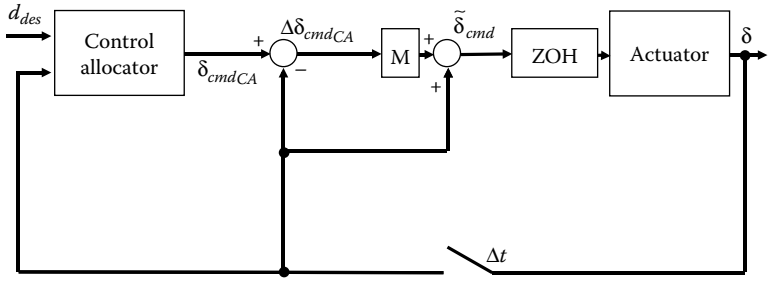

FIGURE 8.6 Block diagram of command increment compensation.

defined by $\Delta\delta_{cmd_{CA}}(t_k) \triangleq \delta_{cmd_{CA}}(t_k) - \delta(t_k)$ and $\delta_{cmd_{CA}}(t_k)$ is the actuator position command from the control allocator. The distinction between Γ and Γ_s is made because it is possible for the actuators to have a bandwidth that is less than the nominal value due to power loss, partial failure, and so on. In other words, the potential exists for $a < a_{nom}$. In this analysis, it has been assumed that a_{nom} is the nominal bandwidth of the actuators and is an upper bound on bandwidth. That is, if $a \neq a_{nom}$, then $a < a_{nom}$. Therefore, bandwidth of the actuator cannot be larger than the nominal bandwidth. Under this assumption, $\Gamma \leq \Gamma_s$.

Since Γ_s can be computed from the known quantities a_{nom} and Δt, one can compensate for command increment attenuation using Equation 8.69. For a bank of decoupled first-order actuators with nominal bandwidths of a_{nom_i}, corresponding values of Γ_{s_i} can be computed using $\Gamma_{s_i} = (1 - e^{-a_{nom_i}\Delta t})$. The command increment compensation can then be implemented in discrete time as shown in the block diagram of Figure 8.6. For multiple actuators, **M** in Figure 8.6 is a diagonal matrix with the entries on the main diagonal being $\Gamma_{s_1}, \Gamma_{s_2}, \ldots, \Gamma_{s_n}$, where the subscript n is defined as the number of control effectors. Hence, the magnitude of the control allocation command increment is modified to counteract the attenuation that results from the interaction between first-order actuator dynamics and the control allocator. The allocator continues to operate under the assumption that the actuator will respond instantaneously and ensures that the commands obey rate and position limits. It is after the allocator computes a new set of commands that the command increment is scaled and added to the measured actuator position. A beneficial consequence of this is that the effective rate limit will be equal to the hardware rate limit when the magnitude increment is scaled in this fashion. Thus, when using this technique, there is no need to adjust the software rate limit.

It should be pointed out that this technique effectively modifies the gain of the inner-loop. In the typical situation, where actuator dynamics are assumed to be much faster than the rigid body modes to control and are hence ignored, a stability analysis would use δ_{cmd} as the input to the plant. In this case, it can be shown that for $\Gamma \leq \Gamma_s$, even after this compensation scheme is applied, $\delta \leq \delta_{cmd_{CA}}$. Therefore, something less than or equal to $\delta_{cmd_{CA}}$ is applied to the plant and the loop gain is either unchanged or reduced. The performance of the entire control system will be burdened with the task of mitigating the effects of $\delta \neq \delta_{cmd_{CA}}$. A similar procedure can be applied to second-order actuator dynamics with and without a simple zero [25,26].

8.6 Nonlinear Control Allocation

A linear control allocation problem, as described in Equations. 8.7 and 8.8, suffers from the fact that it is assumed that the individual entries in the control effectiveness matrix (slopes of controlled variable–deflection data) are linear with respect to the control variables. Additionally, it is also assumed that the controlled variable–deflection relationships pass through the origin. In a local sense, this is typically not

the case and it is directly attributable to nonlinear effects in the controlled variable–deflection relationship. In this section, nonlinear control allocation techniques will be discussed.

8.6.1 Affine Control Allocation

A more accurate solution to the control allocation problem, as compared to the linearity assumption, can be obtained using an affine control allocation problem formulation [11], that is, one of the form: find $\boldsymbol{\delta}$ such that

$$\mathbf{B}\boldsymbol{\delta} + \boldsymbol{\epsilon}(\boldsymbol{\delta}) = \mathbf{d}_{des} \tag{8.70}$$

is minimized, subject to

$$\overline{\boldsymbol{\delta}} \leq \boldsymbol{\delta} \leq \underline{\boldsymbol{\delta}} \tag{8.71}$$

where $\boldsymbol{\epsilon}(\boldsymbol{\delta})$ is an intercept term that provides a more robust control allocation algorithm when the controlled variable–deflection curves are not linear. Clearly, when $\boldsymbol{\epsilon}(\boldsymbol{\delta}) \neq 0$, this allocation scheme is not strictly linear, however, it is not what most researchers would consider nonlinear allocation. Equation 8.70 is called an affine control allocation problem. This technique is suitable for controlled variable–deflection relationships that are monotonic but possibly nonlinear. Mild slope reversals can be accommodated on a case-by-case basis. The advantages of this technique are that some nonlinearities can be handled and it allows the use of the simplex algorithm when posed as a LP problem. True nonlinear programming algorithms can handle more severe nonlinearities in the data, however, they provide no guarantee of convergence. Hence, it is difficult to apply nonlinear programming techniques to flight critical systems.

Figure 8.7 illustrates the issue with linear versus affine control allocation problems. Here, a one-dimensional example is utilized; however, the affine control allocation formulation also applies to multidimensional problems. The horizontal axis is the control effector position, δ, while the vertical axis is the value (in controlled variable units) produced by the control effector, $g(\delta)$. When the control effector is operating at the position given by δ_k, the effectiveness is given by the tangent to the controlled variable–deflection curve at that operating condition and the actual value produced by δ_k is $g(\delta_k)$. The tangent to the controlled variable–deflection curve at δ_k intersects the $g(\delta)$ axis at a nonzero value given by $\epsilon(\delta_k)$. In a purely linear control allocation problem, this slope is translated along the vertical axis until it intersects the origin, so that $\epsilon(\delta_k) = 0$. Assume that at the next control update, the new command is $g_{des}(\delta)$. If the control allocation algorithm had perfect knowledge of the controlled variable–deflection curve, then the algorithm could accurately compute $\delta_{perfect}$ as the required control position. However,

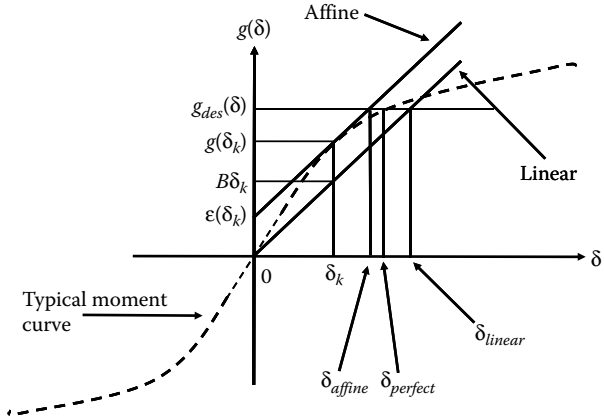

FIGURE 8.7 Linear versus affine control effectiveness.

in a linear (or affine) allocation problem, only the controlled variable–deflection slope (and intercept) is available to the allocation algorithm. If the linear allocation result is utilized, then a large error exists between $\delta_{perfect}$ and the position computed using linear information, δ_{linear}. On the other hand, if an affine representation of the controlled variable–deflection curve is utilized, the error between $\delta_{perfect}$ and δ_{affine} is much less than the error between $\delta_{perfect}$ and δ_{linear}. Hence, this method can produce significantly more accurate results for the computation of δ, as compared to a linear control allocation problem, when nonlinearities exist in the controlled variable–deflection data.

It should be noted that the affine method is well suited to algorithms that enforce rate limits of the effectors and are implemented in digital computers. The rate limits and digital implementation essentially limit the distance a control effector can travel in one timestep. Therefore, even if operating in a nonlinear region of the moment–deflection curve, the rate limit restrictions allow the control effectiveness (tangent of moment–deflection curve) to be an accurate representation of the nonlinear curve, in most cases.

The computation of $\boldsymbol{\epsilon}(\boldsymbol{\delta}_k)$ is straightforward and can be expressed in terms of quantities in Figure 8.7 as

$$\boldsymbol{\epsilon}(\boldsymbol{\delta}_k) = \mathbf{g}_{des}(\boldsymbol{\delta}) - \mathbf{B}\boldsymbol{\delta}_k \tag{8.72}$$

That is, $\boldsymbol{\epsilon}(\boldsymbol{\delta}_k)$ is the difference between the desired value, $\mathbf{g}_{des}(\boldsymbol{\delta})$, and the value that the control allocator thinks is being produced, $\mathbf{B}\boldsymbol{\delta}_k$. This is easily seen by considering a one-dimensional example, yet holds for the multidimensional case.

Care must be exercised when using linear or affine control allocation algorithms, particularly when there are slope reversals in the moment–deflection curve. In this case, it is possible for the allocation algorithm to get "stuck" at a local minimum. Fortunately, these slope reversals typically occur at the extreme deflection limits of control effectors. One method to overcome this problem is to postprocess the data so that the model does not include severe slope reversals. Note that only the model is modified, not the actual vehicle.

8.6.2 Nonlinear Programming for Separable Nonlinearities

The assumption that the moments are linear functions of the effectors is quite valid for most control effectors about their primary control axis, thus linear control allocation approaches are often successfully implemented. In this case, slight nonlinearities can be accommodated with affine control allocation or are seen as disturbances or model uncertainty and the control laws must be robust enough that system stability and performance are not compromised. However, there are situations where the effectors must operate away from the linear region, or when it may be beneficial to take advantage of the nonlinear effects. In these cases, it is desired to include the effects of the nonlinearities in the control allocation problem. For example, for left–right pairs of effectors, there is a quadratic relationship between the yawing moment and the effector displacement at low angles of attack. In this case, deflection of a single effector on one side of the vehicle induces a yawing moment to that side of the vehicle due to the asymmetric drag distribution. This yawing moment occurs for both positive and negative deflections of the control surface. In this case, a linear fit can only be applied to one side of the curve, otherwise a gross mismodeling of the controlled variable–effector relationship is introduced.

To further improve control allocation accuracy and to maximize performance of the effector suite on the aircraft requires taking advantage of the nonlinearities present in the data. Ideally, it is desired to solve nonlinear control allocation problems for the control effector vector, $\boldsymbol{\delta}$, such that

$$\mathbf{f}(\boldsymbol{\delta}) = \mathbf{d}_{des} \tag{8.73}$$

Nonlinear programming approaches are capable of solving these types of problems. Unfortunately, high order functions are typically needed to accurately fit the aerodynamic data, thus resulting in a rather difficult set of equations to solve. A more natural approach for aircraft applications is to utilize the inherent piecewise linear nature of the aerodynamic data. This is the approach that is pursued by Bolender [27].

The problem in this case is posed as a mixed integer linear program that can be solved using a branch-and-bound algorithm. It has been shown, in simulation, that performance is enhanced as compared to linear methods; however, the downside is that real time solutions can only be obtained for problems of limited size. The details are beyond the scope of this discussion; however, an interested reader should refer to [27] for more details.

8.7 Summary

This chapter provided a discussion of control allocation along with an overview of some of the techniques used to address linear and nonlinear control allocation problems. Methods to reduce the dimension of the control effector space such as explicit ganging and daisy chaining were presented. More sophisticated algorithms, which take into account control effector rate and position limits, were also discussed. LP has proved to be a viable method for solving online control allocation problems. The LP control allocator methodology was developed as a two branch optimization problem, followed by the mixed optimization framework. Different preference vectors were examined and are useful when multiple solutions exist. Control surface interactions were discussed and a method to take into account these effects, using linear programming, was developed. The interactions between constrained control allocators and actuator dynamics were considered for a simple first-order actuator model. Methods to compensate for these interactions were developed. Finally, two nonlinear control allocation techniques, namely, affine control allocation and piecewise linear control allocator were introduced. Each of the control allocation methods presented has unique advantages and disadvantages. Some are simple algorithms that are inexpensive computationally, while others are more complicated, provide more accurate results, but require significant computing resources. The best method depends on the application, an understanding of the physics of the problem, the computational resources available, and the rigor of the control law verification and validation process.

References

1. Brinker, J. S. and Wise, K. A., Nonlinear Simulation Analysis of a Tailless Advanced Fighter Aircraft Reconfigurable Flight Control Law, *Proceedings of the 1999 AIAA Guidance, Navigation, and Control Conference*, AIAA-1999-4040, August 1999.
2. Ward, D. G., Monaco, J. F., and Bodson, M., Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control, *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 6, 1998, pp. 948–956.
3. Chandler, P. R., Pachter, M., and Mears, M., System Identification for Adaptive and Reconfigurable Control, *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 3, May-June 1995, pp. 516–524.
4. Calise, A. J., Lee, S., and Sharma, M., Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft, *Proceedings of the 1998 Guidance, Navigation and Control Conference*, AIAA-1998-4108, August 1998.
5. Application of Multivariable Control Theory to Aircraft Control Laws, Tech. Rep. TR-96-3099, Wright Laboratory, WPAFB, OH, 1996.
6. Durham, W., Constrained Control Allocation, *Journal of Guidance, Control and Dynamics*, Vol. 16, No. 4, 1993, pp. 717–725.
7. Enns, D. F., Control Allocation Approaches, *Proceedings of the 1998 Guidance, Navigation and Control Conference*, AIAA-1998-4109, August 1998.
8. Paradiso, J. A., A Highly Adaptable Method of Managing Jets and Aerosurfaces for Control of Aerospace Vehicles, *Proceedings of the 1989 Guidance, Navigation and Control Conference*, AIAA-1989-3429, August 1989.
9. Paradiso, J. A., Adaptable Method of Managing Jets and Aerosurfaces for Aerospace Vehicle Control, *Journal of Guidance, Control and Dynamics*, Vol. 14, No. 1, 1991, pp. 44–50.

10. Buffington, J. M., Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3, Tech. Rep. Report. AFRL-VA-WP-TP-1999-3057, U.S. Air Force Research Lab., Wright Patterson AFB, OH, June 1999.
11. Doman, D. B. and Oppenheimer, M. W., Improving Control Allocation Accuracy for Nonlinear Aircraft Dynamics, *Proceedings of the 2002 Guidance, Navigation and Control Conference*, AIAA-2002-4667, August 2002.
12. Härkegård, O., Dynamic Control Allocation Using Constrained Quadratic Programming, *Proceedings of the 2002 Guidance, Navigation and Control Conference*, AIAA-2002-4761, August 2002.
13. Bodson, M., Evaluation of Optimization Methods for Control Allocation, *Journal of Guidance, Control and Dynamics*, Vol. 25, No. 4, 2002, pp. 703–711.
14. Lewis, F. L. and Syrmos, V. L., *Optimal Control*, John Wiley & Sons, Inc., New York, NY, 1995.
15. Kincaid, D. R. and Cheney, E. W., *Numerical Analysis*, Brooks/Cole Publishing Company, Pacific Grove, CA, 1990.
16. Durham, W., Attainable Moments for the Constrained Control Allocation Problem, *Journal of Guidance, Control and Dynamics*, Vol. 17, No. 6, 1994, pp. 1371–1373.
17. Luenberger, D., *Introduction to Linear and Nonlinear Programming*, Addison Wesley Longman, Reading, MA, 1984.
18. Winston, W. L. and Venkataramanan, M., *Introduction to Mathematical Programming, Volume I*, Academic Internet Publishers, Pacific Grove, CA, 2003.
19. Page, A. B. and Steinberg, M. L., A Closed-loop Comparison of Control Allocation Methods, *Proceedings of the 2000 Guidance, Navigation and Control Conference*, AIAA-2000-4538, August 2000.
20. Hac, A., Doman, D., and Oppenheimer, M., Unified Control of Brake- and Steer-by-Wire Systems Using Optimal Control Allocation Methods, *Proceedings of the 2006 SAE World Congress*, SAE-2006-01-0924, April 2006.
21. Oppenheimer, M. W. and Doman, D. B., A Method for Including Control Effector Interactions in the Control Allocation Problem, *Proceedings of the 2007 Guidance, Navigation and Control Conference*, AIAA 2007-6418, August 2007.
22. Doman, D. B. and Sparks, A. G., Concepts for Constrained Control Allocation of Mixed Quadratic and Linear Effectors, *Proceedings of the 2002 American Control Conference*, May 2002.
23. Venkataraman, R., Oppenheimer, M., and Doman, D., A New Control Allocation Method That Accounts for Effector Dynamics, *Proceedings of the 2004 IEEE Aerospace Conference*, IEEE-AC-1221, March 2004.
24. Bolling, J. G., *Implementation of Constrained Control Allocation Techniques Using an Aerodynamic Model of an F-15 Aircraft*, Master's thesis, Virginia Polytechnic Institute and State University, 1997.
25. Oppenheimer, M. W. and Doman, D. B., Methods for Compensating for Control Allocator and Actuator Interactions, *Journal of Guidance, Control and Dynamics*, Vol. 27, No. 5, 2004, pp. 922–927.
26. Oppenheimer, M. and Doman, D., A Method for Compensation of Interactions Between Second-Order Actuators and Control Allocators, *Proceedings of the 2005 IEEE Aerospace Conference*, IEEE-AC-1164, March 2005.
27. Bolender, M. A. and Doman, D. B., Non-linear Control Allocation Using Piecewise Linear Functions, *Journal of Guidance, Control and Dynamics*, Vol. 27, No. 6, Nov./Dec. 2004, pp. 1017–1027.

9

Swarm Stability

Veysel Gazi
TOBB University of Economics and Technology

Kevin M. Passino
The Ohio State University

9.1 Agent Model

It is possible to represent autonomous dynamic agents (e.g., robots, satellites, unmanned ground or air vehicles) using various mathematical models. One common representation is based on Newton's second law of motion of *point mass* particles, which sometimes is also called the *double integrator* model and is given by

$$\dot{x}_i = v_i, \qquad \dot{v}_i = u_i \tag{9.1}$$

where $x_i \in \mathbb{R}^n$ is the position of agent i, $v_i \in \mathbb{R}^n$ its velocity, and $u_i \in \mathbb{R}^n$ its control (force) input. The index i is used to denote that the corresponding dynamics belong to agent i. We assume that there are N identical agents in the swarm. In the above model, without loss of generality, it has been assumed that the mass of all agents is $m_i = 1$. This is because it can be easily compensated for by appropriately scaling the control inputs of the agents.

The model in Equation 9.1 is one of the widely used models in the literature on multiagent dynamic systems. Although the dynamics of realistic agents such as robots, satellites, unmanned ground or air vehicles are more complex it is still a relevant and useful model. This is because it can capture the high-level behavior of the agents without needing to delve into low-level details. Therefore, it allows studying higher level algorithms for various typical swarm behaviors. A recent survey on various relevant problems, agent models, and approaches for studying swarm behavior can be found in [1].

Given the agent dynamics in Equation 9.1, in this chapter we will discuss developing control algorithms for obtaining swarm behaviors such as aggregation, social foraging, and formation control. The approach we will describe for solving these problems is an artificial potential functions-based approach. In the last decades, artificial potential functions have been widely used for robot navigation and control. This is mainly because of the ease of their implementation and fast computation time. The implementation here

has also some differences from the classical approaches since we use artificial potentials to define also the interagent interactions.

We assume that all agents move simultaneously and know the exact relative position of the other agents in the swarm. Let $x^\top = [x_1^\top, x_2^\top, \ldots, x_N^\top] \in \mathbb{R}^{Nn}$ denote the vector of concatenated positions and $v^\top = [v_1^\top, v_2^\top, \ldots, v_N^\top] \in \mathbb{R}^{Nn}$ denote the vector of concatenated velocities of all the agents in the swarm. The pair (x, v) represents the state of the whole swarm, while the pairs $(x_i, v_i), i = 1, \ldots, N$ represent the states of the individual agents.

Let the interagent interactions in the swarm be represented by a potential function $J : \mathbb{R}^{Nn} \to \mathbb{R}$. We will call this function an interagent interaction potential or sometimes aggregation potential. By interagent interactions we mean the attraction or repulsion relationships between the agents in the swarm. Similarly, let the interactions of the agents with the environment be represented by another potential $\sigma : \mathbb{R}^n \to \mathbb{R}$. We call the environment potential $\sigma(\cdot)$ *the resource profile*. The environment potential may represent targets or goals to be moved toward and obstacles or threats to be avoided. The interagent interaction potential $J(x)$ is a function chosen by the designer based on the desired behavior of the swarm. We will briefly discuss which properties the potential functions should satisfy for aggregation and present results for a particular potential function. Similarly, the environment potential $\sigma(x)$ might represent an existing "resource" in the environment or might be chosen by the designer based on the layout of the environment within which the swarm moves as well as the desired navigation goals from the swarm.

9.2 Aggregation

Aggregation is one of the most basic behaviors seen in swarms in nature (such as insect colonies) and is sometimes the initial phase in collective tasks performed by a swarm. In this section, we discuss how to achieve aggregation for the point mass model in Equation 9.1.

A common approach is to design the potential function $J(x)$ such that it has a minimum at the desired behavior and then design the control inputs of the agents so as to force motion along its negative gradient. However, since we have second order dynamics in Equation 9.1 one needs also to have a damping term in the control input in order to avoid oscillations and instabilities. Therefore, one can choose the control input u_i of agent i in Equation 9.1 in the form

$$u_i = -kv_i - \nabla_{x_i} J(x) \tag{9.2}$$

where the first term with $k > 0$ is a damping term, whereas the second term represents motion along a potential field (i.e., the negative gradient of the potential function). The controller in Equation 9.2 has some similarities with a proportional-derivative (PD) controller, where the first (damping) term is similar to a derivative term (with zero desired velocity) and the potential function term is similar to a proportional term since $\nabla_{x_i} J(x)$ represents in a sense the error between a zero gradient potential and the current potential.

9.2.1 Aggregation Potential

Intuitively, in order to achieve aggregation, the agents should be attracted to each other. However, if there is only attraction then the swarm will probably collapse to a single point.* Therefore, while very close to each other, the agents should repel each other in order to avoid collisions and to keep appropriate

* The problem of collapsing to a point is another relevant problem and is known as distributed agreement, consensus, or rendezvous. It is usually studied under the conditions of limited sensing, local information, and dynamic (i.e., time-varying) neighborhood.

separation. Under these conditions, one possible choice of an artificial potential function is in the form

$$J(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[J_a\Big(\|x_i - x_j\|\Big) - J_r\Big(\|x_i - x_j\|\Big) \right] \tag{9.3}$$

where $J_a : \mathbb{R}^+ \to \mathbb{R}$ represents the attraction component and $J_r : \mathbb{R}^+ \to \mathbb{R}$ represents the repulsion component of the potential function.

Given the above type of potential function, the control input of agent i can be calculated as

$$u_i = -kv_i - \sum_{j=1\, j\neq i}^{N} \left[\nabla_{x_i} J_a\Big(\|x_i - x_j\|\Big) - \nabla_{x_i} J_r\Big(\|x_i - x_j\|\Big) \right] \tag{9.4}$$

Let us represent the gradients $\nabla_y J_a(\|y\|)$ and $\nabla_y J_r(\|y\|)$ in Equation 9.4 as

$$\nabla_y J_a(\|y\|) = y g_a(\|y\|) \qquad \text{and} \qquad \nabla_y J_r(\|y\|) = y g_r(\|y\|)$$

which are obtained using the chain rule and with appropriate definitions of the functions $g_a : \mathbb{R}^+ \to \mathbb{R}^+$ and $g_r : \mathbb{R}^+ \to \mathbb{R}^+$. Note that the attraction term $-y g_a(\|y\|)$ and the repulsion term $y g_r(\|y\|)$ both act on the line connecting the two interacting agents, but in opposite directions. The vector y determines the alignment (i.e., it guarantees that the interaction vector is along the line on which y is located), the terms $g_a(\|y\|)$ and $g_r(\|y\|)$ affect only the magnitude, whereas their difference determines the direction along vector y. Let us define the function $g(\cdot)$ as

$$g(y) = -y\Big[g_a(\|y\|) - g_r(\|y\|) \Big] \tag{9.5}$$

The function $g(\cdot)$, which we call the attraction/repulsion function, represents the gradient of the potential function and is the function generating the potential field. We assume that at large distances attraction dominates, at short distances repulsion dominates, and there is a *unique distance* at which the attraction and the repulsion balance. In other words, we assume that the potential function $J(x)$ and therefore the corresponding attraction/repulsion function $g(\cdot)$ satisfy the following assumption.

Assumption 9.1:

The potential function $J(x)$ is such that the level sets $\Omega_c = \{x | J(x) \leq c\}$ are compact, the corresponding attraction/repulsion function $g(\cdot)$ in Equation 9.5 is odd, and $g_a(\cdot)$ and $g_r(\cdot)$ are such that there exists a unique distance δ at which we have $g_a(\delta) = g_r(\delta)$. Moreover, we have $g_a(\|y\|) > g_r(\|y\|)$ for $\|y\| > \delta$ and $g_r(\|y\|) > g_a(\|y\|)$ for $\|y\| < \delta$.

In this chapter we consider only attraction/repulsion functions $g(\cdot)$, which are *odd* functions and therefore are symmetric with respect to the origin. This is an important feature which leads to reciprocity in the interagent interactions. Therefore, we consider only swarms with reciprocal interactions. The compactness of the level sets narrows the set of possible potential functions $J(x)$. However, it results in easier cohesiveness analysis since a controller that guarantees nonincrease in $J(x)$ immediately restrains the motion of the agents in a compact set. It might be possible to relax it, but special care must be taken to ensure that divergence of the agents from each other does not result in a decrease in $J(x)$.

Intuitively, since $J_a(\|x_i - x_j\|)$ is an attraction potential (and the motion is along the negative gradient) its minimum should occur on or around $\|x_i - x_j\| = 0$. In contrast, since $-J_r(\|x_i - x_j\|)$ is a repulsion potential its minimum should occur when $\|x_i - x_j\| \to \infty$. Moreover, from Assumption 9.1 the minimum of the combined $J_a(\|x_i - x_j\|) - J_r(\|x_i - x_j\|)$ occurs at $\|x_i - x_j\| = \delta$ at which distance the attraction and

repulsion between two agents balance. Note, however, that when there are more than two agents involved, the minimum of the overall potential $J(x)$ does not necessarily occur at $\|x_i - x_j\| = \delta$ for all $j \neq i$ and also there might be multiple minima.

It is possible to view $J(x)$ as a representation of the artificial potential energy of the swarm, whose value depends on the interagent distances. Then the control input in Equation 9.2 (and therefore the one in Equation 9.4), which force motion along the negative gradient of $J(x)$, are controllers which try to minimize the artificial potential energy of the swarm or at least the portion of the artificial potential energy, which is due to the relative position of the corresponding agent to the other agents in the swarm.

One potential function which satisfies the above conditions, including Assumption 9.1, and that has been used in the literature for swarm aggregations (see e.g., [2]) is

$$J(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[\frac{a}{2} \|x_i - x_j\|^2 + \frac{bc}{2} \exp\left(-\frac{\|x_i - x_j\|^2}{c} \right) \right] \tag{9.6}$$

where a, b, and c are parameters to be chosen by the designer. The parameter a is an attraction parameter, whereas the parameters b and c are repulsion parameters. In order for the repulsion to dominate on short distances it is required that $b > a$ is satisfied. The corresponding attraction/repulsion function of the function in Equation 9.6 can be calculated as

$$g(y) = -y \left[a - b \exp\left(-\frac{\|y\|^2}{c} \right) \right] \tag{9.7}$$

Note that this function falls within potential functions with linear attraction and bounded repulsion and is shown in Figure 9.1. (See [3] for some other types of potential functions.) Figure 9.1a shows the attraction potential, the repulsion potential and the combined total potential in Equation 9.6 acting between two agents for the parameters $a = 1$, $b = 10$, and $c = 1$, whereas Figure 9.1b shows the corresponding attraction/repulsion function in Equation 9.7. We will use the functions in Equations 9.6 and 9.7 in the subsequent analysis and state most of the results based on them. However, note that the results immediately hold for all potential functions with linear attraction and bounded repulsion and satisfying Assumption 9.1 and can be extended for other potential functions as well.

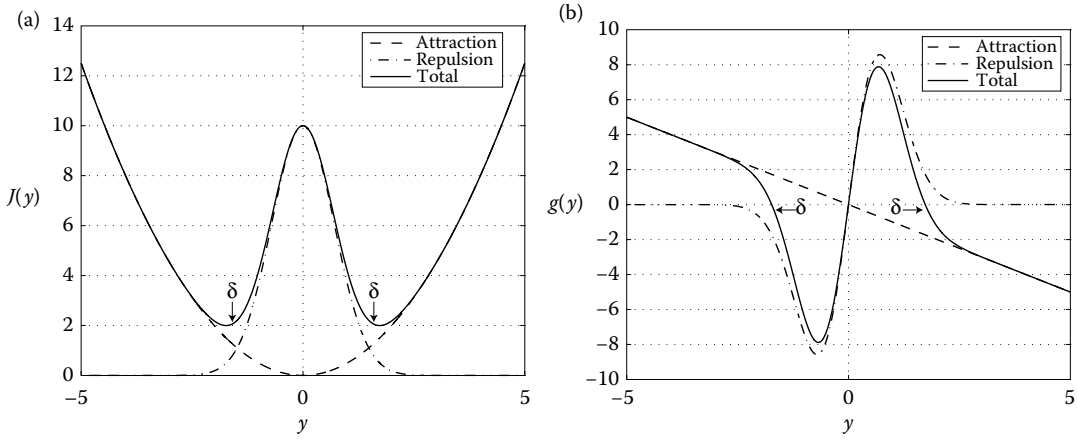

FIGURE 9.1 Plot of the potential functions in Equation 9.6 (between two agents only) and the corresponding attraction/repulsion function in Equation 9.7 for $a = 1$, $b = 10$, and $c = 1$. (a) Potential function. (b) Attraction/repulsion function.

9.2.2 Analysis of Swarm Motion

In this section, we will represent the collective motion of the swarm by the motion of its *centroid* which is defined as

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \bar{v} = \frac{1}{N}\sum_{i=1}^{N} v_i$$

Note once more that since the functions $g(\cdot)$ are odd, and therefore symmetric with respect to the origin, the interagent interactions are reciprocal. Therefore, the effect of interagent interactions on the dynamics of the centroid sum up to zero. If the agent dynamics were first order (i.e., single integrator) the centroid would be stationary. However, in the case here the centroid is not in general stationary since we have a second order system. In fact it is possible to show that the velocity of the centroid converges to zero exponentially fast meaning that the centroid converges to a constant position. This is stated formally in the following lemma.

Lemma 9.1:

The centroid $\bar{x}$ of the swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.4 with potential function $J(x)$ which satisfies Assumption 9.1 converges to a stationary point x_c exponentially fast.

Proof. The time derivative of velocity of the centroid is given by

$$\dot{\bar{v}} = -k\frac{1}{N}\sum_{i=1}^{N} v_i - \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1\, j\neq i}^{N}\left[g_a\Big(\|x_i - x_j\|\Big) - g_r\Big(\|x_i - x_j\|\Big)\right](x_i - x_j) = -k\bar{v}$$

where the second term cancels out from the reciprocity in the interactions. From this equation one can see that since $k > 0$ as $t \to \infty$ the centroid velocity $\bar{v} \to 0$ is exponentially fast implying that $\bar{x} \to x_c$ is exponentially fast for some $x_c \in \mathbb{R}^n$. ■

Note that the speed of convergence of the centroid to a constant location depends on the damping parameter k. Also, since the dynamics are deterministic the final centroid position x_c depends only on the initial positions and velocities of the agents. In fact, it is possible to show that $x_c = \bar{x}(0) + \frac{1}{k}\bar{v}(0)$. Therefore, if the initial velocities of the agents are zero the centroid will be stationary. The fact that the centroid stops does not, in general, imply that the agents stop as well. However, it is possible to show that this is the case. With this objective let us denote the invariant set of equilibrium (or stationary) points for the swarm with

$$\Omega_e = \{(x, v) : v_i = 0 \text{ and } \nabla_{x_i} J(x) = 0 \ \ \forall i\} \tag{9.8}$$

Note that $(x, v) \in \Omega_e$ implies that $v_i = 0$ for all $i = 1, \ldots, N$, implying that all agents are stationary. Below we show that the state of the swarm converges to Ω_e and the agents stop motion.

Theorem 9.1:

Consider a swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.4 with a potential function $J(x)$ which satisfies Assumption 9.1. For any $(x(0), v(0)) \in \mathbb{R}^{2Nn}$, as $t \to \infty$ we have $(x(t), v(t)) \to \Omega_e$.

Proof. Let us define the Lyapunov function $V = V(x, v)$ of the system as

$$V = J(x) + \frac{1}{2}\sum_{i=1}^{N} \|v_i\|^2$$

From the compactness of the level sets of $J(x)$ and the definition of $V(x, v)$ we know that the level sets of V are also compact. Taking its derivative with respect to time one can show that

$$\dot{V} = -k\sum_{i=1}^{N} \|v_i\|^2 \leq 0$$

for all t implying a decrease in V unless $v_i = 0$ for all $i = 1, \ldots, N$. Therefore, the motion of the swarm is confined to the compact set $V(x(0), v(0))$. Define the set $\Omega_v = \{(x, v) : v_i = 0 \ \forall i\}$ and note that $\Omega_e \subset \Omega_v$. It is obvious that $(x, v) \to \Omega_v$ as $t \to \infty$. Further, we see that the agent dynamics on Ω_v are given by

$$\dot{x}_i = v_i, \quad \dot{v}_i = -\nabla_{x_i} J(x)$$

implying that Ω_v is not invariant. In fact, one can see that the largest invariant subset of Ω_v is the set Ω_e. Therefore, from the LaSalle's invariance principle we can conclude that as $t \to \infty$ we have $(x, v) \to \Omega_e$. ■

Besides stating that the agents will eventually stop, the above result also states that $\nabla_{x_i} J(x) = 0$ will be achieved for all agents implying that minimization (at least local) of the potential function $J(x)$ will be achieved. This result will be useful later in showing the cohesiveness of the swarm. As mentioned above, aggregation is a very basic behavior seen in many swarms in nature. Note, however, that the aggregation discussed here is different from the flocking behavior in which the agents do not stop but achieve heading alignment and perform cohesive motion in a common direction. It is possible to achieve also flocking behavior for the swarms composed of agents with dynamics in Equation 9.1. For that objective the velocity damping term in the control input in Equation 9.2 needs to be replaced with a velocity matching term in the form

$$u_i = -k\sum_{j=1\, j\neq i}^{N} (v_i - v_j) - \nabla_{x_i} J(x) \tag{9.9}$$

In that case the agent velocities will converge to the same value and the agents will move in the same direction. Note that the direction of motion and the common velocity is not pre-defined and is an emergent property.* $\nabla_{x_i} J(x) = 0$ will still be achieved for all agents. Note that behavior similar to flocking might be more suitable for engineering swarm applications such as *uninhabited air vehicles* (UAVs) in which the agents never stop (excluding UAVs such as helicopters, quadrotors, or balloons). Similarly, if the swarm is required to track a given reference trajectory $\{\ddot{x}_r, \dot{x}_r, x_r\}$, which is known (or estimated) by the agents, the control inputs of the agents can be set as

$$u_i = \ddot{x}_r - k_v(v_i - \dot{x}_r) - k_p(x_i - x_r) - \nabla_{x_i} J(x), \tag{9.10}$$

which will result in the fact that the centroid of the swarm tracks the reference trajectory and the swarm moves as a cohesive entity following the reference trajectory.

* Since everything here is deterministic the direction of motion depends on the initial conditions, that is, the initial agent positions and velocities. If the agent interaction definitions were dynamic, it would depend also on the change in the interaction topology.

9.2.3 Swarm Cohesion Analysis

There might be different definitions of swarm stability and size. Here we consider cohesion as a stability property and define the swarm size in terms of agent distances to the swarm centroid $\bar{x}$, which for agent i is defined in vector form as $e_i = x_i - \bar{x}$ and the bound on the magnitude of e_i is taken as a measure of the swarm size.

From Theorem 9.1 we know that as $t \to \infty$ we have

$$\nabla_{x_i} J(x) = \sum_{j=1\, j\neq i}^{N} \Big[g_a\big(\|x_i - x_j\|\big) - g_r\big(\|x_i - x_j\|\big) \Big](x_i - x_j) = 0 \tag{9.11}$$

for all agents i. We will perform the analysis for the potential function in Equation 9.6. However, note that it is applicable for all functions with linear attraction and bounded repulsion and also can be extended to other types of potential functions. For the potential function in Equation 9.6, Equation 9.11 becomes

$$\nabla_{x_i} J(x) = \sum_{j=1\, j\neq i}^{N} \left[a - b \exp\left(-\frac{\|x_i - x_j\|^2}{c} \right) \right] (x_i - x_j) = 0 \tag{9.12}$$

Using the fact that $\sum_{j=1\, j\neq i}^{N} (x_i - x_j) = Ne_i$ and rearranging Equation 9.12 can be written as

$$e_i = \frac{1}{aN} \sum_{j=1, j\neq i}^{N} b \exp\left(-\frac{\|x_i - x_j\|^2}{c} \right) (x_i - x_j)$$

from which one can obtain

$$\|e_i\| \leq \frac{b}{a} \sqrt{\frac{c}{2}} \exp\left(-\frac{1}{2} \right)$$

Note that this is a bound obtained for the potential function $J(x)$ in Equation 9.6. Given another potential function with linear attraction and bounded repulsion, the expression of the bound might be slightly different but will still depend on the attraction and repulsion parameters of the function. The above result is stated formally in the following theorem.

Theorem 9.2:

Consider a swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.4 with a potential function $J(x)$ given in Equation 9.6 (which satisfies Assumption 9.1, and has linear attraction and bounded repulsion). As time progresses all agents in the swarm will converge to a hyperball

$$B_\varepsilon(\bar{x}) = \left\{ x : \|x - \bar{x}\| \leq \varepsilon = \frac{b}{a} \sqrt{\frac{c}{2}} \exp\left(-\frac{1}{2} \right) \right\}$$

We would like to emphasize once again that though stated for the potential function $J(x)$ in Equation 9.6, the result in Theorem 9.2 holds directly with modification of only the value of the bound ε for other potential functions with linear attraction and bounded repulsion and satisfying Assumption 9.1. Also note that if the repulsion were removed, that is, if $g_r\Big(\|x_i - x_j\|\Big) \equiv 0$ for all i and j, $j \neq i$, then the swarm would shrink to its centroid $\bar{x}$. In contrast, if the attraction were removed, that is, if $g_a\Big(\|x_i - x_j\|\Big) \equiv 0$ for all i and j, $j \neq i$, then the swarm would disperse in all directions away from the centroid $\bar{x}$ toward infinity. Having the attraction dominate at large distances prevents the swarm from dispersing, whereas

having the repulsion dominate on short distances prevents it from collapsing to a single point, and the equilibrium is established in between.

In *some* biological swarms it has been observed that individuals are attracted more to larger crowded swarms. This property is also present in the above model in which the attraction is proportional to the number of agents in the swarm. Note also that the dependence of the resultant swarm size on the attraction and repulsion parameters a, b, and c makes intuitive sense and larger attraction (larger a) leads to a smaller swarm size, whereas larger repulsion (larger b and/or larger c) leads to a larger swarm size.

9.3 Formation Control

The formation control problem can be described as the problem of choosing the control inputs for the agents such that they asymptotically form and maintain a predefined geometric shape. Given the agent model in Equation 9.1 and the control input in the form of Equation 9.4, the task is to design a potential function $J(x)$ so that the formation control problem is solved. Given the results in the preceding section one can immediately see that if the potential function is chosen intelligently, such that it has a minimum and $\nabla_{x_i} J(x) = 0$ at the desired formation, then the formation control problem will be solved at least locally using the procedure discussed for the case of aggregation. The problem is solved only locally since, in general, the potential functions suffer from the so-called local minima problem and the potential functions considered here (which are of type based only on relative distances) suffer from that problem as well.* Nevertheless, assuming that one would like to use the potential functions-based method discussed for aggregation in the preceding section, the procedure boils down to the proper choice of the potential function $J(x)$. In order to achieve a geometric formation, in which the interagent distances are pair dependent, the interaction parameters and therefore the attraction/repulsion functions $g(\cdot)$ also need to be pair dependent. For this reason, let us assume that the potential function $J(x)$ is denoted as

$$J(x) = J_{formation}(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[J_{ija}\Big(\|x_i - x_j\|\Big) - J_{ijr}\Big(\|x_i - x_j\|\Big) \right] \tag{9.13}$$

Then, the corresponding control input in Equation 9.4 is given by

$$\begin{aligned} u_i &= -kv_i - \sum_{j=1, j\neq i}^{N} \left[\nabla_{x_i} J_{ija}\Big(\|x_i - x_j\|\Big) - \nabla_{x_i} J_{ijr}\Big(\|x_i - x_j\|\Big) \right] \\ &= -kv_i - \sum_{j=1, j\neq i}^{N} \left[g_{ija}\Big(\|x_i - x_j\|\Big) - g_{ijr}\Big(\|x_i - x_j\|\Big) \right](x_i - x_j) \end{aligned} \tag{9.14}$$

Note that it is still assumed that the potential function $J(x)$ satisfies Assumption 9.1 and that for all pairs (i, j) the pair-dependent attraction/repulsion functions

$$g_{ij}(y) = -y\Big[g_{ija}(y) - g_{ijr}(y)\Big] \tag{9.15}$$

are such that the attraction and repulsion balance at pair-dependent equilibrium distances δ_{ij} which can be different for different pairs of individuals. In order to fit the potential function in Equation 9.6 to the

* A local minimum is defined as a point x_0 at which $J(x_0) \leq J(x)$ for all $x \in N(x_0)$ where $N(x_0)$ is some neighborhood of x_0. Some of these points may not correspond to the desired formation. At these points, the gradient is zero as well and the agents can get stuck at such points.

formation control framework, its parameters a, b, and c need to be set as pair dependent of the form

$$J(x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left[\frac{a_{ij}}{2} \|x_i - x_j\|^2 + \frac{b_{ij} c_{ij}}{2} \exp\left(-\frac{\|x_i - x_j\|^2}{c_{ij}} \right) \right] \tag{9.16}$$

Then, the control input for agent i will be of the form

$$u_i = -kv_i - \sum_{j=1, j\neq i}^{N} \left[a_{ij} - b_{ij} \exp\left(-\frac{\|x_i - x_j\|^2}{c_{ij}} \right) \right] (x_i - x_j) \tag{9.17}$$

Assume that the desired formation is specified *a priori* by *formation constraints* of the form

$$\|x_i - x_j\| = d_{ij}$$

for all $(i, j), j \neq i$. It is required that the potential function and the control inputs for the agents are chosen such that the formation is achieved (i.e., stabilized). The above type of formation constraints, in which only relative distances are used, are sufficient to specify a formation only in terms of relative interagent arrangement and rotation or translation of the whole swarm is allowed. This has the advantage that global absolute position information is not required. However, as mentioned above, it also results in the existence of local minima and in general allows for only local results. We would like to also mention that in order for the problem to be solvable, the formation constraints should not be conflicting and the formation should be feasible.

Having set the framework, the last step is to choose the potential function (and therefore the corresponding attraction/repulsion functions $g_{ij}(\cdot)$) such that the pair dependent inter-agent distances δ_{ij} are equal to the desired distances d_{ij} ($\delta_{ij} = d_{ij}$). To achieve this for the potential function in Equation 9.16 one needs to choose the parameters a_{ij}, b_{ij}, and c_{ij} appropriately. One possible choice is to set $b_{ij} = b$, $c_{ij} = c$ for all pairs (i, j) and for some constants b and c and to calculate a_{ij} as $a_{ij} = b \exp\left(-d_{ij}^2/c\right)$. Then, from the results of the case of aggregation one can deduce the following.

Corollary 9.1:

Consider a swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.14 with potential function $J(x)$ which satisfies Assumption 9.1 and has pair dependent interactions. Assume that the pair-dependent interagent attraction/repulsion functions $g_{ij}(\cdot)$ are chosen such that the distances δ_{ij} at which the interagent attractions and repulsions between pairs (i, j) balance satisfy $\delta_{ij} = d_{ij}$, where d_{ij} are the desired formation distances. Then, the equilibrium at the desired formation is locally asymptotically stable.

The reason for obtaining only local asymptotic stability in the above result is the local minima problem. Since there might be local minima in the potential function $J(x)$ which do not correspond to the desired formation, in general it is not possible to guarantee that the swarm will converge to the desired formation from any initial position and convergence to the desired formation is only guaranteed if the initial positions of the agents are "sufficiently close" to that configuration. Also the size of the region of attraction of the desired formation may be different for different formation shapes and potential function parameters. There are strategies such as simulated annealing which can be used to escape local minima. However, discussing such strategies is outside the scope of this chapter.

We would like to also mention that as in the case of aggregating swarms the procedure can be extended to the cases of tracking a desired trajectory or flocking-type velocity matching in a predefined formation.

9.4 Social Foraging

The case of social foraging is different from the cases of aggregation and formation control in the sense that in addition to the interagent interactions the dynamics of the swarm is affected by the environment. Within the potential functions framework considered in this chapter the effect of the environment can be incorporated by representing it also by a potential function in the form $\sigma : \mathbb{R}^n \to \mathbb{R}$. As mentioned previously we call this potential function the "resource profile." It represents the resources in the environment, which can be nutrients or some attractant or repellent substances (e.g., food/nutrients, pheromones laid by other individuals, or toxic chemicals in biological swarms or goals, targets, obstacles, or threats in engineering swarm applications). Let us assume without loss of generality that the areas with lower values of the resource profile represent regions with higher nutrient/target/goal concentration (and lower toxic/threats/obstacle concentration) and are therefore more favorable to the agents. In the context of multiagent (i.e., multirobot) systems, given an environment with obstacles or threats to be avoided (analogous to toxic substances) and targets or goals to be moved toward (analogous to food), the resource profile $\sigma(x)$ can be chosen or "designed" by the user to incorporate these properties/effects.

Under this setup incorporating the effect of the environment into the framework in the preceding sections is straightforward and one can choose the overall potential function $\bar{J}(x)$ as

$$\bar{J}(x) = \sum_{i=1}^{N} \sigma(x_i) + J(x)$$

where $J(x)$ is the potential function used for aggregation in the preceding sections. In other words, besides the effect of the environment the overall potential function contains a part for aggregation which is assumed to satisfy Assumption 9.1. Therefore, the corresponding controller will also contain terms based on both the resource profile and the interagent interaction potential and will be in the form

$$\begin{aligned} u_i &= -kv_i - \nabla_{x_i}\sigma(x_i) - \nabla_{x_i}J(x) \\ &= -kv_i - \nabla_{x_i}\sigma(x_i) - \sum_{j=1, j\neq i}^{N} \left[g_a\Big(\|x_i - x_j\|\Big) - g_r\Big(\|x_i - x_j\|\Big) \right](x_i - x_j) \end{aligned} \tag{9.18}$$

Here, the term $-\nabla_{x_i}\sigma(x_i)$ guides agent i toward lower values (i.e., targets, goals) and away from the higher values (i.e., threats, obstacles) of the resource profile $\sigma(x_i)$. It can be viewed as a "selfish" part of the agent controller, whereas the second term $-\nabla_{x_i}J(x)$ is a "social" part, which tends to keep the agents together in the group. Note also that the controller in Equation 9.18 implicitly assumes that the gradient of the resource profile $-\nabla_{x_i}\sigma(x_i)$ is known to (i.e., either measured or estimated by) the agents. This is not too much restrictive since it is known that even simple bacteria such as *E. coli* can estimate and climb gradients.*

Let us choose the generalized Lyapunov function candidate† as

$$V = \bar{J}(x) + \frac{1}{2}\sum_{i=1}^{N} \|v_i\|^2 = \sum_{i=1}^{N} \sigma(x_i) + J(x) + \frac{1}{2}\sum_{i=1}^{N} \|v_i\|^2, \tag{9.19}$$

which is basically the Lyapunov function for the case of aggregation with the effect of the environment incorporated as well. Taking its time derivative, after a few straightforward manipulations one can once

* See the references cited in [4,5] for studies on *E. coli* foraging. There are even optimization algorithms inspired from foraging behavior of *E. coli*.

† Here V is called a generalized Lyapunov function candidate since it is not necessarily positive definite.

again obtain

$$\dot{V} = -k \sum_{i=1}^{N} \|v_i\|^2 \leq 0 \tag{9.20}$$

for all t. Therefore, provided that the motion of the swarm is confined to a compact set (which is automatically guaranteed if the level sets of the Lyapunov function V in Equation 9.19 are compact) conclusions similar to the case of aggregation can be drawn. However, note that the compactness of the level sets of V depends also on the properties of the resource profile $\sigma(x)$. Therefore, the stability properties of the swarm might be different for different profiles. We will analyze the motion of the system in several profiles in the subsequent sections. In particular we will consider plane, quadratic, and Gaussian profiles. For all these cases it is possible to only show that in the case of valley-type quadratic or Gaussian profiles the agents will stop or basically as $t \to \infty$ the state $(x(t), v(t))$ converges to Ω_e, where Ω_e is the set defined in Equation 9.8. Otherwise the agents do not necessarily stop. We state this formally below, although the expressions of the profiles are provided in the following sections.

Theorem 9.3:

Consider a foraging swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with interagent interaction potential function $J(x)$ which satisfies Assumption 9.1. Assume that the resource profile $\sigma(\cdot)$ of the environment is one of the following

- *A valley-type quadratic profile (i.e., the profile in Equation 9.24 below with $A_\sigma > 0$)*
- *A valley-type Gaussian profile (i.e., the profile in Equation 9.25 below with $A_\sigma > 0$)*

Then, as $t \to \infty$ we have $(x(t), v(t)) \to \Omega_e$.

Proof. The proof is very similar to the proof of Theorem 9.1. ■

Note that Theorem 9.3 implies that for the mentioned cases (i.e., quadratic and Gaussian profiles with $A_\sigma > 0$) as $t \to \infty$ for all i we have

$$-\nabla_{x_i}\sigma(x) - \nabla_{x_i}J(x) = 0 \tag{9.21}$$

This equality will be useful in the subsequent analysis. Another issue to note here is that for the cases excluded in Theorem 9.3, that is, for the plane profile, hill-type quadratic profile (i.e., quadratic profile with $A_\sigma < 0$), and hill-type Gaussian profile (i.e., Gaussian profile with $A_\sigma < 0$) the motion of the agents may not be confined to a compact set. Therefore, we cannot apply the LaSalle's Invariance Principle (which is the main tool in the proof). For these cases, while the swarm might stay cohesive it might diverge and never stop as we will see below. However, before that, let us first turn our attention to the motion of the *centroid* of the swarm. With a straightforward derivation one can show that

$$\dot{\bar{x}} = \bar{v}, \qquad \dot{\bar{v}} = -k\bar{v} - \frac{1}{N}\sum_{i=1}^{N} \nabla_{x_i}\sigma(x_i) \tag{9.22}$$

from which we can see that the motion of the swarm centroid is "guided" by the average of the gradient of the resource profile evaluated at the agent locations. Therefore, as is the case for the stability properties, the dynamics of the centroid, which in a sense represents the overall collective motion of the swarm, can be different for different profiles.

9.4.1 Plane Resource Profile

The first resource profile we will consider is the *plane* resource profile described by an equation of the form

$$\sigma(y) = a_\sigma^\top y + b_\sigma \tag{9.23}$$

where $a_\sigma \in \mathbb{R}^n$ and $b_\sigma \in \mathbb{R}$. Calculating its gradient at a point $y \in \mathbb{R}^n$ one obtains

$$\nabla_y \sigma(y) = a_\sigma$$

Then, from Equation 9.22 the motion of the centroid of the swarm can be described by the equations

$$\dot{\bar{x}} = \bar{v}, \qquad \dot{\bar{v}} = -k\bar{v} - a_\sigma$$

from which one can see that as $t \to \infty$ we have $\dot{\bar{v}}(t) \to 0$ and $\bar{v}(t) \to -\frac{1}{k} a_\sigma$ implying that the centroid of the swarm will be moving with this constant velocity vector (which is along the negative gradient of the plane resource profile) and eventually will diverge toward infinity (where the minimum of the profile occurs). This can be stated formally as follows.

Lemma 9.2:

Consider a foraging swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with interagent interaction potential function $J(x)$, which satisfies Assumption 9.1. Assume that the resource profile $\sigma(\cdot)$ of the environment is given by Equation 9.23. Then, as $t \to \infty$ the centroid of the swarm moves along the negative gradient of the profile toward infinity with velocity $\bar{v}(t) = -\frac{1}{k} a_\sigma$.

In order to analyze the relative dynamics of the agents in the swarm let us define the state transformation $z_i = x_i - \bar{x}$ and $\zeta_i = v_i - \bar{v}$. Also, let us assume that the aggregation potential is the one given in Equation 9.6. Then, the agent dynamics under these relative state variables can be expressed as

$$\dot{z}_i = \zeta_i, \qquad \dot{\zeta}_i = -k\zeta_i - \sum_{j=1, j\neq i}^{N} \left[a - b \exp\left(-\frac{\|z_i - z_j\|^2}{c} \right) \right] (z_i - z_j),$$

which is exactly the equation for aggregation obtained from Equations 9.1, 9.2, and 9.6 and the results obtained for the case of aggregation hold for these relative dynamics as well and can be stated as follows.

Corollary 9.2:

Consider a swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with inter-agent interaction potential $J(x)$ given in Equation 9.6 (which satisfies Assumption 9.1 with linear attraction and bounded repulsion). Assume that the resource profile $\sigma(\cdot)$ of the environment is given by Equation 9.23. As time progresses all agents in the swarm will converge to a hyperball

$$B_\varepsilon(\bar{x}) = \left\{ x : \|x - \bar{x}\| \leq \varepsilon = \frac{b}{a} \sqrt{\frac{c}{2}} \exp\left(-\frac{1}{2} \right) \right\}$$

This result implies that the cohesiveness property of the swarm will be preserved. Note that for the plane profile we have $\Omega_e = \emptyset$. In other words, there is no equilibrium for the swarm moving in a plane profile. Moreover, simultaneous application of Lemma 9.2 and Corollary 9.2 results in the conclusion that the swarm will move as a cohesive entity along the negative gradient of the profile toward infinity.

9.4.2 Quadratic Resource Profile

The second resource profile we will consider is the *quadratic* resource profile given by

$$\sigma(y) = \frac{A_\sigma}{2}\|y - c_\sigma\|^2 + b_\sigma \tag{9.24}$$

where $A_\sigma \in \mathbb{R}$, $b_\sigma \in \mathbb{R}$, and $c_\sigma \in \mathbb{R}^n$. Note that this profile has a global extremum (either a minimum or a maximum depending on the sign of A_σ) at $y = c_\sigma$. Taking its gradient at a point $y \in \mathbb{R}^n$ one obtains

$$\nabla_y \sigma(y) = A_\sigma(y - c_\sigma)$$

The motion of the centroid $\bar{x}$ can be calculated as

$$\dot{\bar{x}} = \bar{v}, \qquad \dot{\bar{v}} = -k\bar{v} - A_\sigma(\bar{x} - c_\sigma)$$

which with the change of coordinates $\bar{x}_c = \bar{x} - c_\sigma$ and $\bar{v}_c = \bar{v}$ and rewriting in matrix form can be expressed as

$$\begin{bmatrix} \dot{\bar{x}}_c \\ \dot{\bar{v}}_c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -A_\sigma & -k \end{bmatrix} \begin{bmatrix} \bar{x}_c \\ \bar{v}_c \end{bmatrix}$$

Note that $\bar{x} = c_\sigma$ and $\bar{v} = 0$ is an equilibrium point of this system. Calculating the eigenvalues of the system matrix one can obtain $\lambda_{1,2} = (-k \pm \sqrt{k^2 - 4A_\sigma})/2$. The motion of the centroid will depend on the sign of the real parts of the eigenvalues $\lambda_{1,2}$ and can be stated as in the following result.

Lemma 9.3:

Consider a foraging swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with interagent interaction potential $J(x)$ which satisfies Assumption 9.1. Assume that the resource profile $\sigma(\cdot)$ of the environment is given by Equation 9.24. As $t \to \infty$ we have

- *If $A_\sigma > 0$, then $\bar{x}(t) \to c_\sigma$ (i.e., the centroid of the swarm converges to the global minimum c_σ of the profile), or*
- *If $A_\sigma < 0$ and $\bar{x}(0) \neq c_\sigma$, then $\bar{x}(t) \to \infty$ (i.e., the centroid of the swarm diverges from the global maximum c_σ of the profile).*

Proof. By inspecting the eigenvalues of the system matrix one can see that for $A_\sigma > 0$ both of the eigenvalues have negative real parts, whereas for $A_\sigma < 0$ one of the eigenvalues has a positive real part from which the conclusions immediately follow. ■

This is an intuitive result, which basically states that the centroid of the swarm will converge to the global minimum in the case of a valley-type profile and will diverge from the global maximum in the case of a hill-type profile. Also, assuming that $A_\sigma > 0$, the sign of $k^2 - 4A_\sigma$ will determine the characteristics of the centroid motion. In particular, if $0 < A_\sigma < k^2/4$ its motion will be overdamped, whereas for $A_\sigma > k^2/4$ its motion will be underdamped. Recall from Theorem 9.3 that for a quadratic profile with $A_\sigma > 0$ it was shown that the agents will eventually stop motion. Now, let us assume that the interagent interaction potential is the one in Equation 9.6 and analyze the cohesiveness of the swarm.

Theorem 9.4:

Consider a foraging swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with interagent interaction potential $J(x)$ in Equation 9.6 (which satisfies Assumption 9.1

and has linear attraction and bounded repulsion). Assume that the resource profile $\sigma(\cdot)$ *of the environment is given by Equation 9.24 with* $A_\sigma > 0$*. Then, as* $t \to \infty$ *all agents* $i = 1, \ldots, N$*, will enter*

$$B_\varepsilon(c_\sigma) = \left\{ x : \|x - c_\sigma\| \leq \varepsilon = \frac{b(N-1)}{aN + A_\sigma} \sqrt{\frac{c}{2}} \exp\left(-\frac{1}{2}\right) \right\}$$

Proof. Since $A_\sigma > 0$ from Theorem 9.3 we know that as $t \to \infty$ we have $(x(t), v(t)) \to \Omega_e$. Therefore, as $t \to \infty$ Equation 9.21 is satisfied. Manipulating this equation after substituting the expressions of $\sigma(x)$ in Equation 9.24 and $J(x)$ in Equation 9.6 and using the result from Lemma 9.3 that $\bar{x}(t) \to c_\sigma$ the bound is obtained. ■

This result is important because it gives convergence to nutrient-rich regions (targets, goals) of the resource profile of *all* agents in the swarm. It is also possible to show divergence from toxic regions (threats, obstacles) of the resource profile for $A_\sigma < 0$.

Note that from Equation 9.21 it is possible to show that as $t \to \infty$ (or basically at equilibrium) $\bar{x}(t) = c_\sigma$ using another approach as well. To see this let us sum up Equation 9.21 for all i

$$\sum_{i=1}^{N} \Big(-\nabla_{x_i}\sigma(x) - \nabla_{x_i} J(x) \Big) = -\sum_{i=1}^{N} \nabla_{x_i}\sigma(x) = -A_\sigma \sum_{i=1}^{N} (x_i - c_\sigma) = -A_\sigma(\bar{x} - c_\sigma) = 0$$

where $\sum_{i=1}^{N} \nabla_{x_i} J(x)$ cancels out due to reciprocity in the agent interactions and the result is obtained from the last equality.

9.4.3 Gaussian Resource Profile

The third type of resource profile is the *Gaussian* profile described by an equation of the form

$$\sigma(y) = -\frac{A_\sigma}{2} \exp\left(-\frac{\|y - c_\sigma\|^2}{l_\sigma}\right) + b_\sigma \tag{9.25}$$

where $A_\sigma \in \mathbb{R}$, $b_\sigma \in \mathbb{R}$, $l_\sigma \in \mathbb{R}^+$, and $c_\sigma \in \mathbb{R}^n$. Note that this profile also has the unique extremum (either a global minimum or a global maximum depending on the sign of A_σ) at $y = c_\sigma$. Its gradient at a point $y \in \mathbb{R}^n$ can be calculated as

$$\nabla_y \sigma(y) = \frac{A_\sigma}{l_\sigma}(y - c_\sigma) \exp\left(-\frac{\|y - c_\sigma\|^2}{l_\sigma}\right)$$

which is bounded and satisfies

$$\|\nabla_y \sigma(y)\| \leq \bar{\sigma} = \frac{|A_\sigma|}{\sqrt{2l_\sigma}} \exp\left(-\frac{1}{2}\right) \tag{9.26}$$

for all $y \in \mathbb{R}^n$. This bound will be useful in the analysis below.

Using the expression of the gradient the dynamics of the centroid can be obtained as

$$\dot{\bar{x}} = \bar{v}, \quad \dot{\bar{v}}(t) = -k\bar{v} - \frac{A_\sigma}{Nl_\sigma} \sum_{i=1}^{N} (x_i - c_\sigma) \exp\left(-\frac{\|x_i - c_\sigma\|^2}{l_\sigma}\right)$$

It is possible to show that for a valley-type Gaussian profile eventually the swarm will surround/enclose the global minimum of the profile, which is stated in the following result.

Lemma 9.4:

Consider a foraging swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with interagent interaction potential $J(x)$ which satisfies Assumption 9.1. Assume that the resource profile $\sigma(\cdot)$ of the environment is given by Equation 9.25 with $A_\sigma > 0$. Then, as $t \to \infty$ we have $c_\sigma \in conv\{x_1, \ldots, x_N\}$ where conv stands for the convex hull (i.e., the agents encircle the minimum point of the profile c_σ).

Proof. Since $A_\sigma > 0$ from Theorem 9.3 we know that as $t \to \infty$ we have $(x(t), v(t)) \to \Omega_e$. Therefore, as in the case for quadratic profile with $A_\sigma > 0$ Equation 9.21 is satisfied. Summing up Equation 9.21 for all i we have

$$-\sum_{i=1}^{N} \nabla_{x_i} \sigma(x) = -\frac{A_\sigma}{l_\sigma} \sum_{i=1}^{N} (x_i - c_\sigma) \exp\left(-\frac{\|x_i - c_\sigma\|^2}{l_\sigma}\right) = 0$$

which, if we rearrange, we get

$$c_\sigma = \frac{\sum_{i=1}^{N} x_i \exp\left(-\frac{\|x_i - c_\sigma\|^2}{l_\sigma}\right)}{\sum_{i=1}^{N} \exp\left(-\frac{\|x_i - c_\sigma\|^2}{l_\sigma}\right)}$$

This expression can be written as

$$c_\sigma = \sum_{i=1}^{N} \alpha_i x_i$$

where $\alpha_i = \left(\exp\left(-(\|x_i - c_\sigma\|^2)/l_\sigma\right)\right) / \left(\sum_{i=1}^{N} \exp\left(-(\|x_i - c_\sigma\|^2)/l_\sigma\right)\right)$ and satisfies $0 < \alpha_i < 1$ for all $i = 1, \ldots, N$. Moreover, we have $\sum_{i=1}^{N} \alpha_i = 1$ implying the result. ■

Although this result is not as strong as its counterpart for the case of a quadratic profile (since it does not show convergence of $\bar{x}$ to c_σ) it is still a valuable result as it means that the agents will be enclosing the global minimum. Then, once a bound on the swarm size is established, one will know that all agents will locate themselves within that distance to the global minimum. Next, we analyze the cohesiveness of the swarm.

Theorem 9.5:

Consider a foraging swarm consisting of agents with dynamics in Equation 9.1 and with control input in Equation 9.18 with interagent interaction potential $J(x)$ in Equation 9.6 (which satisfies Assumption 9.1 and has linear attraction and bounded repulsion). Assume that the resource profile $\sigma(\cdot)$ of the environment is given by Equation 9.25 with $A_\sigma > 0$ (and whose gradient is bounded by $\bar{\sigma}$ in Equation 9.26). Then, as $t \to \infty$ we have $x_i(t) \to B_\varepsilon(\bar{x}(t))$ for all i, where

$$B_\varepsilon(\bar{x}(t)) = \left\{ y(t) : \|y(t) - \bar{x}(t)\| \leq \varepsilon = \frac{\bar{\sigma}}{Na} + \frac{b}{a}\sqrt{\frac{c}{2}} \exp\left(-\frac{1}{2}\right) \right\}$$

Proof. Since $A_\sigma > 0$, from Theorem 9.3 we know that as $t \to \infty$ we have $(x(t), v(t)) \to \Omega_e$. Therefore, as in the case for quadratic profile with $A_\sigma > 0$ we have Equation 9.21 satisfied for all agents i. For our

potential function we have

$$-\nabla_{x_i}\sigma(x_i) - \sum_{j=1,j\neq i}^{N}\left[a - \frac{bc}{2}\exp\left(-\frac{\|x_i - x_j\|^2}{c}\right)\right](x_i - x_j) = 0$$

which on rearranging can be written as

$$aNe_i = -\nabla_{x_i}\sigma(x_i) + b\sum_{j=1,j\neq i}^{N}\exp\left(-\frac{\|x_i - x_j\|^2}{c}\right)(x_i - x_j)$$

from which using the bound in Equation 9.26 and overbounding the second term one can obtain

$$\|e_i\| \leq \frac{\bar{\sigma}}{Na} + \frac{b}{a}\sqrt{\frac{c}{2}}\exp\left(-\frac{1}{2}\right)$$

and this concludes the proof. ■

One can see from above that besides the parameters of the potential function a, b, and c (which affect the swarm size similarly to the case of aggregation) the size of the swarm depends also on the bound of the gradient of the resource profile. In particular, larger $\bar{\sigma}$ (fast changing landscape) leads to a larger swarm.

9.5 Simulation Examples

In this section, illustrative numerical simulation examples will be given to provide better insights into dynamics of swarms. We will use $n = 2$ or $n = 3$ in the examples for easy visualization.

9.5.1 Aggregation

First we consider the case of aggregating swarms. Figure 9.2a shows the paths of the agents in a swarm of $N = 21$ agents with randomly generated initial agent positions within region $[0, 10]^3$ and zero initial velocities. The damping parameter was set to $k = 20$ for all agents. The initial and final positions of the agents are represented with squares and circles, respectively, while their paths are shown with dots. It is easily seen from Figure 9.2a that the agents move toward each other and form a cohesive cluster. The

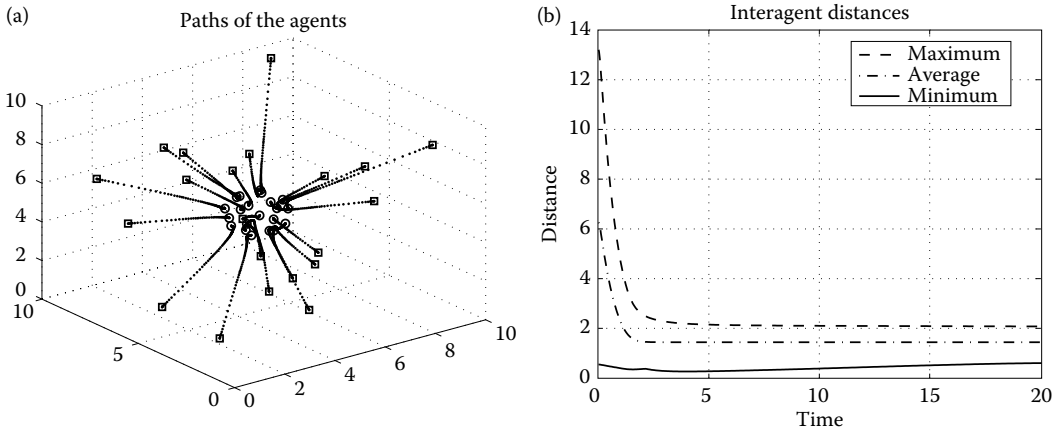

FIGURE 9.2 Aggregating swarm (zero initial velocities). (a) Agent paths. (b) Minimum, maximum, and average distances.

centroid of the swarm is stationary for all time (although not shown in the plots). Figure 9.2b shows the minimum, the maximum, and average distances between the agents in the swarm. As one can see due to the repulsion some distance between agents is preserved and no collisions occur. In these simulations, the potential function $J(x)$ in Equation 9.6 with parameters $a = 1$, $b = 10$, and $c = 1$ was used. For these values of the parameters, the bound on the swarm size can be computed as $\varepsilon \approx 4.29$. Note that the actual swarm size is much smaller than this since ε is a conservative bound. Smaller values of the damping parameter k result in a more oscillatory behavior but the final arrangement of the agents is similar to the one shown in Figure 9.2.

Figure 9.3 shows results for another simulation with the same parameters except that the initial velocities of the agents were set at random within the region $[-20, 20]^3$. The non-zero initial velocities result in initial bending of the agent trajectories as can be seen from Figure 9.3a. This bending also results in the motion of the centroid which is shown in Figure 9.3b with stars (the dark thick line). The final positions of the agents are also shown in Figure 9.3b. The rest of the motion is similar to the previous case. Also smaller values of the damping parameter k results in larger motion of the centroid (since the effect of the initial velocities dies out slowly than for larger k and results in slower recovery).

9.5.2 Formation Control

As a second example let us consider the case of formation control. Assume that there are six agents in the swarm which are required to create a formation of an equilateral triangle with three of the agents in the middle of each edge and distances between two neighboring agents equal to 1. For this case, we again use the potential function in Equation 9.6 but with pair-dependent interagent interaction parameters as in Equation 9.16. In particular, we choose $\delta_{ij} = 1$, $\delta_{ij} = 2$, or $\delta_{ij} = \sqrt{3}$ for different pairs (i, j) of individuals depending on their desired relative location in the formation. To achieve these values we used $b = 20$ and $c = 0.2$ for all agents and calculated a_{ij} based on $a_{ij} = b \exp\left(-\delta_{ij}^2/c\right)$. We used $k = 1$ as a damping parameter in the controllers of all agents. Figure 9.4a shows the trajectories of the agents together with the formed geometric shape. Initial positions of the agents were chosen randomly and the initial velocities were set to zero. As one can see, the agents move and create the required formation. Figure 9.4b shows the plot of interagent distances. As one can easily notice the interagent distances converge to the desired values of $d_{ij} = 1$, $d_{ij} = 2$, and $d_{ij} = \sqrt{3}$. Moreover, as can be seen from the figure, collisions between agents are avoided (no interagent distance goes to zero).

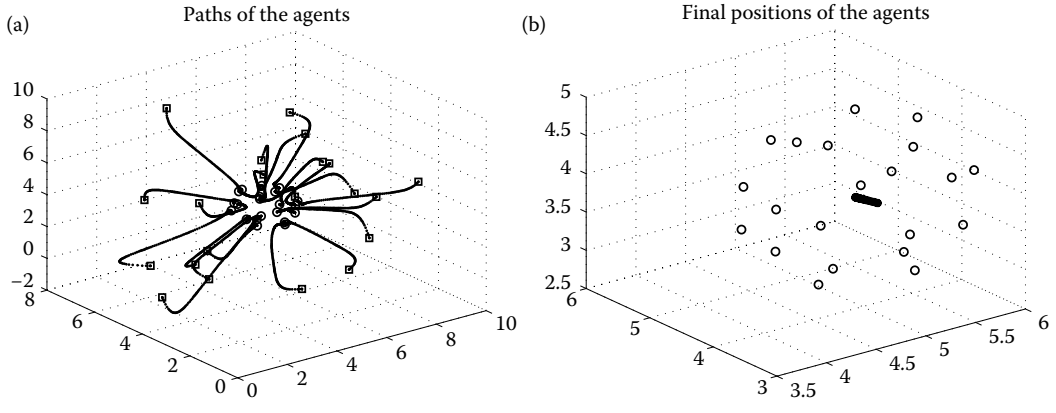

FIGURE 9.3 Aggregating swarm (random initial velocities). (a) Agent paths. (b) Final agent arrangement.

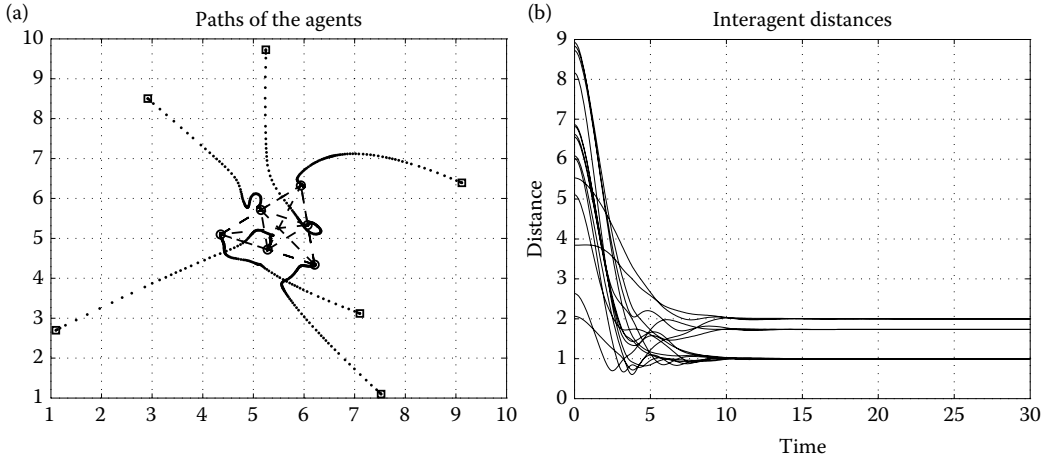

FIGURE 9.4 Equilateral triangle formation by 6 agents. (a) Agent paths. (b) Interagent distances.

9.5.3 Social Foraging

Finally, we perform illustrative numerical simulations of a foraging swarm moving in the profiles discussed in the preceding sections. In all the simulations, the primary simulation region is the region $[0, 30] \times [0, 30]$ in the space. The potential function in Equation 9.6 is used as the interagent interaction potential.

9.5.3.1 Plane Profile

Figure 9.5a shows the paths of the agents in a plane resource profile with $a_\sigma = [0.1, 0.2]^\top$. The parameters of the interagent interaction potential were chosen as $a = 1$, $b = 20$, and $c = 0.2$ and the simulations are performed with $N = 21$ agents. For this simulation, the damping parameter was set to $k = 2$ for all agents. One can easily see that, as expected, the swarm moves along the negative gradient $-a_\sigma$. Note that initially some of the individuals move in a direction opposite to the negative gradient. This is because the interagent attraction is much stronger than the intensity of the resource profile. By decreasing the attraction (or increasing the repulsion) this can be avoided. Figure 9.5b shows the motion of the centroid,

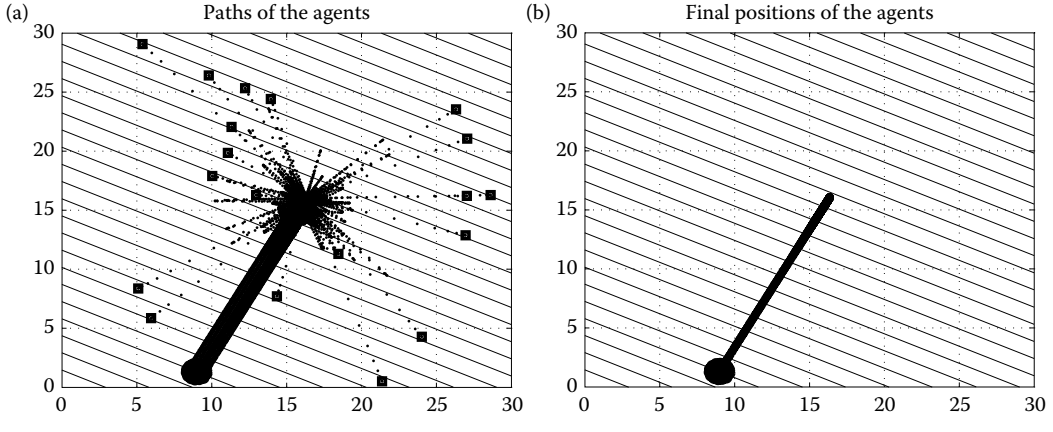

FIGURE 9.5 Swarm motion in a plane profile. (a) Agent paths. (b) Motion of the centroid.

which, as one can notice, moves along the negative gradient of the profile. As expected, increasing the damping parameter k slows down the motion of the swarm (simulations for this case are not shown).

9.5.3.2 Quadratic Profile

We will show two different simulations for a quadratic profile. In both of these simulations we used a quadratic resource profile with extremum at $c_\sigma = [15, 20]^\top$ and magnitude $A_\sigma = 2$ and $a = 1$, $b = 20$, and $c = 1$ as the parameters of the interagent interaction potential and the simulations are performed with $N = 11$ agents. Initial agent positions are generated randomly within the region $[0, 10] \times [0, 10]$ and the initial agent velocities are set to zero.

Figure 9.6 shows the result for a simulation with damping parameter $k = 3$ for which we have $k^2 - 4A_\sigma > 0$ implying real eigenvalues for the system matrix. The paths of the agents can be seen in Figure 9.6a. As one can easily notice, the agents move toward the global minimum and enclose/surround it. The motion of the centroid with respect to time is shown in Figure 9.6b. It converges to the global minimum at $c_\sigma = [15, 20]^\top$ as expected (the two curves in the figure represent the x and the y dimensions).

Figure 9.7 shows the result for another simulation with damping parameter $k = 1$ for which we have $k^2 - 4A_\sigma < 0$ implying complex eigenvalues for the system matrix. Here again the agents converge to a small region around the global minimum and the centroid converges to the minimum itself. However, the motion of the centroid is underdamped compared to the previous case which was overdamped. Similar characteristics can be seen in the agent trajectories as well.

Performing simulations with $A_\sigma < 0$ (hill-type quadratic profile, simulations not shown) reveals that the swarm stays cohesive and diverges away from the centroid. The only case in which the swarm might get stuck around the global maximum is the case in which $\bar{x} = c_\sigma$ initially. Note that this location is an unstable equilibrium point for the centroid and even very small perturbations will help the swarm escape from the deadlock.

9.5.3.3 Gaussian Profile

Results similar to those for the quadratic profile are also obtained for the Gaussian profile. The profile is set to have global minimum at $c_\sigma = [10, 15]^\top$ and magnitude $A_\sigma = 100$ and spread $l_\sigma = 20$. Initial agent

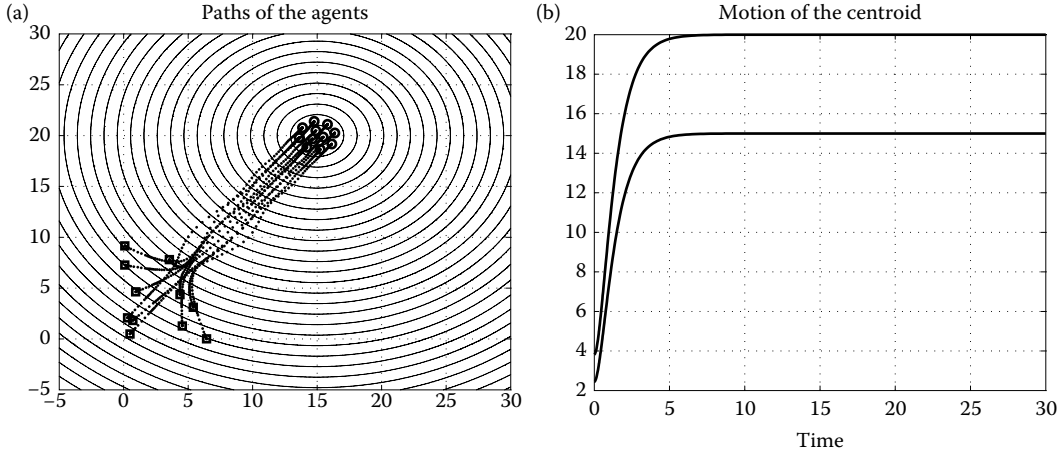

FIGURE 9.6 Swarm motion in a quadratic profile with $k^2 - 4A_\sigma > 0$ (real eigenvalues). (a) Agent paths. (b) Motion of the centroid.

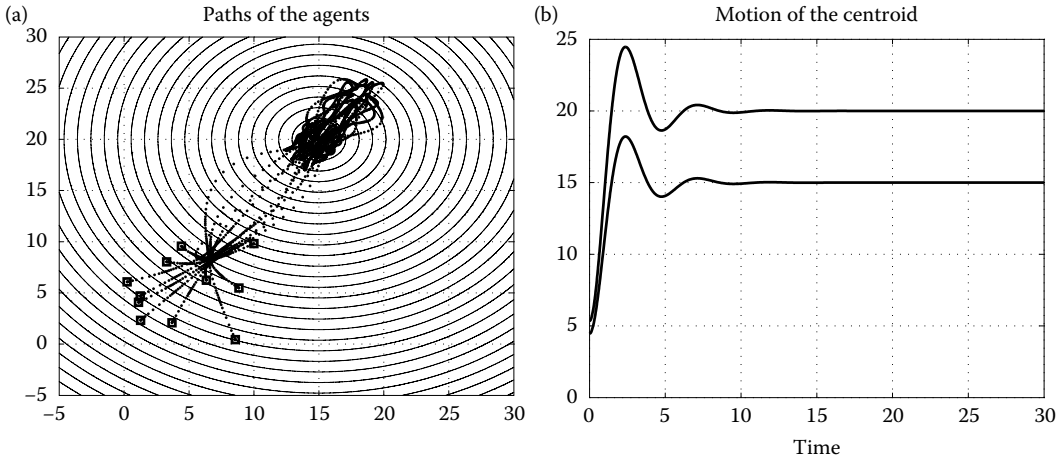

FIGURE 9.7 Swarm motion in a quadratic profile with $k^2 - 4A_\sigma < 0$ (complex eigenvalues). (a) Agent paths. (b) Motion of the centroid.

positions are generated randomly within the region $[0, 15] \times [0, 15]$ and the initial agent velocities are set to zero. The other conditions such as the number of agents, potential function and controller parameters are set to the values used for the quadratic profile. Figure 9.8 shows a simulation result for a case in which $k = 3$ was used as a damping coefficients. As one can see, the agents converge in the vicinity of and enclose the global minimum as predicted by the analysis. One can also see that the motion of the centroid is similar to the case of the quadratic profile and we have $\bar{x}(t) \to c_\sigma$. This is a nice observation although not analytically proven. Using the fact that the agents converge to a small region around c_σ and linearizing the effect of the resource profile on the motion of the agents and the centroid within that region one can also show analytically that $\bar{x} \to c_\sigma$ for the linearized system. Similar results are also obtained for the case in which $k = 1$ is used, which is shown in Figure 9.9. As one can see, the agents again converge in the vicinity of and enclose c_σ. Moreover, the centroid converges to c_σ and has an underdamped response (similar to the case of the quadratic profile).

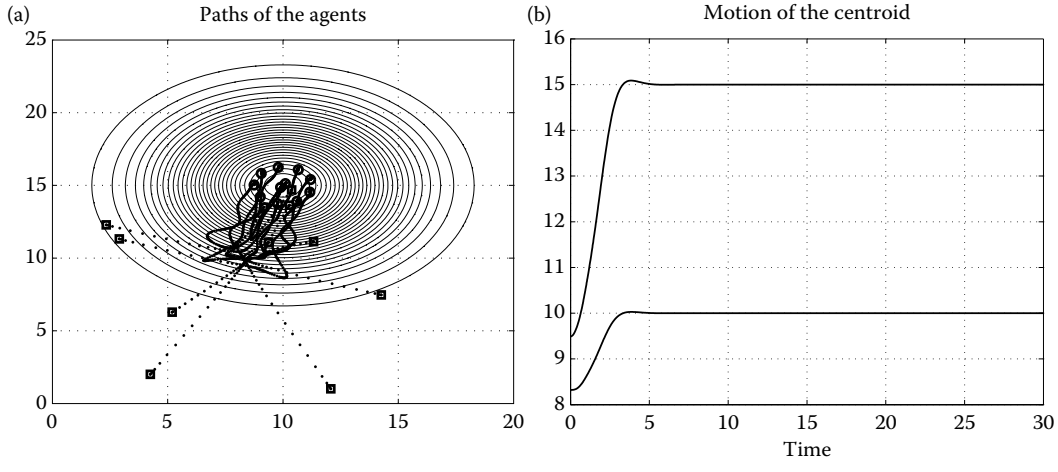

FIGURE 9.8 Swarm motion in a Gaussian profile for $k = 3$. (a) Agent paths. (b) Motion of the centroid.

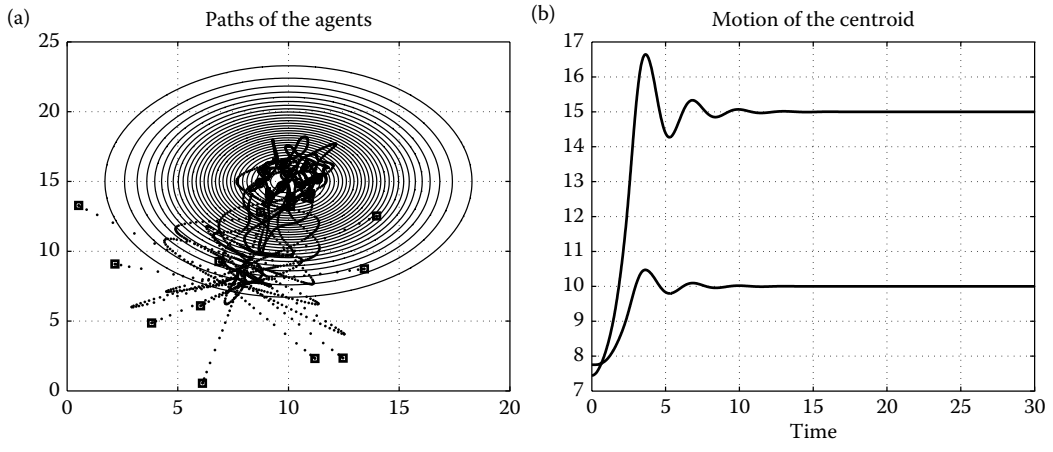

FIGURE 9.9 Swarm motion in a Gaussian profile for $k = 1$. (a) Agent paths. (b) Motion of the centroid.

9.5.3.4 Multimodal Gaussian Profile

Finally, we would like to present a simple simulation for a case which was not treated analytically, which is the case of a multimodal Gaussian profile. The plot of an example multimodal Gaussian profile, which is also used in the simulation here, is shown in Figure 9.10a.

Its expression is in the form

$$\sigma(y) = -\sum_{i=1}^{M} \frac{A_{\sigma i}}{2} \exp\left(-\frac{\|y - c_{\sigma i}\|^2}{l_{\sigma i}}\right) + b_\sigma, \tag{9.27}$$

where $c_{\sigma i} \in \mathbb{R}^n$, $l_{\sigma i} \in \mathbb{R}^+$, $A_{\sigma i} \in \mathbb{R}$ for all $i = 1, \ldots, M$, and $b_\sigma \in \mathbb{R}$. Note that it has several minima and maxima. In particular the function in Figure 9.10a is composed of $M = 10$ Gaussian components among which the minimum of the "largest" valley-type Gaussian is located at $c_{\sigma i} = [15, 5]^\top$ with a "magnitude" of $A_{\sigma i} = -20$ and a spread of $l_{\sigma i} = 10$ (and the global minimum is located very near to $[15, 5]^\top$). The plot in Figure 9.10b shows an example run for which the initial positions were set randomly in the whole region and the initial velocities were set to zero. As can be seen from the figure, the agents

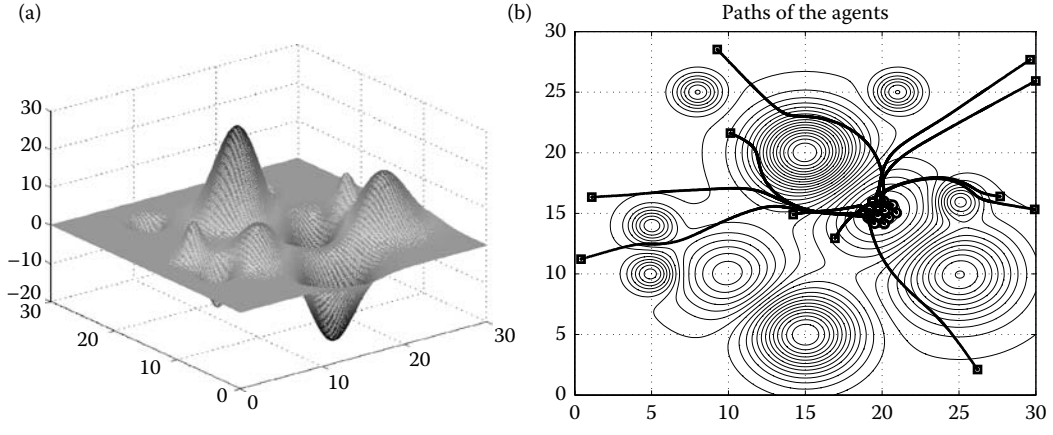

FIGURE 9.10 Simulation for a multimodal Gaussian profile. (a) The resource profile. (b) Agent paths.

form a cohesive swarm and move to a local minimum while avoiding the local maximum regions of the profile. Different behavior can also be obtained by varying the potential function and the profile parameters.

9.6 Further Issues and Related Work

The results in this chapter were obtained using Lyapunov analysis. Since it is based on worst case analysis the bounds obtained are conservative and the actual sizes of the swarms are usually smaller than the analytically obtained upper bounds. Also, in this chapter it was assumed that the agents can sense the positions of all the other agents in the swarm. Therefore, the interaction topology of the swarms considered in this chapter can be represented with a complete graph. Moreover, we assumed that the interagent interactions are reciprocal. Relaxing these assumptions (such as assuming general or time-varying interaction topology or nonreciprocal interactions) may lead to different and interesting behavior and there are already existing works on these topics [6]. Time varying or switching interactions would be more realistic from a biological perspective as well as possibly easier to implement and more scalable from a multirobot systems perspective. Nonreciprocal interaction will possibly lead to swarms drifting in the space even without the effect of the environment.

Other issues which can be considered are imperfections such as measurement errors, time delays, or model uncertainties. A work toward that direction and very much related to the results in this chapter can be found in [5] where social foraging in a noisy environment is considered. Another work which uses a realistic fully actuated model for agent dynamics (in contrast to single integrator or double integrator models) and imperfections such as agent model uncertainties and a robust control strategy to suppress these and to achieve swarm behaviors such as those considered in this chapter is the work in [7]. There are also other works considering nonholonomic unicycle agents with or without model uncertainties and achieving swarm behavior similar to those discussed in this chapter or related behavior such as cyclic pursuit.

There are related problems which have been considered in the multiagent dynamic systems research community. One of these problems is the flocking problem, in which beside cohesiveness the agents are required to achieve velocity matching and heading alignment [8]. Another related problem is the problem of distributed agreement (or consensus or rendezvous) which has been extensively considered in the literature [9]. Resource allocation and task assignment in swarms of cooperative agents are also topics which have been considered and are still under consideration by the research community.

Problems which may need further attention are dynamics of swarms consisting of different and possibly ńoncooperative or even competing agents such as predator–prey models and the interactions between them. To the best of our knowledge this topic has not been explored to the same extent as the topic of swarms with identical agents. Moreover, most of the swarm models considered in the literature are deterministic. Incorporating probabilistic components to the swarm dynamics might lead to further horizons of research. It is known that connectivity can greatly affect the performance of the cooperative robotic system [10]. Therefore, in robotic swarms where the communication (and therefore interaction) range is limited in order to facilitate cooperation it is important to preserve connectedness in the swarm. This is a topic of on-going research. For other relevant problems and existing approaches one can consult [1]. Also, the book [11] provides a deeper treatment of concepts related to swarm stability, optimization, and control.

The results in this chapter are directly related to the results in [2–4]. In fact, they are straightforward extensions of the subset of the results in [2–4] from a single-integrator agent model to a double-integrator newtonian agent model. Therefore, we tried to closely follow the treatment in those works. Similar treatment can be found in [12] where the authors have also closely followed the analysis in [2–4] and obtained results very similar to part of the results discussed in this chapter.

References

1. V. Gazi and B. Fidan. Coordination and control of multi-agent dynamic systems: Models and approaches. In E. Sahin, W. M. Spears, and A. F. T. Winfield, eds., *Proceedings of the SAB06 Workshop on Swarm Robotics*, Lecture Notes in Computer Science (LNCS) 4433, pp. 71–102. Springer-Verlag, Berlin, 2007.
2. V. Gazi and K. M. Passino. Stability analysis of swarms. *IEEE Trans Automatic Control*, 48(4):692–697, April 2003.
3. V. Gazi and K. M. Passino. A class of attraction/repulsion functions for stable swarm aggregations. *Int. J. Control*, 77(18):1567–1579, December 2004.
4. V. Gazi and K. M. Passino. Stability analysis of social foraging swarms. *IEEE Trans. Systems, Man, and Cybernetics: Part B*, 34(1):539–557, February 2004.
5. Y. Liu and K. M. Passino. Stable social foraging swarms in a noisy environment. *IEEE Trans. Automatic Control*, 49(1):30–44, 2004.
6. W. Li. Stability analysis of swarms with general topology. *IEEE Trans. Systems, Man, and Cybernetics: Part B*, 38(4):1084–1097, August 2008.
7. V. Gazi. Swarm aggregations using artificial potentials and sliding mode control. *IEEE Trans. Robotics*, 21(6):1208–1214, December 2005.
8. R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Automatic Control*, 51(3):401–420, March 2006.
9. L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Automatic Control*, 50(2):169–182, February 2005.
10. J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Trans. Automatic Control*, 49(9):1465–1476, September 2004.
11. V. Gazi and K. M. Passino. *Swarm Stability and Optimization*. Springer-Verlag, Berlin, 2011. to appear.
12. D. Jin and L. Gao. Stability analysis of swarm based on double integrator model. In D.-S. Huang, K. Li, and G. W. Irwin, eds., *Proceedings of ICIC 2006*, Lecture Notes in Bioinformatics (LNBI) 4115, pp. 201–210. Springer-Verlag, Berlin, 2006.

III

Industrial

10

Control of Machine Tools and Machining Processes

Jaspreet S. Dhupia
Nanyang Technological University

A. Galip Ulsoy
University of Michigan

10.1 Introduction

The manufacturing industry has a significant impact on a country's economic prosperity. Its importance to the U.S. economy can be judged by the fact that at the close of the twentieth century, manufacturing accounted for one-fifth of the nation's gross domestic product and employed 17% of the U.S. workforce. Control methods and technologies are utilized in the manufacturing industry at several levels. At the system level, manufacturing controls are used to achieve maximum efficiency and to eradicate waste wasted resources. Techniques such as lean manufacturing and six-sigma are used at the machining systems level to achieve such objectives. However, the focus of this chapter is primarily on the various control methodologies applied at the machine level. A manufacturing plant is typically composed of many, and varied, machine tools. Machine tools are used for machining operations, where unnecessary material is removed from the workpiece to produce the desired part shape.

Control of machine tools and machining processes is a mature technology. Research on chip formation dates back to the eighteenth century. The early work by F.W. Taylor in 1906, in which he introduced an empirical approach to metal cutting, is still regarded as significant in metal cutting research and applications [1]. Taylor investigated the effect of tool material and cutting conditions on tool life during roughing operations, with the main objective being to determine the empirical laws that allow optimum cutting conditions to be established. Eugene Merchant ushered in a more scientific approach with his fundamental work in mechanics of cutting processes based on shear plane models of machining [2].

The most important development in the control of machine tools began with the introduction of numerically controlled (NC) and computer numerically controlled (CNC) machines in the 1960s. Shortly after World War II, John Parsons at the Propeller Lab at Wright-Patterson Air Force Base envisioned the

use of mathematical data to actuate a machine tool. In June 1949, the US Air Force funded a program to develop a mathematical or numerical control system for machine tools. This led to the introduction of the first commercial computer-controlled or NC unit for machine tools by Bendix Corporation in 1955. With the advent of minicomputers, CNC machine tools gained wide acceptance in commercial use and prompted many advances in servo controls and interpolators. Two broad techniques employed in this area have been (a) point-to-point (PTP) and (b) contouring CNC systems [3].

A common drawback of these machines is that their process control variables, such as machining feed and speeds, must be prescribed by a part programmer and, consequently, depend on his or her process knowledge and experience. In order to minimize the chance of a tool failure or unacceptable part quality, the part programmer must consider the most adverse cutting conditions and select conservative values of the process control variables. Consequently, the production rates achievable in practice tend to be smaller than ideally possible. The availability of the processing power of an onboard CNC computer led to the developments in process control techniques, also known as adaptive control (AC) in the machine tool literature. Here the process variables, usually the feed, are adjusted continuously to achieve maximum safe production rates [4,5]. Tool wear or tool breakage are usually the main causes of machine downtime. The use of CNC machine tools enabled the addition of monitoring functions. Tool wear and breakage have usually been monitored by acoustic emission (AE) sensors, tool temperature, static and dynamic cutting forces, vibration signature using accelerometers, and miscellaneous methods using ultrasonic and optical measurements, workpiece surface finish quality, workpiece dimensions, stress/strain analysis, and spindle motor current [6]. However, such process control and monitoring functions, while tested in laboratories, have not been widely adopted in commercial use.

In complex manufacturing machines, many options exist to deploy the available resources to perform tasks that lead to the desired manufacturing purpose, resulting in various machine behaviors. Supervisory machine control may be used to decide when to perform which tasks using which resources. Researchers [7] have developed a theory for supervisory control. In that theory, the system under control is described using finite state machines. Supervisory machining control is an intelligent regulation of multiple complex modules in a machining operation. The supervisory controller regulates the activity of the machining modules to ensure that each module performs properly and that adverse interactions between modules do not occur.

This section has provided an introduction to the various important control applications in machine tools. These applications can also be represented in terms of hierarchical control levels as illustrated in Figure 10.1. The lowest level is servo (or machine) control, where the motion of the cutting tool relative to the workpiece is controlled. Next is a process control level, where process variables such as cutting forces and feed rates are controlled to maintain high production rates and good part quality. Finally, the highest level is a supervisory one, which directly measures product-related variables such as part dimensions and surface roughness. The supervisory level also performs functions such as chatter detection, tool monitoring, machine monitoring, and so on. The following sections discuss applications in each of these

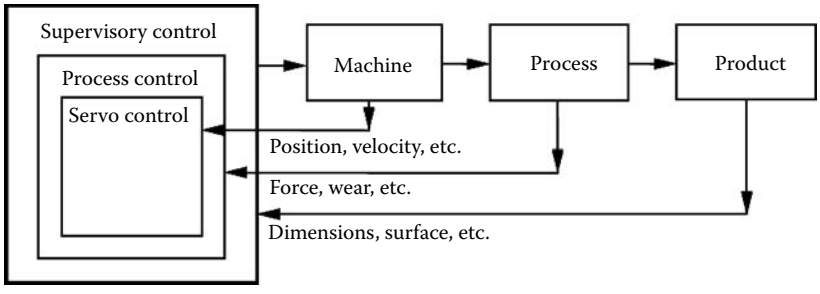

FIGURE 10.1 Levels of machine controls.

control levels, in more detail, and provide examples of important research in each of these areas. In the next section, the lowest level of machine control, that is, the servo control is discussed. Afterward, the machining process is described. After reviewing the major research activities in the monitoring of tool wear and failure, the machine process control is described. Finally, this chapter considers the supervisory control level, in which the appropriate process control strategies are selected based on the machine state.

10.2 Servo Control

Servo motors drive the motion stages of machine tools. The control of the machine motion is classified as [3] (1) point-to-point (PTP) and (2) contouring (or continuous-path). In a PTP system the controller moves the tool to a required position, where the process operation is performed. The trajectory between the points is not controlled. PTP control is required in applications such as drilling and spot welding, and is the simpler of the two cases. It has much in common with PTP servo control problems in other application areas, such as robotics. However, in machining, the accuracy requirements are typically more stringent (e.g., 0.01 mm or better). Also, in case of machine tools, the cutting forces and heat generation must be considered to ensure accuracy in the PTP problem. Reductions in machine geometric and thermal errors can improve accuracies by an order of magnitude (e.g., from 200 to 20 μm) in medium to large machining centers [8].

The contouring problem, which aims to minimize the "contour error," is a more difficult problem from the control point of view. Contouring is required in processes such as milling, turning, and arc welding. The term "contour error" is used to denote the error component orthogonal to the desired trajectory (i.e., the deviation of the cutter location from the desired path). The contour error in machining a desired contour on a two-axis system is shown in Figure 10.2. A block diagram of a typical implementation of a two-axis CNC machine achieving a planar contour is shown in Figure 10.3 [9]. The reduction of contour errors can be done using three kinds of control strategies [3]. They are feedforward control, cross coupling control, and optimal control. Optimal control is further divided down into predictive control, adaptive

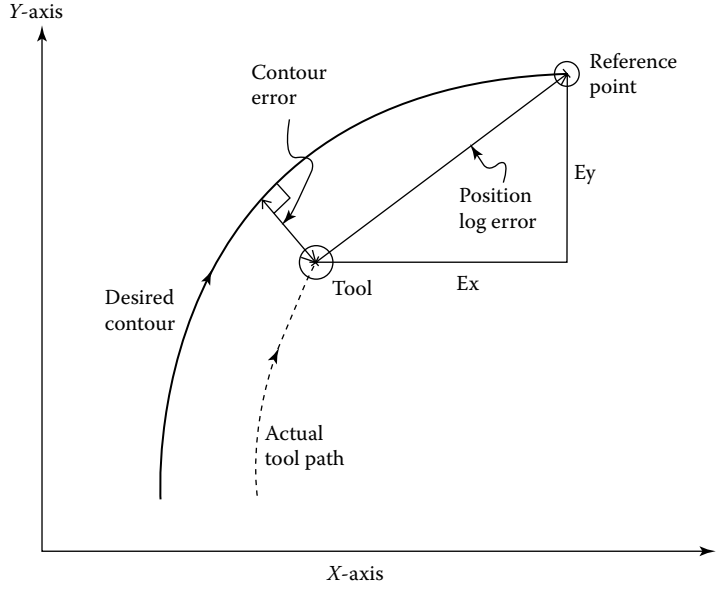

FIGURE 10.2 Two-dimensional contouring error.

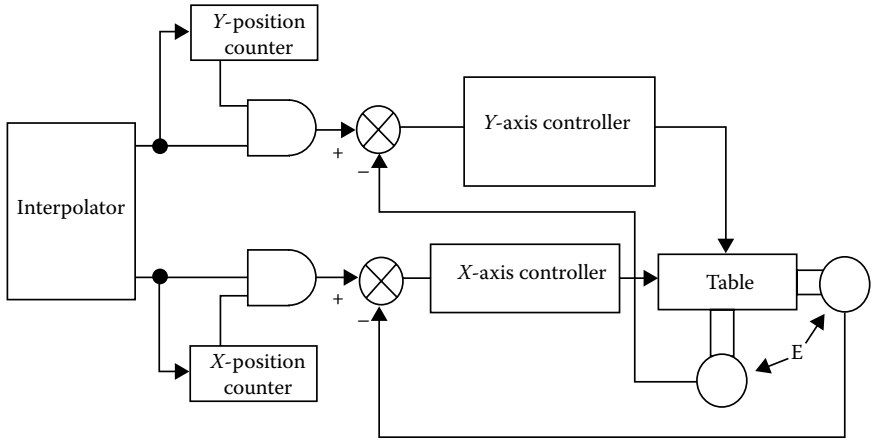

FIGURE 10.3 Two-axis contouring system.

control, and learning control. An overview of recent work in tracking and reduction of contour error is given in [10].

Feedforward controllers use disturbances that can be measured before they affect a plant, and use a model to determine the command signal required to minimize error. Typically, the command signal is used in combination with an inverted plant model. Tomizuka [11] introduced the idea of zero-phase error tracking control as a compromise when a plant model could not be inverted. While an improvement on standard pole-placement techniques, this strategy did not eliminate amplitude error or address the saturation problem of feedforward controllers. Weck [12] developed an "inverse compensation method" to low-pass filter the target path, thus reducing the saturation problem. This reduced the issue of loss of control due to command saturation, but it did not address requirements for coordinated motion. Feedforward is open-loop compensation; it does not use past or current path error information in calculating command signals, nor does it address issues with coordinated motion. As a result, feedforward techniques respond poorly in the presence of unknown disturbances or modeling inaccuracies.

Cross-coupled control (CCC) was developed by Koren in [13]. This method switched the focus of controllers from maintaining each axis at its target position, to minimizing path error. CCC estimates the point on the target path that is nearest to the plant's position, and uses this to determine the error for each axis. This error signal is then used in combination with any one of a number of controllers to control the coordinated position. The difficulty with this method, as described, is in determining the point on the target path nearest to the plant's position. This is accomplished by using various closed-form solutions, specific to the type of path being followed. This fails when two path segments meet, or for an arbitrary path. This seems to be an issue of developing appropriate formulae or applying sufficient computational force. A form of CCC was developed by Seethaler and Yellowley [14] which slows the movement of the target, in real-time, in response to excessive error on any individual axis. The effectiveness of CCC is dependent on the method used to calculate the point on the target path nearest to the plant's position, and on the control schemes used to control each axis once this point is computed. CCC can address a number of issues that other control schemes do not, such as coordinating motion among several axes. However, CCC lacks the ability to compensate for future path changes. Specifically, there is nothing to slow down a CCC system before encountering a corner or any other obstacle that could exceed the deceleration capability of one of the axes. This indicates that CCC should be used in combination with feed rate scheduling.

Optimal controllers create a command sequence which optimizes system performance over some horizon. Typically, this is the minimization of the difference between target and predicted output. The cost of the command signal also is often included. Currently, generalized predictive control (GPC) and

its variations represent advanced optimal control. GPC typically assumes a linear plant model. As such, it cannot address issues such as backlash and asymmetric performance. In its basic form, predictive control also does not consider command saturation issues. This was addressed by Tsang and Clarke [15] using "GPC with Input Constraints." However, GPC is computationally intense, and "GPC with Input Constraints" is too computationally intense for servo control. However, as computational power increases, these issues may disappear.

10.3 Machine Tools and Machining Processes

Machine tools can be classified in various ways. One classification is based on the different machining processes that may be carried out on them, for example, turning, boring, drilling, reaming, milling, planning and shaping, broaching, tapping and threading, and grinding. Machine tools can also be classified at the systems level based on the range of possible products that can be manufactured on them. They can either be dedicated machine tools which are built around a single part being produced over and over again; flexible machine tools which are built to perform a large variety of machining operations; or the relatively recent reconfigurable machine tools [16–18] that are built around a part family. Reconfigurable systems aim to achieve both efficiency and robustness of dedicated manufacturing systems as well as the ability to change production capability as dictated by market demands as in flexible manufacturing systems.

Regardless of the classification of machine tools, they are utilized for material removal operation, which is a shape transforming process, wherein unnecessary material is removed from the workpiece. The chip removal due to workpiece cutting tool interaction results in cutting forces. Various models have been developed to describe these cutting forces, which account for the frictional forces between the cutting tool, workpiece and the chip, and the stresses along with the resulting deformations developed in the chip. The commonly used models for evaluation of cutting forces have been developed by Ernst and Merchant [2] and Lee and Shaffer [19]. However, these models are quite complex, and not suitable for controller design. The structure of a typical static cutting force model used for controller design is [20]

$$F = Kd^{\beta} V^{\gamma} f^{\alpha}, \tag{10.1}$$

where F is the cutting force, K the specific cutting force coefficient, d the depth of cut, V the cutting speed, f the feed, and α, β, and γ the coefficients describing the nonlinear relationships between the force and the process variables (i.e., d,V, and f). During a cutting operation the parameters d, V, and f are selected by the operator, K is determined by the workpiece and tool properties, and the coefficients α, β, and γ are calibrated through curve-fitting of experimental data. Typically, control of machining forces is achieved by changing the feed online, as the depth of cut is fixed from the part geometry and the force–speed relation is weak (i.e., $\gamma \approx 0$). Therefore, these variables cannot be actively adjusted for force control. Static models are also used when considering a force per spindle revolution such as a maximum or average force. Such models are suitable for interrupted operations (e.g., milling) where, in general, the chip load changes throughout the spindle revolution and the number of teeth engaged in the workpiece varies during steady operation.

The first-order cutting force model, assuming a zero-order hold is

$$F = Kd^{\beta} V^{\gamma} \frac{1+a}{z+a} f^{\alpha}, \tag{10.2}$$

where a is the discrete-time pole that depends upon the time constant and the sampling period, and z is the discrete time forward shift operator. In addition to the other model parameters, a must be calibrated for each different operation. First-order models are typically employed when considering an instantaneous force, which is sampled several times per spindle revolution. Such models are suitable for uninterrupted

operations (e.g., turning) where, typically, a single tool is continuously engaged with the workpiece and the chip load remains constant during steady operation.

Direct measurement of cutting forces is usually not possible in industrial settings. Load cells or dynamometers are often used in laboratory settings for cutting force measurements. However, they are impractical for industrial applications. This has been one of the major impediments to the industrial adoption of force-based monitoring and control techniques. Cutting forces can be estimated from the electric current signal of the servo drive or spindle motors [21]. For this the power from the spindle and axis motors is typically monitored using Hall effect sensors, which are easy to install and guard from the process. However, due to the large masses of the motor drive, the signal typically has a limited bandwidth [22,23]. Piezoelectric accelerometers are also widely available in most production lines and are well suited to the harsh industrial environments. A robust method to identify machining forces during milling from acceleration measurements obtained using acceleration measurements is proposed in [24]. Evaluation of machining forces requires an inverse of the frequency response function (FRF) matrix of the machine structure. However, an ill-conditioned FRF matrix can amplify the measurement noise in acceleration signal and yield poor results. This challenge was overcome using regularization techniques. Figure 10.4 illustrates the experimental results from measured and estimated machining forces.

These cutting forces act as excitation forces for inducing vibrations during cutting, which leads to continuously varying cut width. Regenerative chatter can arise due to such interactions between the machine tool and workpiece, both of which get excited due to cutting forces and may result in excessive forces, rapid tool wear, tool failure, and scrap parts due to unacceptable surface finish [25]. The dynamics of the turning process is typically described by a single-degree-of-freedom model such as

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = Kd\left[f_n + x(t) - x(t-\tau)\right], \tag{10.3}$$

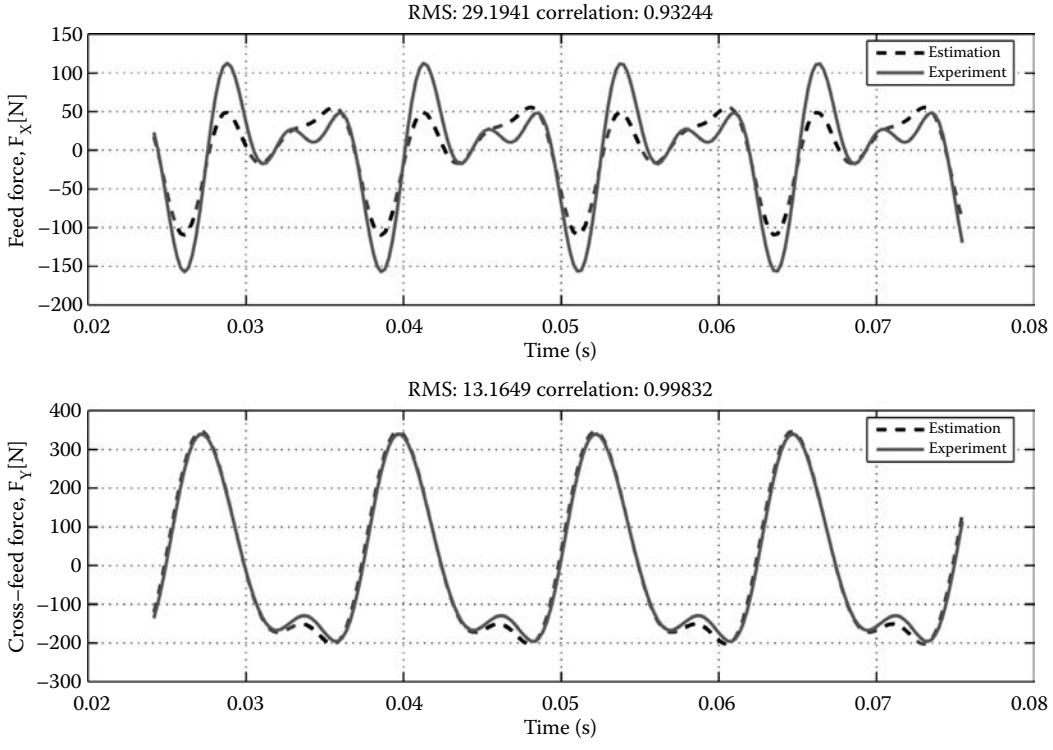

FIGURE 10.4 Estimated forces and measured machining forces using a dynamometer during milling.

where f_n is the nominal feed, x is the displacement of the tool in the feed direction, and τ is the time for one tool revolution, and m, c, and k are its effective mass, damping, and stiffness of the tool structure. The right-hand side of the equation defines the feed–force relationship assuming that the force is proportional to the instantaneous feed and the depth of cut, and does not explicitly depend upon the cutting speed. The left-hand side of the equation describes the vibration of tool structure. If the compliance of the workpiece is comparable with the tool, then the parameters m, c, and k have to be defined for the relative tool–workpiece deflections.

Equation 10.3 can be solved to obtain the stability regions of the machine tool operation. Machine chatter occurs when the system response becomes unstable. The presence of machine chatter is graphically represented on the stability lobe diagram that divides the rotational speed and the depth-of-cut plane into stable and unstable regions as shown in Figure 10.5 [25]. Usually the stability lobe diagram is determined using a linear chatter analysis developed by Budak and Altintas [26]. However, the linear milling model is an approximation. The effect of several nonlinearities such as force–feed relationship [27], intermittent cutting [28], variable time delays [29], structural nonlinearities [30,31], and so on, on stability lobe diagrams have recently been studied. While it is easy for an experienced machine tool operator to recognize chatter from the high-pitched sound or characteristic chatter vibration marks left on a workpiece, the automatic detection and suppression of chatter have been challenging. In-process chatter detection for milling using the sound spectrum was developed by Altintas and Chan[32]. These techniques relied on detecting components of spectral density at chatter frequency above a certain threshold value. Tarng and Li [33] created threshold values for the spectrum and the standard deviation of thrust forces and torque signals in machining operations. It should be noted that the tooth passage frequency contains significant energy and the process signal must be properly filtered if the tooth passage frequency is close to a dominant structural frequency. In grinding, usually the AE signals are combined with neural networks as a pattern recognition tool for detecting chatter [34,35]. In one of the methods [35], a back-propagation neural network was employed to detect chatter based on power spectra of the enveloped AE signal. A different approach to automatic chatter detection has been proposed recently [36], employing a scalar indicator, the coarse-grained entropy rate, calculated from the fluctuations of the normal grinding force.

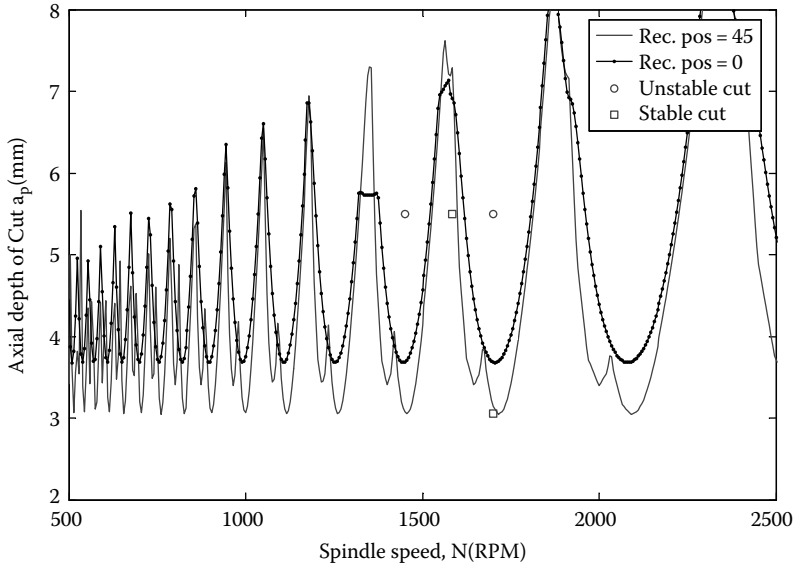

FIGURE 10.5 Stability lobe diagrams for arch-type reconfigurable machine tool while cutting AISI 1018 steel in full immersion.

Regardless of all the research in automatic detection of chatter, in industry, chatter is avoided by using the stability lobe diagrams to select operating parameters of the machine tool, that is, spindle speed and depth of cut, to obtain a specific material removal rate while maintaining stable machining operation. Thus, by properly choosing the machine operating parameters a good quality of product may be obtained while avoiding excessive tool wear. Research has also been carried out to suppress chatter vibrations during milling. A common technique to suppress chatter during milling is using spindle speed variation. In this technique, typically a small sinusoidal variation in spindle speed is superimposed on the nominal spindle speed [37,38].

Besides the vibration of machine tools and workpiece, dimensional and geometrical errors can also arise due to servo errors, misalignments, and thermal deformations of the machine tool. Extensive research on error compensation has been carried out to remove such errors. Research in error compensation for servo misalignment dates back to early 1960s. A good discussion on forecasting-based compensatory control for reducing systematic errors in machine tools is provided in [39]. When machine tools are used for a prolonged period of time, internal and external heat sources cause thermoelastic deformations leading to geometric inaccuracies in the workpiece. Thermal effects can contribute more than 50% to the overall error. The necessity of reducing the thermally induced error source was recognized in the early 1960s, and research in this field was pioneered by Bryan et al. [40,41]. Reference [42] gives a detailed description of important research carried out in the area of compensation for machine tools of errors arising from geometrical and thermal sources.

Small, undesirable metal fragments left on the workpiece, after the machining operation is complete, are known as *burrs*. Burrs cause improper part mating, accelerated device wear, and decreased device performance. Since it is typically impossible to avoid the formation of burrs, the designer strives to reduce the complexity of the subsequent deburring operation by minimizing the burr strength and ensuring that the burrs formed at workpiece locations are easy to access. Burrs form due to workpiece plastic deformation [43]. Burr measurement is typically performed offline by measuring the average height, base thickness, and toughness. Burr location and its accessibility are also of concern. Process variables are known to have a strong effect on the physical characteristics of burrs. If the depth of cut in a face milling operation is too small, the cutting tool will "push" the material over the side of the workpiece and form a large, strong burr on the workpiece edge. Without adequate models, one is left with empirical techniques to predict and control burr formation.

10.4 Monitoring and Diagnostics

Machine downtime is the duration of time during which no machining operation is being performed on a given workpiece. Machine downtime from certain sources such as transfer of workpieces or machine maintenance is unavoidable. However, monitoring of tool condition can be used to avoid the downtime due to excessive tool wear and breakage. Tool breakage is a major cause of unscheduled stoppage in a machining environment, and is costly not only in terms of time lost, but also in terms of capital destroyed [44]. Some estimates state that the amount of downtime due to tool breakage on an average machine tool is in the order of 6 to 8% [44], while others put the figure closer to 20% [45]. Even if the tool does not break during machining, the use of dull or damaged cutters can put extra strain on the machine tool system and cause loss of quality in the finished workpiece.

The earliest work on tool wear and tool life, which is still considered significant, was reported by F.W. Taylor in 1906 [1]. Taylor was interested in the application of piecework systems in machine shops, where a time allowance was set for a particular job and a bonus was given to the workman performing his tasks in the allotted time. To assist in the application of such a system, Taylor investigated the effect of tool material and cutting conditions on tool life during roughing operations. The resultant empirical law that

governs the relationship between the cutting speed and tool life as suggested by Taylor was

$$VT^n = C,$$

where V is the cutting speed and T is the tool life, and n and C are constants that are found experimentally and depend on feed rate, and tool and workpiece material. Modified Taylor equations include the effects of feed rate and the depth of cut, as well as interaction effects between these variables [46].

The cutting tool wear occurs due to tool–workpiece interaction in any one or combination of the following modes:

a. Adhesive wear associated with shear plane deformation.
b. Abrasive wear resulting from hard particles cutting action.
c. Diffusion wear occurring at high temperatures.
d. Fracture wear such as chipping due to fatigue.

Wear on the face of the tool that contacts the workpiece is termed flank wear, whereas crater wear occurs on the tool face that contacts the chips. Nose wear, or edge rounding, may occur through the abrasion wear mechanism on the cutting tool's major edges resulting in an increase in negative rake angle. Typical tool wear features are shown in Figure 10.6 [6]. The tool wears rapidly in the initial phase and then levels off to a constant rate during the steady phase. Finally, the tool enters an accelerated wear phase where it may eventually fail (Figure 10.7).

A good review of various methods to predict tool wear is given in [6,47]. Several methods using accelerometers, AE sensors, current/voltage readings, cutting force measurements, microphones, cutting speed, and vision systems have been used for monitoring the tool condition.

Acoustic emission is a very high-frequency oscillation or stress wave, generated due to deformation occurring when metals are cut or fractured. It is generally accepted that AE is linked to the plastic deformation process occurring during chip formation, due to the interaction between the workpiece and cutting tool. AE has been very successful in its application to tool monitoring during turning operations [48]. Its application to milling has been less straightforward [49]. The difficulty in applying AE signal analysis to milling is that pulse shock loading occurs during the entry and exit of each individual tooth to the workpiece. Using AE on its own to monitor the state of a cutting tool is a difficult task. AE is attractive for monitoring the state of a cutting tool together with an additional sensing method for increased reliability. Dornfeld [50,51] presented comprehensive reviews on the application of AE sensing techniques in manufacturing processes, particularly as applied to tool wear detection in machining.

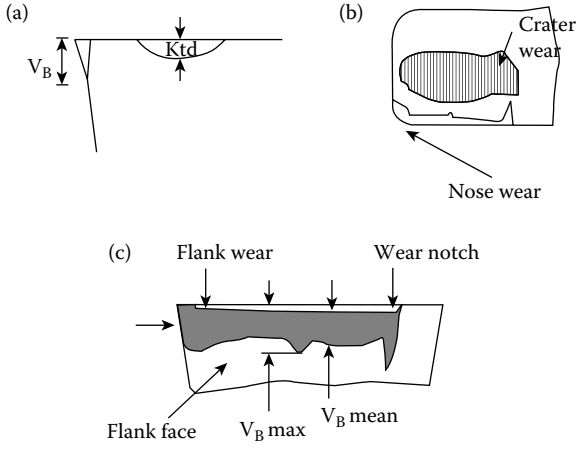

FIGURE 10.6 Typical wear features on a cutting tool during turning.

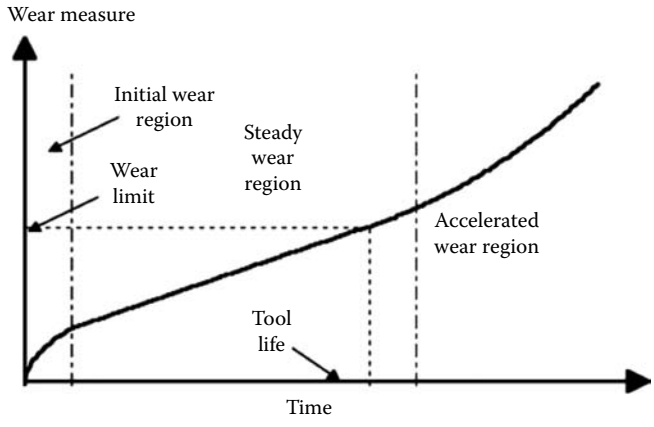

FIGURE 10.7 Typical flank wear versus time trend.

The coefficient of friction between tool and chip varies considerably due to changes in cutting speed and rake angle resulting in high pressures and temperatures. Chow and Wright [52] devised an online method for tool–chip interface temperature measurement in a turning process using a standard thermocouple inserted at the bottom of the tool insert. Experiments were conducted from which practical cutting data were collected for comparison with predicted interface temperatures from a theoretical model. The test cuts involved dry machining performed on plain steel tubes (AISI 1020) with coated and uncoated controlled contact tool inserts. Analysis of the experimental results obtained and verified by the theoretical model showed that an increase in the tool wear resulted in an increase in the cutting temperature which could be used for tool condition monitoring (TCM) during metal cutting. Recently, Choudhury et al. [53] experimentally correlated the relation between the flank wear and cutting zone temperature in turning where the temperature sensor was the naturally formed thermocouple between the tool and the workpiece. In this case, only the average temperature in the cutting zone is sensed. For practical applications such as online TCM, remote thermocouple sensing appears to be the only reasonable way to measure the workpiece–tool temperature, since a direct measurement of the tooltip or rake face temperature distribution cannot be obtained. Most currently available remote thermocouple sensor instruments can only allow either the cutting tool–workpiece interface or some other remote area temperature, to be measured and not the tool tip temperature. This process parameter, though a suitable tool wear indicator and desirable, is extremely difficult to measure accurately for online applications as in TCM due to the inaccessibility to the cutting area. Another challenge with temperature monitoring of cutting tool is that the chip carries approximately 90% of the energy dissipated during cutting, and will, thus, dominate the intensity of emitted radiation [54].

It has been widely established that variation in the cutting force can be correlated to tool wear [55–57]. Early attempts found that the feed and radial forces are more sensitive to tool wear than the cutting force. The radial force component was reported to be the most sensitive to nose wear, with the feed force and the radial force components affected by flank wear [58]. Similarly, flank wear was observed to correlate with the feed and cutting force components [55–57,59]. Force ratios can also be used to predict tool wear since they present a certain pattern as the tool wears [60]. The feed force to cutting force ratio was found to be sensitive to flank wear [59]. The physical characteristics of the dynamometer can seriously limit the physical size of the workpiece and, additionally, dynamometers are expensive. Thus, research has been done to indirectly measure the cutting forces to overcome such practical constrains [24]. They have also led to the development of a range of tool holder-mounted dynamometers that do not experience these limitations [61]. Cutting forces result in the excitation of machine tool vibrations which is also a related phenomenon.

TCM using vibration signals can be done with the use of accelerometers. Vibrations are also used, along with wavelet analysis, to create discrete hidden Markov models (HMMs) in [62] for turning. Feature vectors are extracted from vibration data, and converted into a symbol sequence for use with HMMs. HMMs could be further improved by combining them with wavelet-based statistical signal processing methods, as was demonstrated in [63]. Accelerometers are used in [64] as part of a multisensory artificial neural network trained fuzzy logic TCM system for milling. The mean, standard deviation and mean power of the vibration signal in 10 frequency ranges are measured. When combined with AE, power, and cutting force measurements, the system is able to detect and classify worn tools with a confidence level of at least 80%.

Apart from the work already described, other techniques that have been investigated for TCM are optical and machine vision systems, stress/strain measurements, workpiece dimensions, electric motor current/power measurement, magnetism, and ultrasonic methods.

10.5 Machining Process Control

Machine tool operators perform online and offline process control by adjusting feeds and speeds to suppress chatter, initiating an emergency stop in response to a tool breakage event, rewriting a part program to increase the depth of cut to minimize burr formation, and so on. Offline process control is performed at the process planning stage; typically by selecting process variables from a machining handbook or based on operator experience. Computer-aided process planning is a more sophisticated technique which, in some cases, utilizes process models offline to select process variables. The drawbacks of offline planning are the dependence on model accuracy and the inability to reject disturbances. Process automation can autonomously tune machine parameters (feed, speed, depth of cut, etc.) online and offline to substantially increase the machine tool's performance in terms of part tolerances and surface finish, operation cycle time, and so on. Process automation holds the promise of bridging the gap between product design and process planning, while reaching beyond the capability of a human operator.

Machine tool controllers consist of a programmable logic controller (PLC) that handles the sequencing and operator interface, and a microprocessor that coordinates the real-time control functions. The microprocessor architecture can be generally divided into three levels: servomechanism control loop, interpolator loop, and process control loop as shown in Figure 10.8. The servomechanism controllers regulate the velocity and position of individual axes and spindles and interpolators generate the reference positions for the axes. These functions are found in all modern CNCs. The process control loop is also referred to as AC [9] in the machine tool literature, but is not commonly available in today's CNCs, although it has been the focus of a tremendous amount of research due to its potential to significantly increase operation productivity and quality. The term AC is actually a misnomer, as machine process control may or may not be adaptive in the sense commonly used in the control theory literature [65,66]. The main objective of AC techniques is to increase productivity, for example, by increasing the metal removal rate (MRR). This is illustrated in Figure 10.9, where AC may be used to adjust the feed rate for a milling operation with variable depth and/or width of cut. Without the adaptive control, the smallest feed rate needs to be selected for the worst-case conditions for that particular part.

Adaptive control techniques [4,66], which include adaptive control with optimization (ACO), adaptive control with constraints (ACCs), and geometric adaptive control (GAC), set process variables to meet productivity or quality requirements. While ACO systems represent the most general class of AC systems, they are difficult to implement. The idea of ACO as means for process control developed with the well-known research project for milling conducted at Bendix in 1962–1964 under a U.S. Air Force contract [67]. This system was designed to perform an online maximization of an economic index of performance to determine the required feed rate and spindle speed in a milling operation based on measurements of the cutting torque, tool temperature, and the tool vibration. The index of performance as a function of feed, f (in./min), and spindle speed, N (rev/min), can be expressed as the following, where TWR is the

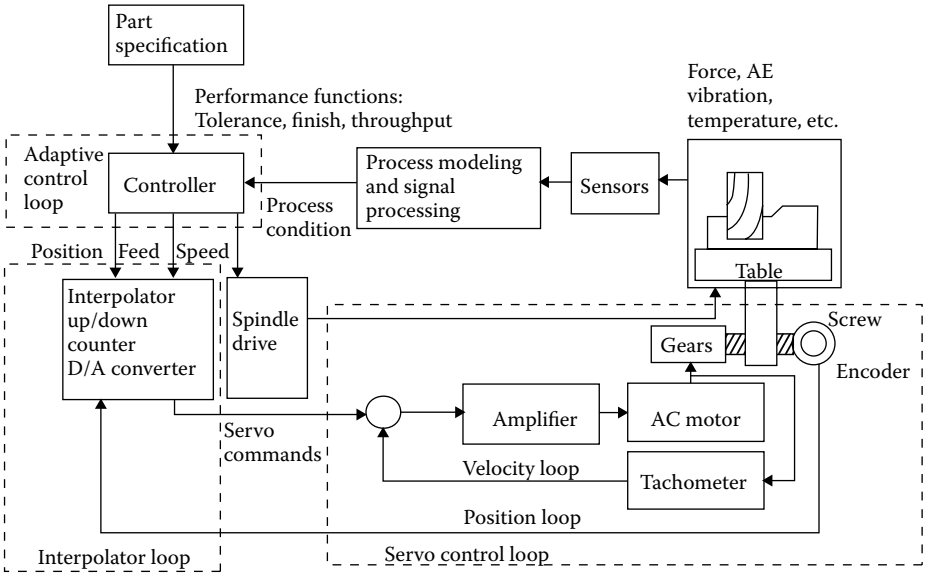

FIGURE 10.8 General scheme for machine tool control and monitoring.

tool wear rate (in./min), W_0, the terminal allowable width of flank wear (in.), C_1, the cost of machine and operator per unit time ($/min), C_2, the cost of tool and regrind per tool change ($/change), and t_l, the tool changing time (min):

$$J\left(f,N\right)=\frac{MRR}{\left[C_1+\left(C_1t_1+C_2\beta\right)TWR/W_0\right]}. \tag{10.4}$$

The constant β is an adjustable parameter in the range $0 \le \beta \le 1$, which determines the type of performance index J. If $\beta = 1$, then J represents the reciprocal of the cost per unit produced; if $\beta = 0$, then

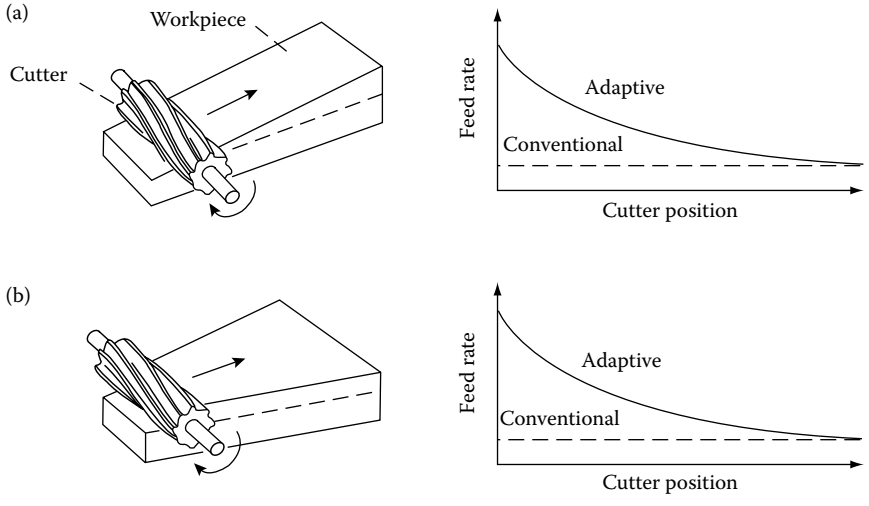

FIGURE 10.9 Comparison of the feed rate for adaptive and nonadaptive milling when cut varies: (a) variable depth of cut and (b) variable width of cut.

J represents the production rate; and for intermediate values of β, J represents a weighted combination of these two objectives, which can be adjusted to represent the profit (i.e., maximum production with minimum cost). The Bendix system was implemented successfully in the laboratory, but never worked robustly in an industrial environment. Another ACO system was reported for grinding in [68], which was used in an offline mode in industry. The poor acceptance of such systems in industry primarily is due to the need for an online tool wear sensor for the implementation of an ACO strategy [9]. Although full ACO systems could not be implemented successfully, suboptimal systems, which can provide most of the benefits of the ACO systems, have been developed. These are typically categorized as ACCs systems and GAC systems.

Most current commercial process control systems use ACCs. These systems take advantage of the fact that, under certain cutting conditions, the process optimization problem has an optimal solution, which occurs on a constraint boundary. For example, in rough cutting operations, the economic objective function typically is dominated by the need to maximize the MRR, which, in turn, requires the highest possible feed rates consistent with the tool breakage constraint. Thus, for particular tool geometry, this can be expressed as the following process control objective: "adjust the feed rate to maintain the reference cutting force, which is determined as the highest possible cutting force such that the tool is not in danger of breaking." There are ACC systems commercially available for turning, milling, and drilling, but they are not very widely used [69]. The reason for their poor industrial acceptance is the potential problems of controller instability and tool breakage, which can occur due to process parameter variations.

Geometric adaptive control [70] is suited for finishing operation where the economic process optimization problem is dominated by the need to maintain product quality (e.g., dimensional accuracy and/or surface finish). If the part dimensions and surface roughness can be measured, then the process control strategy can be expressed as: "based on the dimensional measurement, introduce a tool offset to compensate for the tool wear and adjust the feed rate to produce parts at the reference surface roughness value determined from the maximum allowable surface roughness constraint." The GAC systems have also not found widespread industrial acceptance, although the GAC system described in [71] was implemented in a manufacturing facility.

10.6 Supervisory Control and Statistical Quality Control

Much of the research in the area of machining process control has been focused on regulating a single process phenomenon, such as, force or chatter, using a single process variable, such as feed or spindle speed. However, there are many levels of control in a machining process control hierarchy. Recently, some research has focused on integrating such multiple controllers at different levels of machining process control hierarchy. Supervisory control includes functions such as selection of control strategies, sensor fusion, generation of reference commands for the process control level, tool breakage monitoring, chatter detection, and machine monitoring. Teltz and Elbestawi [72] proposed a hierarchical control system consisting of a supervisory level and a process level (force and chatter controller). The supervisory level monitored signal and alarm events and utilized an inference engine that searched a knowledge base to relate these events to recovery actions. Ramamurthi and Hough [73] used a machining influence diagram (MID) for supervision and applied it to a drilling operation. A knowledge base is tuned during a training phase and the MID is used to identify failures.

Supervisory control can be illustrated by the example illustrated in Figure 10.10, where an appropriate control strategy for the process control and reference signals for that strategy are selected based on the drill's depth. At the entrance phase, that is, phase I, supervisory controller selects the appropriate feed and speed control strategy, feed and speed references are selected based on the hole location error constraints. After the hole is initiated, the supervisory controller switches to a speed and torque control strategy, with reference values determined to maximize the material removal rate, while avoiding tool breakage

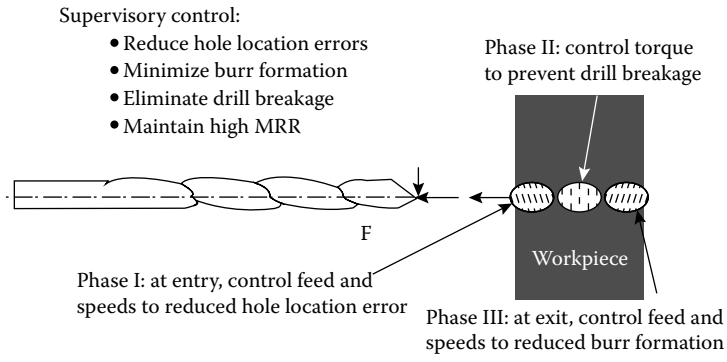

FIGURE 10.10 Supervisory control of a through hole drilling operation.

TABLE 10.1 Comparison of Drilling Control Strategies

	No Controller	Feed/Speed Controller	Torque/Speed Controller	Supervisory Controller
Machining time (s)	11.11	11.28	9.79	11.71
Burr rating	2.93	2.94	2.26	1.58
Hole location quality (in.)	4.43×10^{-3}	4.53×10^{-3}	6.28×10^{-3}	4.25×10^{-3}
Event stoppages	25	15	0	0

and considering tool wear rates. Finally, as the drilling of the through hole is nearly completed, the supervisory controller switches back to a feed and speed control strategy to minimize the burr formation. The results of using such a supervisory control strategy are illustrated in Table 10.1. This table compares experimental results obtained using four types of controllers: (1) no control (conventional approach where nominal feed and speed values are selected but not controlled during drilling), (2) feedback control of feed and speed, (3) feedback control of torque and speed, and (4) the supervisory controller (which combines (2) and (3) as described above). The comparison is made in terms of: (1) machining time for one hole, (2) burr rating, (3) hole location error (in terms of the pooled standard deviation of the hole location error), and (4) percentage of holes drilled with stoppage events. The machining time per hole is given in seconds, the burr rating ranges from 1 = very little burr formation to 5 = large burrs; the hole quality is given in terms of a pooled standard deviation with smaller values indicating better hole location accuracy. The percentage of stoppage events is for holes for which the torque exceeded a maximum allowable value for drill breakage and the process was stopped. In this application a hole location error pooled standard deviation of $< 4.5 \times 10^{-3}$ in., and a burr rating of <1.75 are required. It is also desired that there be no stoppage events, and that the machining time be minimized subject to these hole quality, burr, and breakage constraints.

The supervisory control strategy is the only one which meets the required hole quality and burr constraints, while eliminating stoppage events. It also yields a machining time very comparable to the uncontrolled or feed/speed control cases, and only slightly longer than the torque/speed controller. The results are average values based on a statistical study involving the drilling of 20 holes with each strategy in a randomized order [74–76]. These results clearly illustrate the potential advantages of a supervisory control strategy over each of the individual process control strategies (i.e., feed, speed or torque control), and additional experimental results are presented in [74–76].

10.7 Summary, Conclusions, and Future Directions

This chapter provides a brief overview of the important research carried out for control of machine tools and machining process over the last few decades. The control of machine tools may be classified into different levels of machine control. At the lowest level is the servo control, which takes the cutting tool from one orientation to another, while following a prescribed trajectory relative to the workpiece. The two major control strategies are the PTP and the contouring control strategy. In contouring control strategy, the cutting tool tries to track a prescribed trajectory and minimize the contour error while rejecting disturbances and handling model inaccuracies. The next level of machine tool control is the machining process control. In this the cutting forces and feed are controlled to achieve maximum material removal rates, while trying to optimize tool wear and avoid tool breakage. A successful implementation of such strategies requires significant advances in the areas of sensing and diagnostics. Several sensors and techniques have been employed to estimate the tool wear and tool life. Tool life is also severely affected by the regenerative self-excited vibrations of the machine tools, also commonly referred as the machine chatter. Much of the research in monitoring and control of machining processes to date has been concerned with regulating a single process via a single process variable. Future research will be concerned with utilizing multiple process variables to control single and multiple processes, which forms the highest level of machine control, that is, the supervisory controls.

Control of machine tools and machining processes is a mature technology. This research topic gained prominence in the 1960s, with the need to have good accuracies along with high production rates, and ushered in the use of numerical controlled and computer numerical controlled machines in industry. Since then several related research areas in servo control, error compensation, adaptive machining process control, sensing, diagnostics, and so on, have been investigated. However, the impact of this research in industry has yet to be fully seen. The major impediment to applying such research in industry has been the high cost associated with designing and deploying new systems. Also, the current research is not easy to implement in industry, as in most cases, it requires considerable knowledge and experience on the part of the operator.

References

1. Taylor, F.W., Art of cutting metals [with discussion]. *Proceedings of the American Society of Mechanical Engineers*, 1906. **28**: 1–248.
2. Ernst, H. and Merchant, M.E., Chip formation, friction, and high quality machined surfaces. In *Surface Treatment of Metals*, NY: American Society of Metals, 1941. **29**: 299pp.
3. Koren, Y. and Lo, C.C., Advanced controllers for feed drives. *CIRP Annals*, 1992. **41**(2): 689–698.
4. Koren, Y. and Ulsoy, A.G., Adaptive control. In *Metals Handbook: Machining*, J.R. Davis (Ed.), 1989, Metals Park, OH: ASM Int., pp. 618–626.
5. Ulsoy, A.G. and Koren, Y., Applications of adaptive control to machine tool process control. *Control Systems Magazine, IEEE*, 1989. **9**(4): 33–37.
6. Dimla, D.E.S., Sensor signals for tool-wear monitoring in metal cutting operations—A review of methods. *International Journal of Machine Tools and Manufacture*, 2000. **40**(8): 1073–1098.
7. Ramadge, P.J. and Wonham, W.M., Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 1987. **25**(1): 206–230.
8. Chen, J.S., Yuan, J.X., Ni, J., and Wu, S.M., Real-time compensation for time-variant volumetric errors on a machining center. *Transactions of the ASME, Journal of Engineering for Industry*, 1993. **115**(4): 472–479.
9. Koren, Y., *Computer Control of Manufacturing Systems*. 1983, McGraw-Hill: New York.
10. Ramesh, R., Mannan, M.A., and Poo, A.N., Tracking and contour error control in CNC servo systems. *International Journal of Machine Tools and Manufacture*, 2005. **45**(3): 301–326.
11. Tomizuka, M., Zero phase error tracking algorithm for digital control. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1987. **109**(1): 65–68.

12. Weck, M. and Ye, G., Sharp corner tracking using the IKF control strategy. *CIRP Annals—Manufacturing Technology*, 1990. **39**(1): 437–441.
13. Koren, Y., Cross-coupled biaxial computer control for manufacturing systems. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1980. **102**(4): 265–272.
14. Seethaler, R.J. and Yellowley, I., Regulation of position error in contouring systems. *International Journal of Machine Tools and Manufacture*, 1996. **36**(6): 713–728.
15. Tsang, T.T.C. and Clarke, D.W., Generalised predictive control with input constraints. *IEEE Proceedings, Part D: Control Theory and Applications*, 1988. **135**(6): 451–460.
16. Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., Reconfigurable manufacturing systems. *CIRP Annals—Manufacturing Technology*, 1999. **48**(2): 527–540.
17. Koren, Y. and Ulsoy, A.G., Reconfigurable manufacturing system having a production capacity method for designing same and method for changing its production capacity, in U.S. Patent # 6,349,237. 2002, The Regents of the University of Michigan.
18. Mehrabi, M.G., Ulsoy, A.G., Koren, Y., and Heytler, P., Trends and perspectives in flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing*, 2002. **13**(2): 135–146.
19. Lee, E.H. and Shaffer, B.W., Theory of plasticity applied to problem of machining. *Transactions of the ASME, Journal of Applied Mechanics*, 1952. **19**(2): 234–239.
20. Landers, R.G. and Ulsoy, A.G., Model-based machining force control. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 2000. **122**(3): 521–527.
21. Kim, T.-Y., Woo, J., Shin, D., and Kim, J., Indirect cutting force measurement in multi-axis simultaneous NC milling processes. *International Journal of Machine Tools and Manufacture*, 1999. **39**(11): 1717–1731.
22. Jeong, Y.-H. and Cho, D.-W., Estimating cutting force from rotating and stationary feed motor currents on a milling machine. *International Journal of Machine Tools and Manufacture*, 2002. **42**(14): 1559–1566.
23. Stein, J.L. and Wang, C.-H., Analysis of power monitoring on AC induction drive systems. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1990. **112**(2): 239–248.
24. Powalka, B., Dhupia, J.S., Ulsoy, A.G., and Katz, R., Identification of machining force model parameters from acceleration measurements. *International Journal of Manufacturing Research*, 2008. **3**(3): 265–284.
25. Dhupia, J., Powalka, B., Katz, R., and Ulsoy, A.G., Dynamics of the arch-type reconfigurable machine tool. *International Journal of Machine Tools and Manufacture*, 2007. **47**(2): 326–334.
26. Budak, E. and Altintas, Y., Analytical prediction of chatter stability in milling. I. General formulation. *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, 1998. **120**(1): 22–30.
27. Landers, R.G. and Ulsoy, A.G., Nonlinear feed effect in machining chatter analysis. *Journal of Manufacturing Science and Engineering*, 2008. **130**(1): 011017–1.
28. Davies, M.A., Pratt, J.R., Dutterer, B., and Burns, T.J., Stability prediction for low radial immersion milling. *Journal of Manufacturing Science and Engineering*, 2002. **124**(2): 217–225.
29. Insperger, T., Hartung, F., Stepan, G., and Turi, J. State Dependent Regenerative Delay in Milling Processes. In *Proceedings of IDETC/CIE 2005 ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference September 24–28*, 2005, Long Beach, California USA: American Society of Mechanical Engineers, New York, NY 10016–5990.
30. Dhupia, J., Powalka, B., Ulsoy, A.G., and Katz, R. *Experimental Identification of the Nonlinear Parameters of an Industrial Translational Guide,* 2006. Chicago, IL: American Society of Mechanical Engineers, New York, NY 10016–5990.
31. Dhupia, J.S., Powalka, B., Galip Ulsoy, A., and Katz, R., Effect of a nonlinear joint on the dynamic performance of a machine tool. *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, 2007. **129**(5): 943–950.
32. Altintas, Y. and Chan, P.K., In-process detection and suppression of chatter in milling. *International Journal of Machine Tools and Manufacture*, 1992. **32**(3): 329–347.
33. Tarng, Y.S. and Li, T.C., Detection and suppression of drilling chatter. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1994. **116**(4): 729–734.
34. Tönshoff, H.K., Friemuth, T., and Becker, J.C., Process monitoring in grinding. *CIRP Annals—Manufacturing Technology*, 2002. **51**(2): 551–571.
35. Karpuschewski, B., Wehmeier, M., and Inasaki, I., Grinding monitoring system based on power and acoustic emission sensors. *CIRP Annals— Manufacturing Technology*, 2000. **49**(1): 235–240.
36. Gradisek, J., Baus, A., Govekar, E., Klocke, F., and Grabec, I., Automatic chatter detection in grinding. *International Journal of Machine Tools and Manufacture*, 2003. **43**(14): 1397–1403.
37. Li, C.J., Ulsoy, A.G., and Endres, W.J. *The Effect of Spindle Speed Variation on Chatter Suppression in Rotating-Tool Machining*. 2006. Taipei, R.O.C., Taiwan: Trans Tech Publications Ltd, Stafa-Zuerich, CH-8712, Switzerland.

38. Pakdemirli, M. and Ulsoy, A.G., Perturbation analysis of spindle speed vibration in machine tool chatter. *IVC/Journal of Vibration and Control*, 1997. **3**(3): 261–278.
39. Wu, S.M. and Ni, J., Precision machining without precise machinery. *CIRP Annals—Manufacturing Technology*, 1989. **38**(1): 533–536.
40. McKeown, P.A., The role of precision engineering in manufacturing of the future. *CIRP Annals—Manufacturing Technology*, 1987. **36**(2): 495–501.
41. Bryan, J., International status of thermal error research (1990). *CIRP Annals—Manufacturing Technology*, 1990. **39**(2): 645–656.
42. Weck, M., McKeown, P., Bonse, R., and Herbst, U., Reduction and compensation of thermal errors in machine tools. *CIRP Annals—Manufacturing Technology*, 1995. **44**(2): 589–598.
43. Gillespie, L.K., *Deburring Capabilities and Limitations*. 1976, Dearborn, MI: Society of Manufacturing Engineers, 429pp.
44. Byrne, G., Dornfeld, D., Inasaki, I., Ketteler, G., Konig, W., and Teti, R., Tool condition monitoring (TCM)—The status of research and industrial application. *CIRP Annals—Manufacturing Technology*, 1995. **44**(2): 541–567.
45. Kurada, S. and Bradley, C., A review of machine vision sensors for tool condition monitoring. *Computers in Industry*, 1997. **34**(1): 55–72.
46. Woldman, N.E. and Gibbons, R.C., *Machinability and Machining of Metals*, 1st edition, 1951. New York: McGraw-Hill.
47. Rehorn, A.G., Jin, J., and Orban, P.E., State-of-the-art methods and results in tool condition monitoring: A review. *International Journal of Advanced Manufacturing Technology*, 2005. **26**(7–8): 693–710.
48. Sampath, A. and Vajpayee, S., Tool health monitoring using acoustic emission. *International Journal of Production Research*, 1987. **25**(5): 703–719.
49. Kakade, S., Vijayaraghavan, L., and Krishnamurthy, R., In-process tool wear and chip-form monitoring in face milling operation using acoustic emission. *Journal of Materials Processing Technology*, 1994. **44**(3–4): 207–214.
50. Dornfeld, D.A., Lee, D.E., Hwang, I., Valente, C.M.O., and Oliveira, J.F.G., Precision manufacturing process monitoring with acoustic emission. *International Journal of Machine Tools and Manufacture*, 2006. **46**(2): 176–188.
51. Pruitt, B.L. and Dornfeld, D.A. *Monitoring End Mill Contact Using Acoustic Emission*, 1996. Boston, MA: ASME.
52. Chow, J.G. and Wright, P.K., On-line estimation of tool/chip interface temperatures for a turning operation. *Transactions of the ASME, Journal of Engineering for Industry*, 1988. **110**(1): 56–64.
53. Choudhury, S.K. and Bartarya, G., Role of temperature and surface finish in predicting tool wear using neural network and design of experiments. *International Journal of Machine Tools and Manufacture*, 2003. **43**(7): 747–753.
54. Boothroyd, G., *Fundamentals of Metal Machining and Machine Tools*. 1975. Washington: Scripta Book Co. xxix, 350.
55. Danai, K. and Ulsoy, A.G. *An Adaptive Observer for On-line Tool Wear Estimation in Turning. II. Results,* 1988. Atlanta, GA: American Automatic Control Council.
56. Koren, Y., Ko, T.-R., Galip Ulsoy, A., and Danai, K., Flank wear estimation under varying cutting conditions. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1991. **113**(2): 300–307.
57. Park, J.-J. and Ulsoy, A.G., On-line tool wear estimation using force measurement and a nonlinear observer. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1992. **114**(4): 666–672.
58. Oraby, S.E. and Hayhurst, D.R., Development of models for tool wear force relationships in metal cutting. *International Journal of Mechanical Sciences*, 1991. **33**(2): 125–138.
59. Yao, Y., Fang, X.D., and Arndt, G., Comprehensive tool wear estimation in finish-machining via multivariate time-series analysis of 3-D cutting forces. *CIRP Annals—Manufacturing Technology*, 1990. **39**(1): 57–60.
60. Bayramoglu, M. and Dungel, U., Systematic investigation on the use of force ratios in tool condition monitoring for turning operations. *Transactions of the Institute of Measurement and Control*, 1998. **20**(2): 92–97.
61. Kistler Corporation. Rotating multi-component dynamometer HS-RCD, Type 9125A. 2008.
62. Sim, W.M., Dewes, R.C., and Aspinwall, D.K., An integrated approach to the high-speed machining of moulds and dies involving both a knowledge-based system and a chatter detection and control system. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2002. **216**(12): 1635–1646.

63. Crouse, M.S., Nowak, R.D., and Baraniuk, R.G., Wavelet-based statistical signal processing using hidden Markov models. *IEEE Transactions on Signal Processing*, 1998. **46**(4): 886–902.
64. Fu, P., Hope, A.D., and King, G.A., On-line tool condition monitoring based on a neurofuzzy intelligent signal feature classification procedure. In *Practical Applications of Soft Computing in Engineering*, Sung-Bae Cho (Eds), 2001. pp. 183–199.
65. Goodwin, G.C.S.K.S., Adaptive filtering prediction and control. *Prentice-Hall Information and System Sciences Series,* 1984. Englewood Cliffs, NJ: Prentice-Hall. xii, 540pp.
66. Ulsoy, A.G., Koren, Y., and Rasmussen, F., Principal developments in the adaptive control of machine tools. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1983. **105**(2): 107–112.
67. Huber, J. and Centner, R., Test results with an adaptively controlled milling machine. ASTME Paper No. MS68–638, 1968.
68. Amitay, G., Malkin, S., and Koren, Y., Adaptive control optimization of grinding. *Transactions of the ASME, Journal of Engineering for Industry*, 1981. **103**(1): 103–108.
69. Hirata, M., Makihara, N., Kawai, K., and Nagasawa, M., *Adaptive Control Apparatus for a Machine Tool*. U.S. Patent, Editor. 1988, Toyoda Koki Kabushiki Kaisha US.
70. Watanabe, T. and Iwai, S., Control system to improve the accuracy of finished surfaces in milling. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1983. **105**(3): 192–199.
71. Wu, C.L., Haboush, R.K., Lymburner, D.R., and Smith, G.H. *Closed-loop Machining Control for Cylindrical Turning*, 1986. Anaheim, CA: ASME (DSC v 4), New York, NY, USA.
72. Teltz, R. and Elbestawi, M.A., Hierarchical, knowledge-based control in turning. *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, 1993. **115**(1): 122–132.
73. Ramamurthi, K. and Hough, C.L., Jr., Intelligent real-time predictive diagnostics for cutting tools and supervisory control of machining operations. *Transactions of the ASME, Journal of Engineering for Industry*, 1993. **115**(3): 268–277.
74. Furness, R.J., Galip Ulsoy, A., and Wu, C.L., Supervisory control of drilling. *Transactions of the ASME, Journal of Engineering for Industry*, 1996. **118**(1): 10–19.
75. Furness, R.J., Ulsoy, A.G., and Wu, C.L., Feed, speed, and torque controllers for drilling. *Transactions of the ASME, Journal of Engineering for Industry*, 1996. **118**(1): 2–9.
76. Furness, R.J., Wu, C.L., and Ulsoy, A.G., Statistical analysis of the effects of feed, speed, and wear of hole quality in drilling. *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, 1996. **118**(3): 367–375.

11

Process Control in Semiconductor Manufacturing

Thomas F. Edgar
University of Texas at Austin

11.1 Introduction

Solid-state devices are manufactured on disks of semiconducting material called wafers. These devices are three-dimensional structures made up of stacked two-dimensional layers. Each layer is manufactured by one or two of the four basic unit operations: deposition, patterning, doping, and heat treatment; see Table 11.1 for examples of each operation. The purpose of deposition is to grow a thin layer of a specific material onto the wafer surface. Patterning is the process of selective removal of the top layer (or layers) of the wafer. In the doping processes the conductivity and resistivity of the wafer surface are altered by adding specific contaminants to the exposed areas of the wafer. Finally, heat treatment raises or lowers the wafer temperature to evaporate solvents or anneal the surface [1].

Figure 11.1 shows representative processing steps in a semiconductor fabricator. Usually 10 or more steps shown in Table 11.1 are required to fabricate an integrated circuit. For example, the following steps are necessary to fabricate a typical metal oxide semiconductor (MOS) gate.

1. *Deposition:* Oxidation of the wafer surface to create a silicon dioxide layer and act as a doping barrier.
2. *Patterning:* Creation of two holes in the oxide layer to define the source and drain of the transistor.
3. *Doping:* Insertion of N-type dopant through the openings in the oxide layer.

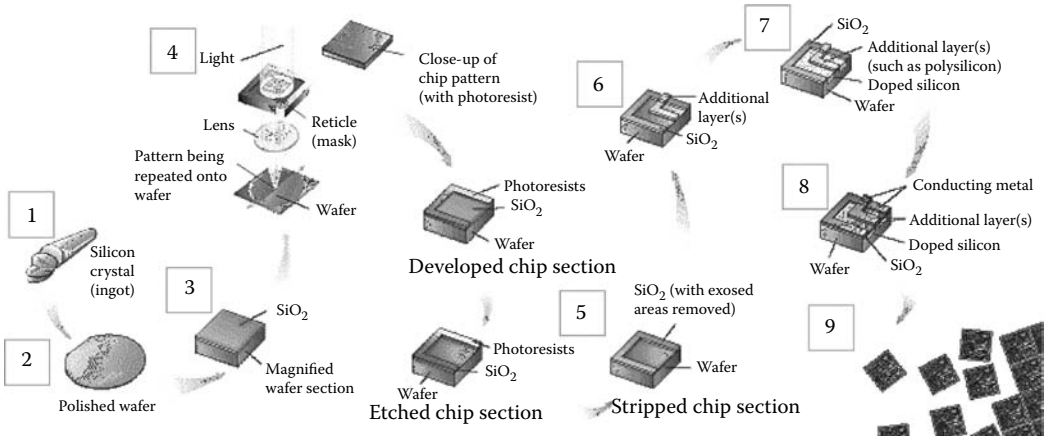

FIGURE 11.1 Flowsheet for key steps in semiconductor manufacturing. (Used with permission from SEMATECH.)

4. *Patterning:* Removal of the oxide between the source and the drain.
5. *Deposition:* Oxidation of exposed silicon to create the gate oxide.
6. *Patterning:* Creation of two holes in the reoxidized source and drain regions.
7. *Deposition:* Deposition of a conductive metal layer.
8. *Patterning:* Removal of portions of the metallization layer.
9. *Heat treatment:* Heating in a nitrogen atmosphere to alloy the metal to the exposed source and drain to improve contact.
10. *Deposition:* Deposition of a passivation layer to protect the transistor.
11. *Patterning:* Removal of portions of a passivation layer to create terminal pads on the periphery of the chip.

11.2 Control Methods in Semiconductor Manufacturing

Process control problems in microelectronics processing can be divided into four categories: plant (fab) management, contamination control, materials handling, and unit operations control. Much attention has been focused on coordinating the schedules of different unit operations, controlling the purity of the required reactants, and monitoring the transfer of wafers between machines. Relatively less effort has been devoted to improving the control of individual unit operations listed in Table 11.1, which is the focus of this chapter.

To be competitive in the global market, semiconductor manufacturing increasingly relies on advanced process modeling and control due to shrinking feature size ($< 0.20\,\mu$m linewidth) and increasing wafer diameter (up to 300 mm). Given these critical dimension (CD) constraints and the trend toward further

TABLE 11.1 The Four Classes of Unit Operations in Semiconductor Manufacturing

Deposition	Oxidation, chemical vapor deposition, molecular beam epitaxy
Patterning	Plasma etch, ion milling, lithography
Doping	Diffusion, ion implantation
Heat treatment	Rapid thermal processing, hot plate heating

miniaturization, extremely tight manufacturing tolerances are required. Achieving such tight specifications represents a major engineering challenge. Comprehensive modeling and control technologies are thus required to achieve satisfactory yields, maximize throughput, and reduce cost.

Traditionally, there have been two distinct approaches to process control in semiconductor manufacturing. Statistical process control (SPC) is a technique in which the process output is monitored (usually by measurements external to the process, or *ex situ*) in order to detect an "out of control" process. SPC attempts to assign a causality relationship to an external disturbance. A process is considered "out of control" if output variance can be attributed to an assignable cause [2]. However, many times the machine is not broken, and the operator compensates for the error by manipulation of a process input variable. SPC does not define the control action necessary to return a process to an "in control" state. This decision is left to the control engineer. SPC has seen widespread acceptance in discrete parts manufacturing where processes generally have high repeatability and natural variability.

The other approach to process control is automatic process control (APC). APC uses measurements of important process variables to implement feedback or feedforward control to keep the product on target. APC essentially accomplishes this by transferring variability in the output variable to an input control variable [3]. Recently, a technique called run-to-run control (run-by-run control) has been used widely to reduce product variability. APC practitioners may view run-to-run control as a supervisory controller which manipulates the setpoints of underlying tool controllers. The ultimate goal of run-to-run control is that of batch control for a lot of wafers. By analyzing the results of previous batches, the run-to-run controller should be able to manipulate the batch recipe in order to reduce output variability for each batch.

There are many different ways that a run-to-run controller can be formulated in order to perform the necessary control tasks. However, almost all controllers will have a similar structure, regardless of the detail. Run-to-run controllers generally are model-based controllers, coupled with an observer of some type. Linear regression and response surface models are the types of models typically employed in model-based control. The majority of models used in run-to-run controllers are steady-state models. These pure gain models assume that the process drift is slow and can be compensated for adequately by the integral control action in the run-to-run controller.

When incorporating feedback from a noisy process, it is useful to employ an observer to estimate the actual state of the process. The design of the observer can be as simple as an arithmetic average of consecutive errors or as sophisticated as a Kalman filter. The observer generally operates in one of two modes. The gradual mode observer is used for a slowly drifting process. In gradual mode, it is assumed that the majority of the output variance is attributable to natural variation in the process. On the other hand, the rapid mode observer is used when the process undergoes a significant shift. When a deterministic change has occurred in the process, the rapid mode observer lends more weight to the output measurements occurring after the large shift.

The most widely implemented design for run-to-run control is based on an exponentially weighted moving average (EWMA) scheme, which can be shown to be identical to internal model control (IMC) [4]. Generally, the process model has a linear regression form

$$y_k = Bu_{k|k-1} + c_{k|k-1} + e_k, \tag{11.1}$$

where y_k is the output at batch k, B is the process gain, $u_{k|k-1}$ is the input at batch k calculated from information up through batch k–1, $c_{k|k-1}$ is the estimate for the intercept, and e_k is unknown process noise entering the system. Typically, the system gain and the initial value of the intercept are obtained *a priori* from designed experiments.

The intercept is updated recursively by an observer of the form

$$c_{k|k-1} = \lambda(y_{k-1} - Bu_{k-1|k-2}) + (1-\lambda)c_{k-1|k-2}, \tag{11.2}$$

where λ is the exponential weighting factor, or tuning parameter, of the observer. The weighting factor λ takes a value between 0 and 1 and is chosen based on the desired properties of the observer. Small values

of λ are appropriate for systems with small deterministic drifts and relatively large natural variance. Conversely, highly correlated output errors are best compensated for thorough use of higher values of the weighting factor. For slowly drifting processes characteristic of the semiconductor industry, λ is typically chosen in the range 0.1–0.3 for the gradual mode of the observer.

One motivation for run-to-run control is a lack of *in-situ* measurements of the product qualities of interest. Typically, in semiconductor manufacturing, the goal is to control qualities such as film thickness or electrical properties, which are difficult to measure in real-time in the process environment. Most semiconductor products must be moved from the processing chamber to a metrology tool before an accurate measurement of the control variable value can be taken. Metrology may not be compatible with the topography of product wafers and can be destructive, in which case one of the product wafers in a lot is sacrificed for measurement purposes. Thus, although pilots (test wafers) are frequently used as a cheaper substitute for product wafers, nontopographic or minimal topographic pilots give results that may not represent actual product wafers. Since *ex-situ* metrology is time consuming, a fab may start a lot before receiving measurements from the previous lot, and this measurement delay can cause problems for run-to-run feedback control. Although few online process sensors are found in modern fabs, in the past few years new sensors from commercial suppliers are beginning to be introduced, including radio frequency (RF) sensors and optical sensors for both the process (optical emission spectroscopy or OES) and the wafer (interferometry, ellipsometry, scatterometry, Fourier transform infrared spectroscopy). Inexpensive mass spectrometers are also being utilized.

Semiconductor processing tools also have real-time controllers, typically PID loops, for controlled variables (CVs) that can be measured in real-time. The variables are typically process inputs, such as chemical flow rates, or reactor states such as temperature or pressure. The manufacturing engineer must specify a recipe which contains the setpoints of these inputs and states that will produce the proper output product. The job of the supervisory, run-to-run controller is to adjust these recipe parameters to reduce variability in the output product.

In other batch processing industries such as specialty chemicals, the major control issue is tracking a setpoint during a batch. In semiconductor manufacturing, disturbance rejection, feedforward control, and product target changes (setpoint changes) between runs are the main control issues. Disturbances include equipment aging, machine maintenance, chamber wall buildup, and unmeasured incoming wafer state changes. Feedforward values that can change include measured incoming wafer state (such as film thickness), as well as measurable machine states (such as age of heat transfer tubes). Product target changes are necessary because the same machine can be repeatedly used for different processes or to make different products.

Figure 11.2 shows a control strategy for a typical set of fab operations at advanced micro devices (AMD). Both feedforward and feedback control methods are utilized around metrology steps; in feedforward control a film thickness measurement determines the starting point for a deposition or etch step. In the latter case, it is necessary to know how much material is to be removed, and then set the etching time based on the known etch rate in nm/s. The feedback control in each step is a run-to-run controller. For more details on fabwide control, see [5].

APC has been applied to a wide range of unit operations in semiconductor manufacturing; including chemical vapor deposition, diffusion furnace, rapid thermal processing (RTP), plasma etching, lithography, and chemical–mechanical planarization (CMP) [3]. Most large manufacturers have in-house control groups who implement modeling and control techniques in production. These activities usually involve customizing existing control theory to treat the specific requirements of semiconductor manufacturing.

A well-designed fab control system is expected to improve throughput, cycle time, yield, maintenance scheduling, flexibility, local and global process understanding, and time to market. APC can also increase the life of the processing equipment in the plant. The International Technology Roadmap for Semiconductor (ITRS) [6] underscores the need for productivity improvement to maintain the industry's historical 30% per-year-per-function reduction in cost. Although cost reductions are attainable through yield improvement, yields are currently high. Thus future improvements must be obtained through

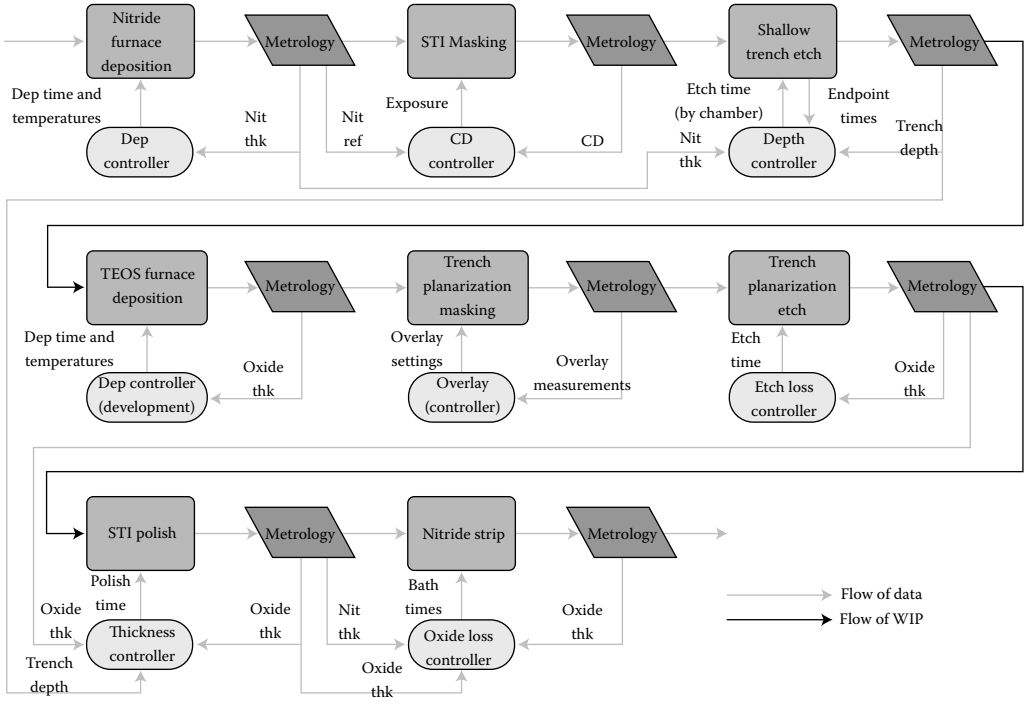

FIGURE 11.2 Fabwide control implementation at AMD. (Adapted from Qin, S.J., et al., *Journal of Process Control*, 16, 179–191, 2007.)

increased capital equipment utilization, which translates to maximizing throughput of product wafers with reduced setup costs.

11.3 Prototypical Example: Lithography Process

To illustrate the principles of modeling and control in semiconductor processing, a lithography process is considered [7]. The goals of lithography are twofold. The first is to create a pattern on the surface of the wafer that resembles as much as possible the design requirements of each layer. The pattern dimensions on the wafer surface are called feature sizes, or more commonly, CD. Although technically the CD is the smallest geometry in the pattern, the size of the geometry used in the metrology tool is usually identified as the CD. The second goal is to ensure the correct alignment of each layer with respect to a previous layer and the global alignment of the device. This alignment is commonly referred to as overlay [8].

In lithography, the pattern transfer is comprised of multiple steps, similar to photography and glass etching. Each pattern to be transferred is first created on a glass and metal mask called a reticle, which is based on the device design. The process for making reticles is very similar to the process used to transfer that same pattern to the wafers in semiconductor manufacturing. A glass or quartz blank is covered with chrome, onto which the pattern is inscribed. This can be done by using a resist-expose-develop cycle, or directly by using an electron beam. Reticles need to be virtually defect-free, and so inspection of the generated patterns is critical. A small defect on the reticle will affect potentially hundreds of wafers before it is detected in the fab.

In optical lithography, typically a light source is focused through the reticle and onto the wafer surface, which is coated with a photosensitive polymer mixture called photoresist or simply resist. Photoresists are designed to respond to specific wavelengths of light, adhere to specific surfaces, and have specific

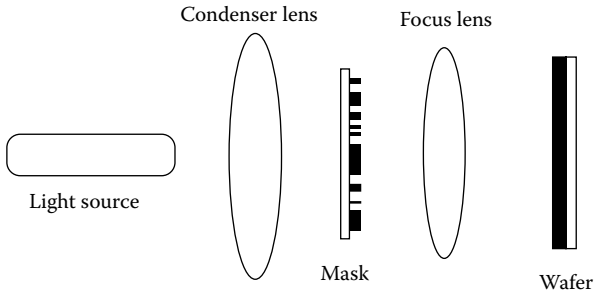

FIGURE 11.3 Basic projection system in optical lithography. (Adapted from Martinez, V. and T.F. Edgar, *IEEE Control Systems Magazine*, 26(6), 46–55, 2006. © IEEE, 2007.)

thermal flow characteristics. Resists are typically composed of four elements: the polymer itself, a solvent, sensitizers, and additives. These optically sensitive resists need to be protected from premature exposure due to ambient light, which is why photolithography bays in semiconductor fabs have yellow lighting. When areas of the wafer are exposed to the energy source, the chemistry of the polymer in the resist changes, making it more or less soluble in a development solution.

A generic projection system for photolithography is shown in Figure 11.3. This system consists of the light source, condenser lens, mask, objective lens, and resist-coated wafer. The light source coupled with the condenser lens is called the illumination system. The lens element is a combination of several refractive (glass) and reflective (mirror) glass pieces. The purpose of the illumination system is to deliver light from the source into the mask, with sufficient intensity, directionality and uniformity. The light passes through the transparent portions of the mask (reticle) where it is diffracted by the glass. It then goes through the objective lens, which projects the image onto the wafer surface. The resist in the exposed areas of the wafer surface becomes more or less soluble in a particular solvent. When the wafer is developed, the pattern on the mask has been transferred to the wafer surface and is now ready for the next operation.

Photolithography is the most complex of all unit operations in semiconductor manufacturing because it combines both chemical and mechanical processing steps. The tools used for processing are called *steppers* or *scanners,* depending on the way the wafer is moved through the exposure field of the reticle. Steppers expose a whole field in a section of the wafer, then move to the next section and expose the whole field again. This process is repeated until the entire wafer surface has been exposed. On the other hand, scanners do not expose the field all at once. They use a narrow field to project the light through the mask and onto the wafer, very similar to the way a photocopying machine operates. Both the wafer and the reticle move in scanners, but the reticle moves faster than the wafer, so the image projected onto the surface of the wafer is several times smaller than the image on the reticle. After each field has been exposed, the wafer steps to the next field, which is why scanners are also called *step-and-scan* tools. Most modern tools are step-and-scan models that use deep ultraviolet (DUV) lasers to project their images.

The formation of each layer of a semiconductor device by transferring the image from the reticle to the wafer surface requires 10 basic steps. There are many variations depending on the layer to be formed and the technology being utilized. Figure 11.4 illustrates the etching and lithography steps to produce a pattern. Each step has associated process control technology, which is discussed below.

11.3.1 Surface Preparation and Resist Coating

Prior to the application of the photoresist, the wafer surface needs to be completely clean and free of contaminants or moisture. Depending on what the previous step was, the wafers might be put through a wet chemical cleaning step. Because the wafer surface needs to be free of water molecules to ensure adhesion of the resist, a dehydration baking step is sometimes employed. Environmental conditions in the fab are

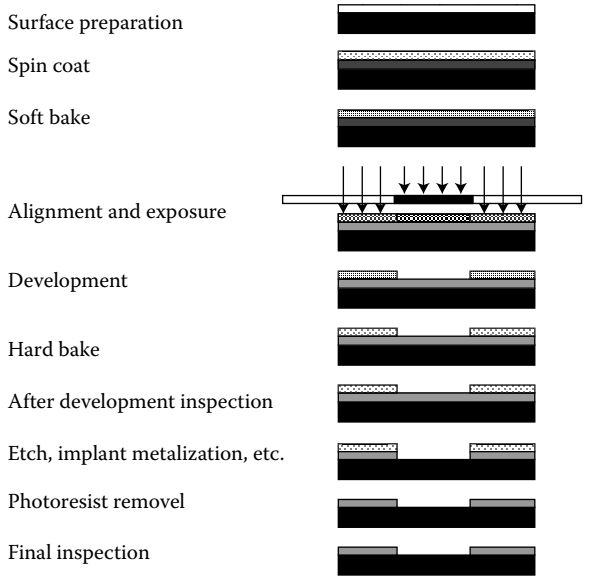

FIGURE 11.4 Ten-step lithography process to transfer images from the reticle to the wafer. (Adapted from Martinez, V. and T.F. Edgar, *IEEE Control Systems Magazine*, 26(6), 46–55, 2006. © IEEE, 2006.)

designed to keep moisture low and prevent water collection on the wafer surface. In addition to dehydration baking, the wafer might go through a chemical priming step to ensure good adhesion of the resist.

When applying resist to the wafer surface, the goal is to create a thin, uniform, defect-free film. A typical resist layer varies from 0.5 to 1.5 μm in thickness and has a uniformity of ± 0.01 μm. In order to achieve such tight uniformity, modern tracks (a separate equipment piece, usually found attached to a stepper) use a moving arm dispenser process. Each wafer is rotated at a low speed, while a moving-arm resist dispenser moves in a slow motion from the center of the wafer toward the edge. After the entire wafer is coated, the rotational speed is increased to thin the resist into a uniform film. This technique creates a more uniform film on the wafer and minimizes the waste of material. The final thickness of the resist film is affected by viscosity, surface tension, and drying characteristics of the resist and the spin speed.

11.3.2 Soft Bake

The heating operation known as soft bake removes some of the solvents present in the resist. There are two reasons for removing the solvents from the film on the wafer surface. First, the solvent molecules can interfere with the chemical reactions triggered by the exposure energy; second, excess solvent will prevent complete adhesion of the film to the wafer surface, which could cause peeling of the resist. It is important to control the temperature and duration of the soft bake to prevent over or underbaking. Overbaking can result in polymerization of the resist molecules, and underbaking can cause incomplete image formation during the exposure step.

There are several methods that can be used for baking: convection oven, vacuum oven, microwaves, hot plates, etc. Hot plates are commonly used due to their good temperature control capabilities as well as short bake time. However, using hot plates reduces throughput, because it is a single-wafer operation.

Conventional thermal systems utilize separate bakeplates and chill plates to accomplish the baking steps. These units are comprised of large thermal mass systems that are held constant at the setpoint temperature. The substrate is placed on the bake or chill plate. The substrate typically rests about 5 mils (thousandths of an inch) from the surface of the plate on small pins, as opposed to direct contact, to prevent contamination. The plates can be single or multi-zone systems.

The performance of conventional bake systems can be analyzed by simulating the energy balance equations [9]. As expected, the temperature at the edge of the wafer is lower than the center. Thus to have a uniform wafer temperature, a nonuniform bakeplate temperature is needed. A conventional bakeplate cannot provide wafer temperature uniformity. Even if the initial resist thickness before the bake step is uniform, the resulting bake step would cause the resist thickness to be nonuniform due to the nonuniform wafer temperature.

With the trends toward larger wafer size and the linewidth going below 100 nm, one of the challenges is to control the resist thickness and uniformity to a tight tolerance in order to minimize the thin-film interference effect on the CD. Ho et al. [10] proposed a new approach to improve the resist thickness control and uniformity through the softbake process. Using an array of thickness sensors, a multizone bakeplate, and an advanced control strategy, the temperature distribution of the bakeplate is manipulated in real-time to reduce resist thickness nonuniformity. The bake temperature is also constrained to prevent the decomposition of a photoactive compound in the resist. They experimentally obtained a repeatable improvement in resist thickness uniformity from wafer-to-wafer and across individual wafers. Thickness nonuniformity of less than 10 Å was obtained. On average, there was 10× improvement in the thickness uniformity as compared to the conventional softbake process.

11.3.3 Alignment and Exposure

Precise alignment of the image patterns and projection of the exact image dimensions are the most critical requirements for producing functional chips, also known as die, in semiconductor manufacturing. Each layer has to be properly aligned with a previous layer and also with an absolute reference so that the entire device is formed correctly. There are several kinds of alignment systems, both optical and nonoptical. For the sake of brevity, we will describe only optical systems used in step-and-scan tools, the more common ones in the industry. Exposure systems also vary widely, depending on the type of energy used.

In step-and-scan systems, both the wafer and the reticle move, which makes alignment of each component very important. The reticle is part of a moving system called the stage, and it moves in two dimensions only. The wafer, on the other hand, can move in the same plane as the stage as well as tilt off the plane in any direction. The first layer on the wafer is aligned by positioning the axis of the reticle at a 90° angle to the wafer notch. Subsequent layers are aligned to a previous layer with the use of alignment marks, also called targets; see Figure 11.5. These special patterns are located on the edge of each die and are used by the onboard alignment system to position the wafer during each step. Automatic alignment is done by focusing a low-energy laser beam through the alignment marks on the reticle and reflecting them off the marks on the wafer surface. The signal is analyzed by an onboard computer which then calculates corrections that are sent to the wafer positioning system to align it with the reticle.

11.3.3.1 Lithography Overlay (Alignment)

The first mask exposed on a wafer is aligned so that the x-axis is parallel to the wafer flat or notch. Subsequent masks are aligned to a previous layer with the use of alignment marks, which are either

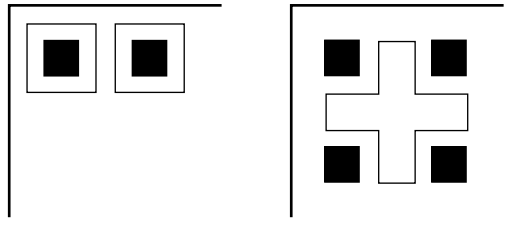

FIGURE 11.5 Examples of alignment marks. (Adapted from Martinez, V. and T.F. Edgar, *IEEE Control Systems Magazine*, 26(6), 46–55, 2006. © IEEE, 2006.)

located in the spaces between die or on the edge of the wafer, as shown in Figure 11.5. The stepper then uses a low-energy laser to align the reticle with the marks on the wafer surface. It can use either marks on the reticle or reference points inside the projection system, in so-called *through the lens alignment*. The goals of the alignment system are twofold: to align the current layer with the reference *align-to layer* and to maintain an overall alignment of the entire device structure as it is built up.

After the wafer is exposed and leaves the stepper, it goes to a standalone metrology tool. The metrology tool uses the same marks on the current layer, but it often uses a different reference layer than the stepper, called the *measure-to layer*. There are many reasons why the stepper and the metrology tool use different reference layers, including device design specifications, overall alignment objectives, and visibility of the marks themselves. It is difficult to control a system like this, where the outputs (metrology) do not have a clear and direct relationship to the inputs (stepper corrections). Overlay alignment errors fall into several categories (linear, rotational, wafer field, reticle field, etc.) depending on how they are generated and what they look like. Figure 11.6 shows schematics of the various wafer and reticle field errors.

Overlay metrology recipes include a selection of sites across the wafer surface, where the wafer and reticle errors are measured. The raw data are fit to an overlay model using mathematical regression techniques. This overlay model varies according to the total manufacturer, but a typical one for intrafield errors is shown in Equations 11.3 and 11.4:

$$\delta e_x = T_X + S_S * x - R_S * y + S_A * x - R_A * y, \tag{11.3}$$

$$\delta e_y = T_Y + S_S * y + R_S * x - S_A * y - R_A * x. \tag{11.4}$$

δe_x = measured x-axis misregistration, δe_Y = measured y-axis misregistration, T_X = translation in x, T_Y = translation in y, S_S = symmetric scaling, R_S = symmetric rotation, S_A = asymmetric scaling, R_A = asymmetric rotation, x = horizontal component of the distance from the wafer center to the data point, and y = vertical component of the distance from the wafer center to the data point.

Using the raw data from the wafer sites and from typically 9 to 15 reticle fields, each of which has several alignment marks, the metrology tool software calculates wafer averages for the overlay parameters. It is not practical to measure all 25 wafers in a lot; hence, a few are randomly selected for metrology and their calculated averages are then averaged together to obtain the overlay metrology values for the lot. A significant effort is made in selecting sites that will provide an accurate representation of all the reticle fields on each wafer that is measured. Sites are selected so that they represent all areas of the wafer, such as the periphery and the center of the wafer.

When the overlay maximum error is within specifications for a given layer, the wafer is allowed to continue to the next step in processing. The subsequent layer's alignment marks will then have some degree of displacement. When the wafer arrives at the lithography equipment for exposure of the next layer, it will be aligned with the reticle based on already misaligned marks. This type of error can accumulate layer after layer until the final product is completely out of alignment with the original layer. The use of different kinds of tools (steppers, scanners) with different light sources (I-line, DUV) contributes to the accumulation of overlay errors.

Model predictive control (MPC) was applied to the control of lithography overlay in high-volume fabrication facilities at AMD [11]. Automated overlay control was able to reduce the maximum site-level error, averaged over all controlled masking operations, by 43% over manual methods. The average maximum error at the beginning of the project was 90% of the allowable overlay error. As the controller was deployed to more masking layers and refined in configuration, it was able to reduce the overall error over a 2-year period to stable operation at roughly 51% of the average specification limit.

The first phase of deployment of run-to-run control was a standard model-based EWMA controller, lasting 23 months. MPC was deployed to the fabrication facility in favor of the EWMA controller and has been used successfully since 2002. The MPC method, along with the other improvements detailed within this work, was able to realize a 9% improvement to the average overlay error over the EWMA controller.

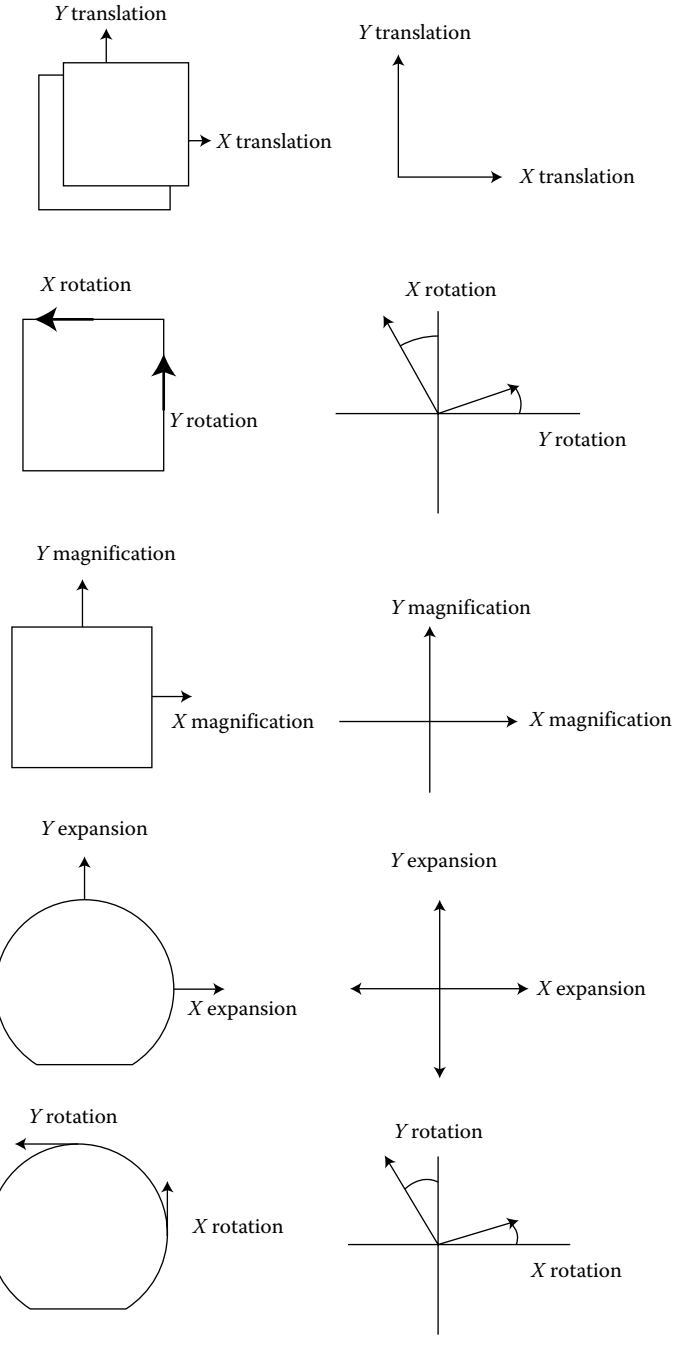

FIGURE 11.6 Wafer or reticle field errors. (Adapted from Martinez, V. and T.F. Edgar, *IEEE Control Systems Magazine*, 26(6), 46–55, 2006. © IEEE, 2006.)

In addition to the improved control, the deployment of the MPC method yielded other manufacturing benefits. Test wafers, used widely within semiconductor manufacturing, are nonproduct wafers or small production wafers lots that are run through a process to assess its performance. As test wafers add to the cost of running the process, both in material costs and tool time taken away from normal production,

reduction of test wafer utilization is desirable. The MPC control method facilitated a virtual elimination of test wafers for the purpose of overlay control at AMD. It also automated recipe management, significantly reducing the amount of engineering time required to maintain the process as well as eliminating human error. These benefits, along with the improved control, increased tool availability and production capacity of the lithography module.

11.3.4 Development and Hard Bake

When the wafer has finished the alignment and exposure steps, the image of the reticle is now latent on the wafer surface. The exposed areas of the resist have undergone a polymerization reaction which will make them insoluble (negative resists) or soluble (positive resists) in a development solution. The processes to develop positive and negative resists are different, as well as the chemicals used in the process.

Negative resists produce a large dissolving rate differential between the two regions of the wafer surface. The development process consists of a chemical developer bath, followed by a rinse step. The developer bath dissolves the unpolymerized sections of the wafer, while leaving the polymerized resist intact. The rinse serves two purposes, to rapidly dilute the developer on the wafer surface to stop the development process, and to remove partially polymerized regions in the transition sections between exposed and nonexposed areas. Positive resists, on the other hand, have a relatively small dissolving rate differential between the exposed and nonexposed areas, which makes the development process a more delicate operation. Too much time in the development solution can cause overthinning of the resist or peeling.

The hard bake is the second heat treatment operation in photolithography and its purpose is the same as the soft bake, that is, to evaporate solvents present and to harden the resist. The goal of this operation is to achieve good adhesion between the resist and the wafer surface. For large wafers, temperature uniformity control is a challenge, and often the equipment has multiple heating zones to achieve a uniform temperature. An approach for multizone temperature control similar to that described for the softbake step could be used, for example [9].

11.3.5 After Development Inspection

The purpose of the after development inspection (ADI) is to evaluate the quality of the pattern present on the wafer surface. There are two variables measured in ADI: the CD and the overlay (discussed below). If either of those variables is outside the spec limit, the lot will be rejected. A rejected lot will be stripped of the resist and reprocessed in the litho bay. Such a lot is commonly called *rework*. Reworked lots go through the exact same stages as new (first pass) lots and can receive the same recipe settings as the first pass, if there is no feedback control, either manual or automatic. Rework rates increase with the complexity of the design and the decreasing size of the CD being printed. While a reworked lot does not have an inferior quality to that of a first pass lot, reworking lots diminishes throughput and wastes process time and materials. It is essential to keep rework at a minimum to maintain an efficient and cost-effective fab running.

There are several methods that can be used to measure CD and overlay. The most common in modern fabs is a scanning electron microscope (SEM), which uses an electron beam as an illumination source. The impinging electrons cause electrons on the surface of the wafer to be ejected. These secondary electrons are collected and translated by the computer into an image. Advanced pattern recognition software then analyzes the images and calculates the CD and overlay for the wafer. These values are then sent to the data archive to be used for adjusting recipe parameters when necessary.

11.3.6 Final Steps

If the wafers pass the ADI inspection, they are sent to the next bay in the fab, which can be an etch step, metallization, doping, etc. In the case of etch, the wafers are exposed to a reactive fluid (which can be a liquid, gas, or plasma) that eats away material from the exposed wafer surface. This etching carves the

pattern of the reticle onto the substrate of the wafer. After the etch process, the wafers are stripped of photoresist and the wafers are inspected again. This inspection is called after clean inspection, or ACI. Because etching does not affect the position of the pattern on the wafer surface, overlay is no longer measured, only the CD is measured. A CD failure, due to over- or underetch, would cause the wafers to be scrapped, as this failure is nonrecoverable and would normally render the devices on the wafer useless. This is why CD control in the etch process is very important and has been the object of many investigations over the last decade [12,13].

11.4 Stepper Matching (Factory Control)

As run-to-run control has become more widely used throughout the semiconductor industry, it has become apparent that some of its unique manufacturing characteristics are driving the need for enhanced algorithm development. One such trait is the high mix of products made in a single factory (such as an ASIC fab or foundry). Not only might there be a great many different products, but the mix of products is therefore constantly changing. In addition, the high cost of process equipment drives manufacturers to maximize the use of their tools, having as little down or idle time as possible, leaving little room for dedication of tools to specific product process streams. Therefore, one lot of a specific product may take a very different processing path through the fab than the next lot of that same product.

Variations in product quality often are functions of the product being produced as well as the manufacturing tools being used, which is termed manufacturing context. Different products behave differently during processing due to factors such as differences in materials used, configuration or layout of devices and interconnects, feature size, and overall chip size. To further complicate matters, seemingly identical tools may process identical wafers differently based on such conditions as the number of lots processed since the last maintenance event, small differences in tool construction, or minor variations in ambient conditions.

One technique used by photolithography engineers in factory control is called tool or stepper matching. This approach is particularly useful in overlay, where the placement of each layer depends on the placement of previous layers. Each stepper has an internal mechanism to align the wafer and the reticle. These mechanisms cause systematic errors in the pattern generation, usually defined in terms of registration, shown in Equation 11.5.

$$R = P_1 - P_0, \tag{11.5}$$

where R = registration, P_1 = position of the wafer geometry, and P_0 = corresponding point in the tool reference grid.

The tool reference grid varies depending on the tool manufacturer, but most use an initial zeroth layer printed on the wafer. When aligning the reticle and the wafer before exposing each field, the tool will use this reference grid along with marks in the reticle to move the wafer into position. Despite being continuously calibrated, different steppers have slightly different registration characteristics.

Registration errors (as well as overlay errors) can be divided into two classes, *intrafield* and *interfield* errors. Intrafield errors are caused by the projection system of the tool and the reticle itself, while interfield errors are caused by the stage positioning system. Errors that are not consistent across the wafer surface cannot be accounted for in the global overlay models used by the tool software, and thus they cannot be controlled by changing parameters in the recipe. These errors, however, will be somewhat consistent for each tool and are repeated from layer to layer. If an entire device were to be built using just one tool, the registration errors would cancel out, thus producing a low overlay error. However, absolute dedication of one equipment producing a single product is not economically feasible due to the excessive capital equipment cost, particularly in high-volume, high-mix manufacturing environments where many final products (e.g., 50) are made during one month of operation. In this case, a given stepper will be used to produce a range of products.

Most fabs have some level of tool dedication; however, they usually reserve their most sophisticated (and expensive) steppers for the more critical layers, while using their older steppers to process the noncritical layers. In the so-called *mix-and-match* environments like these, registration errors account for a significant portion of the accumulated overlay errors. Stepper matching is used to minimize the variation in registration errors across the existing toolset by adjusting internal tool parameters to reproduce a specific registration pattern. A set of *golden wafers*, manufactured specially to have the lowest possible registration error, is used as a reference by processing them in every stepper in the bay. Based on the metrology data, each stepper is adjusted to match the pattern in the golden wafers. By repeating this process over and over, all the tools can be matched to one another.

An alternate stepper matching technique is to define a reference stepper. This stepper is then used to print a reference pattern, then those wafers are processed in the rest of the steppers, and adjustments are made to match them to the reference tool. While this technique has the advantage of matching stepper overlay errors as opposed to registration errors, it is also dependent on the state of the reference tool at one moment in time.

The goal of the stepper matching procedure is to calibrate all the tools in the bay so that they perform as close to one another as possible. Unfortunately, steppers tend to drift over time; hence these adjustments have to be scheduled regularly, sometimes as often as every week. When a tool is found to be off-spec, it has to be taken offline to perform the necessary adjustments. In a high-volume manufacturing environment, idle time hurts productivity and lowers throughput. The cost benefit of manually matching tools can be outweighed by the cost of keeping tools idle for adjustments.

11.5 Rapid Thermal Processing

RTP has become an indispensable step in almost all modern integrated circuit manufacturing. Several steps in the fabrication of integrated circuits including the silicide anneal, the implant anneal, and gate oxide formation require that the silicon wafer be raised to a high temperature (e.g., 1000^{o}C) for a short period of time. The least restrictive of these processes is perhaps the silicide anneal, where a metal film deposited on the wafer surface is annealed with silicon to form a metal silicide (e.g., titanium silicide). Currently, RTP is widely used in these applications because of its comparatively low-energy requirements, its capacity for single-wafer processing (which limits the consequences of misprocessing), its capability to purge undesirable gases (e.g., oxygen) before the wafer is heated, and its short cycle time which decreases the amount of work in progress (WIP). Because the process window for silicidation reactions is comparatively large, the limitations in RTP temperature uniformity have not hindered its application.

The semiconductor wafer in RTP systems is heated by infrared heat sources, usually tungsten-halogen or arc lamps (see Figure 11.7 for a multizone lamp configuration). In these systems, radiation is usually the dominant heat transfer mechanism. Temperature control, uniformity issues, and the introduction of slip dislocations have been the main barriers that have kept RTP from becoming a widely used production tool. The main factors affecting wafer temperature uniformity and process repeatability in RTP systems are the infrared heat source, the chamber design, and the temperature control system, which includes a noninvasive real-time temperature sensing system. It is crucial that temperature at the wafer surface be maintained uniform, since small variations in temperature can lead to large variations in reaction rates. An additional problem that arises for some RTP designs is that for silicon deposition the absorptivity of the surface and the surface roughness can change as the film thickness increases. This in turn causes a time-dependent change in the rate of heat transfer for constant lamp settings. In addition, temperature measurements (typically by pyrometry) will be in error. This suggests that a constant gain proportional controller using lamp voltage as the manipulated variable (MV) and temperature as the controlled variable (CV) could cause nonconstant growth rates due to the nonlinear, time-varying process gain [14]. Breedijk et al. [15] reported an enhanced nonlinear model predictive control scheme using successive model linearization and model predictive control for an RTP reactor, which gives improved control over linear

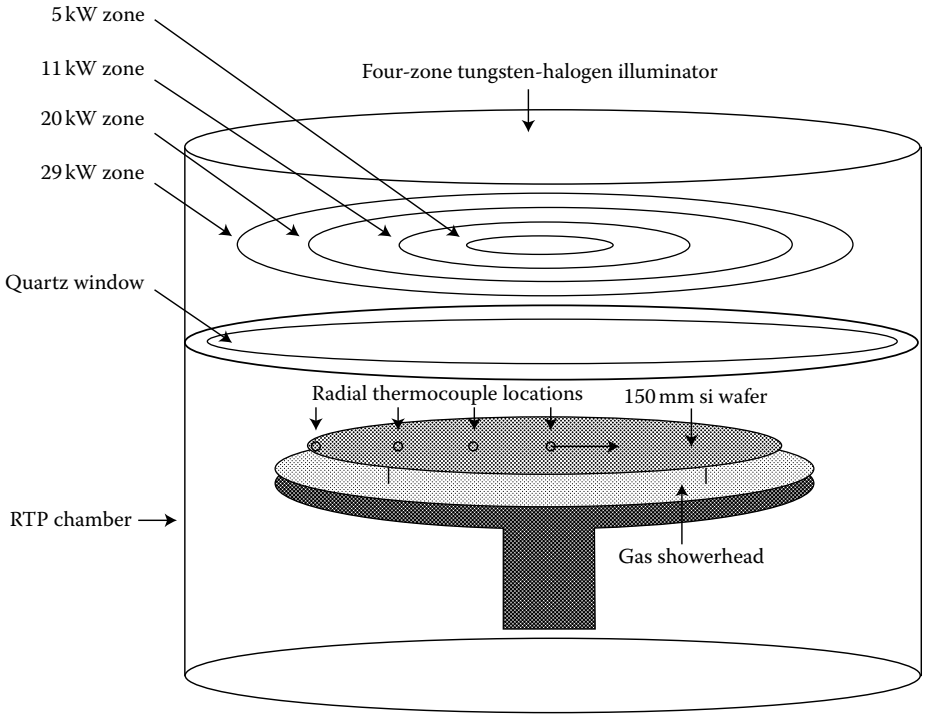

FIGURE 11.7 Schematic of the Texas Instruments four-zone RTP system.

model-based schemes developed previously. They obtained a generalized distributed parameter model for the energy equation in the semiconductor wafer. In developing the multivariable control system for the four-zone reactor (Figure 11.6), Breedijk et al. [15] recognized the ill-conditioned nature of the 4×4 control system as seen in the condition number of the gain matrix. Instead of controlling the average and standard deviation of the four temperatures using the MPC algorithm for model predictive control, transformation of the output equation resulted in a 4×2 reduced system. The normalized gain matrix norms for the 4×2 and 4×4 systems suggested that the transformed system was less nonlinear than the original system, and the gain matrix condition numbers indicated that the transformed system was much better conditioned (by a factor of over 100), and therefore easier to control than the original system.

11.6 Plasma Etching

Control of the plasma etching process is one of the more difficult challenges that faces the microelectronics industry. The process itself is a complex function of nonlinear variables that is still poorly understood at a fundamental level. Key process parameters such as etch rate, uniformity, and device geometry are difficult to obtain directly from the process, and are generally not known during the operation of the etch. In addition, etch performance is dependent on time-varying factors and the chamber architecture; hence repeatable results are hard to obtain.

Conventional operation of plasma etching is relatively rudimentary. Variables such as the individual gas flow rates, RF or microwave power, chamber pressure, applied bias voltage, and other such process parameters are controlled by simple proportional-integral-derivative (PID) algorithms. Most, if not all of these variables, are coupled to one another, making independent manipulation of any of the process parameters difficult. The relationships between these MVs and control variables such as etch rate, final

TABLE 11.2 Important Measured, Manipulated, and Performance Variables for Metal Gate Etch of TiN with Cl_2/N_2

Measured/Controlled Variables	Manipulated Variables	Performance Variables
[Cl], [Cl_2]	RF power	TiN etch rate (center)
DC bias power	Chamber pressure	TiN etch uniformity
(ion energy)	Etchant composition (N_2/Cl_2)	(center edge)
Collision rate	Total etchant flow rate	TiN horizontal etch rate
Electron density		Selectivity
Electron energy distribution		

Source: Adapted from Edgar, T.F., *SEMICON Korea*, STS-2, 65–77, Seoul, Korea, 2008.

device geometry, across-wafer and wafer-to-wafer uniformity, and other process parameters is difficult to model. A lack of techniques to measure these variables *in situ* and in real-time makes APC very difficult to implement.

Table 11.2 shows the process variables for a plasma etching process, where measured variables are those available from process instruments, the MVs are input variables that can be changed independently with respect to time (via a control system), and the performance variables are to be optimized during operation of the process. Table 11.2 is based on a proposed system to monitor and control chlorine-based plasmas such as Cl_2/N_2. The chlorine plasma is chosen here as an example since it has wide application in metal gate etch.

Run-to-run feedback control systems can be designed for an etch process after proper models have been developed. For example, the RF power system is a key subsystem in a plasma etching reactor. It controls plasma density as well as substrate potential. OES is commonly used in plasma etch process monitoring as an *in-situ* measurement of the plasma state. In the OES spectrum there are several characteristic lines or peaks that correspond to certain plasma species. Generally, these lines and peaks must be empirically weighted to approximate the related species concentration.

Center and edge control of linewidth and isolated and nested lines have been the primary control metrics of interest for plasma etching. For gate, contact, and trench processes, there is more than one CV so that wafer uniformity and line density sensitivity can be maintained. Controlling etch profile and linewidth requires adjusting more than one MV for both center and edge profile control. To address profile and uniformity control the system requires a multiinput multioutput (MIMO) approach that recognizes the interactions among CVs and MVs. The etch recipe will usually contain multiple steps, and each step can have different MVs. Typical MVs, for example, include step duration (time), gas flow, power, pressure, temperature, or combination of gas ratios, center and edge gas flows, and center and edge temperatures.

Nonlinear programming can be used to treat nonlinear MIMO relationships and constraints on the MVs and CVs in order to maximize performance of a multistep etch process by changing the recipe after each run. A quadratic objective function utilizes weighting factors to prioritize each CV term in the objective function. Recipe optimization can be combined with run-to-run feedback control to provide closed-loop control that maximizes a specific performance objective. Feedback filters based on EWMA are used to update offset terms after each run.

A commercial polysilicon gate etch process was used by Lee et al. [17] to demonstrate the effect of multivariable recipe optimization. As shown in Figure 11.8, the CVs are etch bias (EB), sidewall angle bias (SWAB), and difference between center and edge CD (CDΔ). Incoming disturbance variables (DVs), namely the incoming CD and sidewall angle (SWA), also affect CVs; so this interaction needs to be included in the model. Control performance measured by the mean square error (MSE) of added CVs (SWA and CDΔ) is better than the current process which controls CD only. Using MIMO optimization allows much better control of SWA and CDΔ, compared to the "No Control" case.

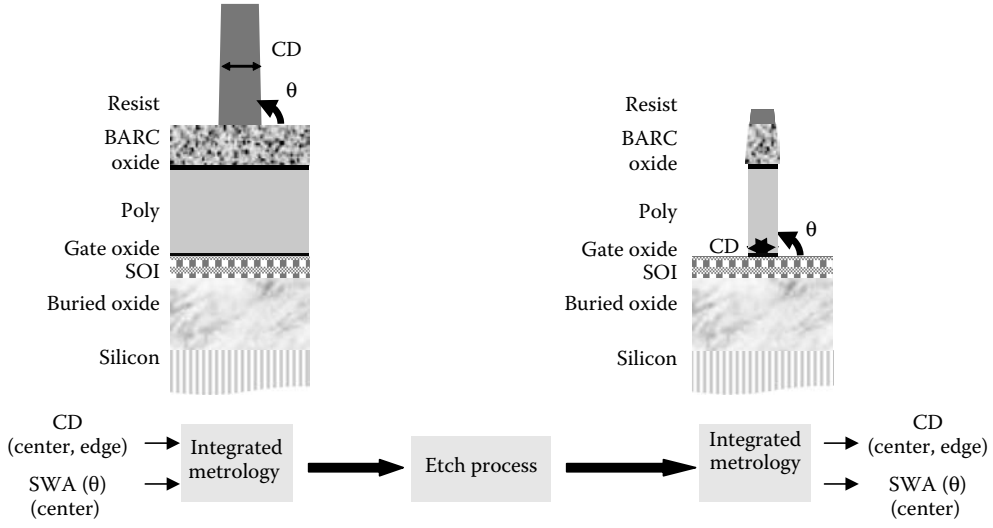

FIGURE 11.8 Inputs and outputs for polysilicon gate etch process in semiconductor manufacturing. The measured inputs (CD) and SWA in the incoming wafer can be used in feedforward control, while the measured outputs (CD, CDΔ, and SWA) are used in feedback control. BARC is bottom antireflective coating, SOI is silicon on insulator.

11.7 Chemical–Mechanical Planarization

CMP is used to provide planarity of interlayer dielectric silicon dioxide and in lithography-limited submicron trench isolation. The CMP process involves a silicon wafer, attached to a carrier by vacuum, being pressed face down into a polishing pad. The polishing environment is flooded with a colloidal slurry which physically enhances abrasion and helps prevent redeposition of the oxide or metals. The polish table is rotated while the wafers also rotate about their axis and orbit about the polish table. Due to the nature of the polishing environment, it is not possible to obtain real-time measurements of the surface planarity. *Ex-situ* measurements of surface thickness and uniformity are required to characterize the process. Today's tools are available in multihead configurations, which allow polishing of as many as five wafers simultaneously.

As CMP is typically used to prepare a wafer for lithography, the goal of the planarization process is to produce wafers that are as flat as possible [3]. Typically, a target thickness is specified with a surface uniformity tolerance. The CMP process is generally subject to three different sources of variation which will affect the outputs. The first is the natural variation of the process. Every manufacturing process is subject to natural variance, and the random noise in the CMP process is amplified by the nonuniform nature of the slurry polishing environment. A second source of variation is incoming variations from previous processes. Other processes that experience output variation, such as deposition, are coupled to the planarization process. This leads to a need for a feedforward controller. Finally, the CMP process is subject to degradation of consumables. The polishing pad wears as multiple wafers are polished causing a slow drift in the polishing rate. Other consumables in the CMP process include wafer carriers and conditioning pads. With the introduction of multihead polishing tools, a new source of variation has been introduced to the process. Small variations in polishing rates due to head dependences can cause significant output variation among the product.

Due to the lack of *in-situ* measurements of surface thickness, real-time control can only be enacted on process inputs. These inputs include polish time, polish table rotational speed, polish head downforce pressure, slurry delivery rate, wafer rotational speed, and carrier backforce pressure. These inputs are typically kept at the setpoints specified in the polish recipe by a series of single-input, single-output PID

control loops. Since the PID controllers respond only to inputs, a supervisory, or run-to-run, control scheme must be used to update the setpoints to compensate for process variation [18].

11.8 Conclusions

Control of microelectronics manufacturing in the future must address the need for faster yield ramp, increasing cost pressures that compel productivity improvements, and environmental, safety and health concerns. The real drivers in the immediate future are the shrinking device dimensions and chip size. Control changes will occur in both real-time equipment control and for run-to-run control systems.

The development of 300 mm platforms since 1999 has spawned equipment with new software systems and capabilities. These systems allow smart data collection, storage, and processing on the equipment and transfer of data and information in a more efficient manner. These new software platforms integrated with wafer processing tools provide the biggest opportunity for a control paradigm shift seen in the industry since the introduction of SPC.

Besides replacing real-time control algorithms such as PID with model-predictive controllers, adding other sensor outputs in the control algorithm will continue in the future. In thermal processes, this entails use of improved temperature sensors. Increased use of cluster tools means that the only opportunity for instrumentation of multistack films or multichamber processes will be to measure either *in situ* or online. Once the metrology is added, opportunities for feedforward control, not just feedback control, become more feasible. Many times the sensors that are available do not measure the primary performance variable. Thus, soft sensors, that is, the fusion of sensor data to estimate another nonmeasured variable, have a role in the future. While some soft sensors for thermal variables are available, widespread use of soft or virtual sensors has not occurred. The reason is that the estimation error is generally much larger than desirable.

As run-to-run feedback control becomes more common, closed-loop identification will become important. While this is commonly used in other industries, it has not been relevant so far to semiconductor processing. The main challenge of closed-loop identification for batch control is to ensure every batch is within the specification. Another identification issue is the emergence of technology for analysis of data with more than one timescale, that is, real-time data gathered approximately once per second with sensors vs. data measured post-run with *ex-situ* metrology.

There will be many opportunities to apply advanced modeling and control techniques in semiconductor manufacturing in the future. Improvements in supervisory (run-to-run) control are likely to have a major impact, especially in reducing the number of test wafers that must be used. The development of fundamental mathematical models for single-wafer reactors has reached a fairly high level of sophistication and should provide a means of evaluating various advanced control techniques for this type of equipment. Mathematical models should also be helpful in analyzing how design parameters affect the quality of control for single-wafer reactors, because precise control for larger wafers will be mandatory. However, control strategies (multivariable, model-predictive, and possible adaptive) need to be developed for such reactors. The use of feedback and feedforward control in semiconductor tools has been hindered by the lack of real-time measurements; additional research on accurate and relatively inexpensive noninvasive measurement techniques should be a high priority in order to implement real-time process control techniques [3].

References

1. Quirk, M. and J. Sedra, *Semiconductor Manufacturing Technology*, Englewood Cliffs, NJ: Prentice-Hall, 2001.
2. Seborg, D.E., T.F. Edgar, and D.A. Mellichamp, *Process Dynamics and Control*, 3rd ed., New York: Wiley & Sons, 2010.

3. Edgar, T.F., S.W. Butler, W.J. Campbell, C. Pfeiffer, C.A. Bode, S.B. Hwang, K.S. Balakrishnan, and J. Hahn, Automatic control in microelectronics manufacturing: Practices, challenges, and possibilities, *Automatica*, 36(11), 1567–1603, 2000.
4. Butler, S.W., J. Stefani, M. Sullivan, Maung, S.G. Barna, and S. Henck, An intelligent model based control system employing *in situ* ellipsometry. *Journal of Vacuum Science and Technology, A*, 12(4), 1984–1991, 1994.
5. Qin, S.J., G. Cherry, R. Good, J. Wang, and C.A. Harrison, Semiconductor manufacturing process control and monitoring: A fabwide framework, *Journal of Process Control*, 16, 179–191, 2007.
6. International Technology Roadmap for Semiconductors, www.itrs.net, 2008.
7. Martinez, V. and T.F. Edgar, Control of lithography in semiconductor manufacturing, *IEEE Control Systems Magazine*, 26(6), 46–55, 2006.
8. Levinson, H.J., *Lithography Process Control*, Bellingham, WA: SPIE Optical Engineering Press, 1999.
9. Ho, W.K., A. Tay, L.L. Lee, and C.D. Schaper, On control of resist film uniformity in the microlithography process, *Control Engineering Practice*, 12, 881–892, 2004.
10. Lee, L.L., C.D. Schaper, and W.K. Ho, Real-time predictive control of photoresist film thickness control, *IEEE Transactions on Semiconductor Manufacturing*, 15(1), 51–59, 2002.
11. Bode, C.A., B.S. Ko, and T.F. Edgar, Run-to-run control and performance monitoring of overlay in semiconductor manufacturing, *Control Engineering Practice*, 12, 893–900, 2004.
12. Krogh, O., M. Freeland, R. Mori, and T. Chowdhury, Gate etch process control, *Proceedings of SPIE*, 5038, 1065–1070, 2003.
13. Toprac, A.J. AMD's advanced process control of poly-gate critical dimension, *Proceedings of SPIE*, 3882, 62–65, 1999.
14. Chatterjee, S., H. Huang, C.J. Spanos, and M. Gatto, Modeling and control of RTCVD of polysilicon, *Proceedings of RTP Conference*, 386–391, 1993.
15. Breedijk, T., T.F. Edgar, and I. Trachtenberg, Model-based control of rapid thermal processes, *Proceedings of the American Control Conference*, 887–892, 1994.
16. Edgar, T.F., Process monitoring and control of plasma etching, *SEMICON Korea*, STS-2, 65–77, 2008, Seoul, Korea.
17. Lee, H., A. Ranjan, D. Prager, K.A. Bandy, E. Meyette, R. Sundararajan, A. Viswanathan, A. Yamashita, and M. Funk, Advanced profile control and the impact of sidewall angle at gate etch for critical nodes, *Metrology, Inspection, and Process Control for SPIE Advanced Lithography*, 69220T-13, 6922, 2008.
18. El Chemali, C., J. Moyne, K. Khan, R. Nadeau, P. Smith, J. Colt, J. Chapple-Sokol, and T. Parikh, Multizone uniformity control of a chemical mechanical polishing process utilizing a pre- and post-measurement strategy, *Journal of Vacuum Science and Technology, Part A*, 18(4), 1287–1296, 2000.

12

Control of Polymerization Processes

Babatunde Ogunnaike
University of Delaware

Grégory François
Swiss Federal Institute of Technology in Lausanne

Masoud Soroush
Drexel University

Dominique Bonvin
Swiss Federal Institute of Technology in Lausanne

With an annual worldwide production well in excess of 100 million metric tons, synthetic polymers constitute a significant part of the modern chemical process industry. Polymer reactors—operated in continuous, batch, or semibatch mode—are therefore important processing units, but there are unique problems associated with controlling them effectively. The most significant characteristics of polymer reactors that make them one of the most challenging units to model, control, and optimize, are discussed in this chapter; we also provide a survey of the strategies that have been proposed, and those that have been successfully employed in industrial practice.

12.1 Introduction and Overview

The primary objective of polymerization processes is to produce polymers that will perform consistently and acceptably in specific end-use applications, for example, in light switches, automobile bumpers, fiber-optic cables, and so on. How well these polymer products perform is determined by such product attributes as tensile strength, toughness, and UV resistance, which ordinarily cannot be measured during the manufacturing process. But these product attributes themselves arise from the molecular and/or macroscopic architecture of the polymer—characteristics that are determined *during* polymer synthesis. Thus, meeting customer specifications on polymer product attributes (in addition to maintaining safe

process operation, and meeting production volume targets and environmental regulations) requires effective control of the manufacturing process in a manner that is unique to this class of processes [1–3].

As discussed further below, polymerization processes (and there is a wide variety of them) are complex, and exhibit significant nonlinear characteristics that are, in many cases, poorly understood; they also lack online measurements of those product properties that are important to the product performance in end-use. Furthermore, what constitutes an effective control strategy is highly dependent on the specific characteristics of the polymer process in question. For example, large volume polymers are produced more economically in *continuous* processes, where the primary objective is to start up the process as quickly as possible and maintain it at an economically desirable steady-state operating condition. On the other hand, low-volume specialty polymers are produced in batch and semibatch processes, where the primary objective is to obtain acceptable product quality at the end of each batch cycle. These two distinct operating modes present different sets of control problems unique to each mode of operation.

The principal difficulties in achieving good control of polymerization reactors are related to inadequate online measurement, a lack of understanding of the dynamics of the process, the nonlinear behavior of these reactors, and the lack of well-established techniques for controlling nonlinear processes. While temperatures, pressures, flow rates, and reactant compositions are routinely measured online, important product quality variables such as molecular weight distributions (MWDs) and copolymer composition are usually measured offline, and typically with very long time delays. End-use polymer properties, which are related to the molecular weight and composition distributions in the polymerization reactor according to relations that are not entirely well understood, can only be measured after lengthy post-manufacturing processing. Finally, each continuous industrial polymer reactor is typically used to manufacture a wide variety of grades of the same basic product, thereby requiring frequent startups, transitions, and shutdowns. Similarly, the same batch or semibatch reactor is often used for the production of many different polymers from different sets of reactants; and while the equipment (reactor) remains the same, how the process is operated and controlled often depends on the product currently being manufactured.

This chapter provides an overview of the key issues associated with controlling polymer reactors, along with a discussion of techniques for addressing them. The rest of the chapter is organized as follows: in Section 12.2, we provide a fundamental scientific context for the control of polymer reactors by presenting a brief introduction to the mechanisms and processes by which polymers are produced; next, we discuss the control of continuous processes in Section 12.3, and control of batch (and semibatch) processes in Section 12.4. A summary and some conclusions are presented in Section 12.5.

12.2 Background: Polymerization Mechanisms and Processes

12.2.1 Polymerization Reaction Mechanisms

Polymers, very large molecules consisting of a huge number of monomer units linked in long chains, are produced via many different reaction mechanisms. These mechanisms influence the fundamental architecture of the final molecule, and hence the final product characteristics. Two of the most common mechanisms, free radical and ionic polymerization, are summarized here.

12.2.1.1 Free-Radical Polymerization

As illustrated below, this mechanism consists of four steps: (1) *Initiation*, where an initiator molecule decomposes to create two primary free radicals, and each radical reacts with a monomer unit to produce a "live" polymer chain of length 1; (2) *Propagation*, where the live polymer molecule reacts rapidly with the monomer to produce a growing polymer chain; (3) *Termination*, where a live polymer molecule reacts with another live polymer molecule to form a dead polymer molecule, either by combination (into a single polymer chain), or by disproportionation (into two dead polymer molecules); (4) *Chain transfer*, where

the free radical at the end of a growing polymer chain is transferred to a chain transfer agent, a monomer molecule, a solvent molecule, or even another polymer molecule.

- *Initiation*

$$I \rightarrow 2I^{\bullet}$$
$$I^{\bullet} + M \rightarrow R_1$$

- *Propagation*

$$R_k + M \rightarrow R_{k+1}; \quad k = 1, 2, \ldots$$

- *Termination by disproportionation*

$$R_k + R_{k'} \rightarrow P_k + P_{k'}; \quad k, k' = 1, 2, \ldots$$

- *Termination by combination*

$$R_k + R_{k'} \rightarrow P_{k+k'}; \quad k, k' = 1, 2, \ldots$$

- *Chain transfer to transfer agent*

$$R_k + T \rightarrow P_k + T^{\bullet}; \quad k = 1, 2, \ldots$$

- *Chain transfer to monomer*

$$R_k + M \rightarrow P_k + R_1; \quad k = 1, 2, \ldots$$

- *Chain transfer to solvent*

$$R_k + S \rightarrow P_k + S^{\bullet}; \quad k = 1, 2, \ldots$$

- *Chain transfer to polymer*

$$R_k + P_m \rightarrow P_k + R_m; \quad k, m = 1, 2, \ldots$$

Here I and $I^{\bullet}$ are the initiator molecule and the initiator radical, respectively; M is a monomer molecule; R_k and P_k represent growing (live) polymer and dead polymer molecules of length k, respectively; T and S are, respectively, transfer agent and solvent molecules with corresponding radicals represented as $T^{\bullet}$ and $S^{\bullet}$. A defining characteristic of polymer molecules is that they grow to varying lengths. Thus, the chain length k indicated above is not fixed; it is a random quantity, determined by a wide variety of factors. Polymers are, therefore, macromolecules with non-uniform molecular weights, which is why they are primarily characterized by their MWDs. Note also from the mechanisms shown above that, with free-radical polymerization, the resulting product molecular weight distribution (and hence average molecular weights) can be controlled by manipulating (directly or indirectly) the rates of initiation, propagation, chain transfer, and termination.

12.2.1.2 Ionic Polymerization

With ionic polymerization, the intermediate species are positively or negatively charged ions—cations or anions, respectively—rather than free-radicals. Furthermore, the reactions differ from the free-radical polymerization reactions in that termination occurs only when the ions react with water. And since termination cannot occur simply by the interaction of two ionized molecules, the average molecular weight is more easily controlled in ionic polymerization than with free-radical polymerization. Furthermore, because the initiation reaction has a low activation energy, ionic polymerization can be performed at low temperatures. However, ionic reactions are more difficult to be carried out on an industrial scale, which is why, whenever possible, free-radical polymerization is preferred.

12.2.2 Polymerization Processes

Whether operated in continuous or batch mode, a wide variety of processes are available for manufacturing polymers, and each process has its own distinct characteristics. The processes most widely employed in industrial practice are summarized below.

1. *Bulk polymerization:* This process, also known as mass polymerization, consists in the polymerization of a (typically liquid) monomer, in the absence of any medium other than a catalyst, initiator, or accelerator. An important feature of bulk polymerization is whether or not the polymer is soluble in the monomer phase.
2. *Solution polymerization:* In this case, the polymerization reactions take place in the medium of a solvent in which the monomer and catalyst are dissolved. The heat generated by the reactions is absorbed by the solvent, making temperature control easier to achieve. In some cases, the solvent must ultimately be removed from the polymer (e.g., via distillation), which may be rather costly.
3. *Suspension polymerization:* This process involves polymerization in the medium of a liquid (usually water) in which the monomer is *not* soluble. Vigorous mechanical stirring and a stabilizing agent are used to generate suspensions of monomer droplets within which the polymerization takes place. As with solution polymerization, temperature control is easier than with bulk polymerization, because part of the heat generated by the reaction is absorbed by the water.
4. *Emulsion polymerization:* This popular process refers to the free-radical polymerization of monomer *emulsions*—water-insoluble monomer molecules dispersed into droplets that are stabilized by a mono-layer of surfactant molecules at the water-monomer interface. Polymerization is initiated via a water-soluble initiator, while propagation proceeds by monomer molecules diffusing from droplets to growing polymer particles, where the presence of a surfactant prevents aggregation of the particles.

   The product of an emulsion polymerization is called a "latex," and in many cases, virtually all the monomer is consumed; furthermore, no solvents are involved. Finally, high molecular weight latexes can be obtained at favorable reaction rates, something not possible with either bulk or solution polymerization processes. These features confer significant economic and environmental advantages and are mostly responsible for the popularity of emulsion polymerization in industry.

Bulk and solution polymerization are therefore classified as *homogeneous* processes while suspension and emulsion polymerization processes are *heterogeneous*.

Another common classification in polymerization is based on the number of different monomers involved in the reactions. In *homopolymerization*, the polymer is made from a single monomer, while in *copolymerization* or in *terpolymerization*, two or three different monomers are involved in the formation of the polymer product.

Observe therefore that the term "polymerization process" covers a broad spectrum of possible operating configurations, reaction mechanisms, and fundamental processes. For example, Figure 12.1 depicts a semibatch, emulsion copolymerization process in which monomer B and surfactant are preloaded into the reactor, with gradual addition of monomer A and initiator. The resulting polymer product is removed at the end of the batch.

Despite such diversity, these manufacturing processes share some common characteristics around which the challenges to effective control may be framed.

1. Polymerization processes exhibit complex steady state and nonlinear dynamic characteristics, including multiple steady states, open-loop instability, and high parametric sensitivity.
2. Polymerization processes involve multiple strongly interacting variables.
3. A typical industrial continuous polymer reactor is used to manufacture a variety of grades of the same basic product, thus necessitating frequent startups, online transitions, and shutdowns. Similarly, the same batch or semibatch reactor is often used for the production of many different

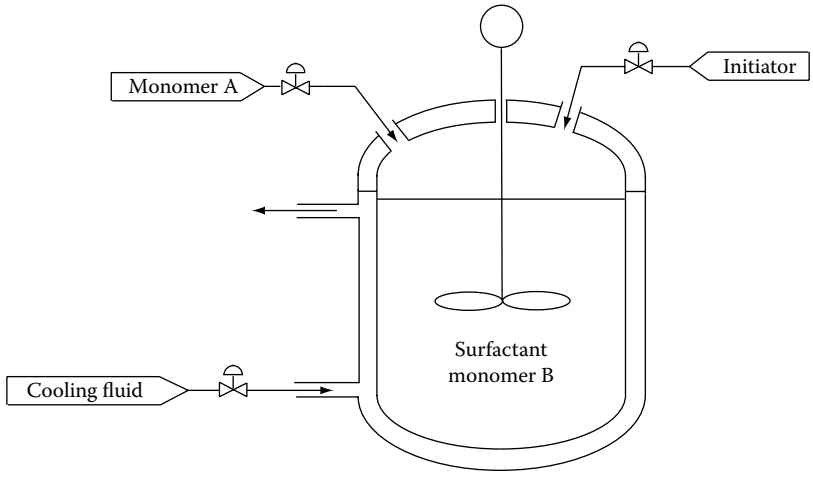

FIGURE 12.1 Typical semibatch emulsion copolymerization reactor.

polymers from different sets of reactants. While the equipment (reactor) remains the same, how the process is operated and controlled often depends on the product currently being manufactured.

4. The important polymer product quality determinants (e.g., average molecular weights, MWD, melt index, Mooney viscosity, etc.) can be measured only very infrequently. Product end-use properties (e.g., tensile strength, UV resistance, etc.), which are dependent on product quality determinants, can only be determined post manufacturing.

Thus, by themselves, classical linear, single-loop controllers with static structures based on frequently available measurements, are often incapable of delivering effective control of product characteristics without some sort of augmentation.

We are now in a position to discuss the strategies for controlling polymerization processes, beginning with continuous processes.

12.3 Continuous Processes

12.3.1 Process Characteristics and Control Problems

Continuous processes, used mainly for high-volume production of commodity polymers, often exhibit strong nonlinearities in the form of multiple steady states, parametric sensitivity, limit cycles, and so on [4–6]. Especially with free-radical polymerization, a major source of the nonlinearity is the autocatalytic nature of the polymerization reactions—the so-called "gel effect"—which frequently causes uncontrollable reactions, resulting in excessive temperature rise, rapid conversion, and equipment plugging.

With continuous polymerization processes, there are four identifiable modes of operation:

1. Startup
2. Steady-state operation
3. Grade transition (i.e., transition from one steady-state to another)
4. Shutdown

During startups and shutdowns, the primary objective is safety; product quality control is mostly of concern during steady-state operation and grade transitions where, in this latter case, the objective is to transition from one steady-state operation to the next one as efficiently as possible. The control objectives, and hence, appropriate control strategies, are different in each case.

The typical modern strategy involves a generic, two-level hierarchical structure, with "base regulatory control" at the first level for controlling such process manipulated variables as monomer flow rates and temperatures. The set-points for these process variables are determined at the higher, "advanced control level," in order to obtain desired product characteristics. The specific details of how the controllers in each level are designed and implemented can vary depending on the specific problems at hand, and the level of sophistication desired by the practitioner. Some general principles that are broadly applicable are discussed next.

12.3.2 Base Regulatory Control

12.3.2.1 Temperature Control

As polymerization reactions are usually highly exothermic, temperature control is of universal importance, regardless of operating mode. In startup and shutdown mode, temperature control is primarily for assuring safety; in the other two operating modes, temperature control is used indirectly to influence polymer properties, since temperature has a strong effect on polymer properties. Temperature control takes on added importance when the reactor must be operated at an unstable steady state. In highly exothermic polymerization, in addition to controlling the actual value of the reactor temperature, the *rate of change of reactor temperature* must also be carefully monitored, especially when a significant amount of unreacted monomer is present in the reactor. This is because a sharp increase in temperature in combination with a significant amount of unreacted monomer in the reactor poses considerable safety challenges.

With the most common equipment designs, heating and cooling is achieved by fluids flowing *outside* the reactor in a surrounding jacket, or else through a heating/cooling tube *inside* the reactor. In industrial applications, effective reactor temperature control is typically achieved with a cascade control scheme consisting of two proportional-integral-derivative (PID) controllers, where the outer temperature controller sets the set-point for the inner cooling/heating fluid flow controller. Standard techniques for cascade control systems design are found in process control textbooks (e.g., [7–9], etc.) are therefore customarily employed for designing and implementing basic temperature control in polymer reactors.

12.3.2.2 Flow Control

For continuous polymerization reactors, the total sum of reactant (feed) flow rates (equivalent to the ratio of the reactor volume to the reactor mean residence time), has a profound effect on the steady-state and dynamic behavior of the reactor. As long as the reactor is operated at the desired steady state, this total feed rate should be maintained as constant as possible. The monomer and solvent (inert) flow rates are usually dominant; the flow rates of chain-transfer agents, cross-linking agents, and initiators, are typically small relative to these dominant flow rates. As such, because changes in these "small" flow rates have little or no effect on the reactor residence time, the "small" flow rates are used as manipulated inputs to influence and fine-tune polymer properties.

Multiple single-loop PID controllers are typically used to maintain the dominant flow rates constant, and to set the desired values for the "small" flow rates when used as manipulated variables. Once again, these controllers may be designed successfully using standard techniques discussed in process control textbooks. Thus, base regulatory control for polymer reactors involves no more than the application of standard single-loop and cascade control, specifically adapted to polymerization processes.

12.3.3 Advanced Control Strategies I: Steady-State Operation

While necessary (and adequate) for meeting safety and production volume requirements, base regulatory control of temperature and flows are insufficient (and inadequate) for achieving product quality objectives. For the purpose of ensuring good product quality during steady-state operation and during

transition from one steady state to another, control of continuous polymerization reactors requires the judicious manipulation of the set-points to the regulatory controllers, a task that is usually carried out with advanced control strategies. This task is particularly challenging primarily because measurements of the variables to be controlled are *not* available online, or frequently enough via other analysis methods. The predominant strategy, to be discussed shortly, is to use available online measurements to infer desired polymer properties. But first, we note three major directions in which research efforts have been directed toward making available reliable online measurements, from which polymer properties can be inferred.

- Development of new online sensors [2,10].
- Development of state estimation techniques for estimating nonmeasurable polymer properties from available measurements ([2], and the references therein). A list of measurements from which certain polymer properties can be observed and/or detected is given in [11].
- Understanding and exploiting the qualitative and/or quantitative relations between easily-available online measurements such as density, viscosity and refractive index, and certain polymer properties such as an average molecular weight [12–14]. See, for example, [15] for an approach to predicting melt index and density in a fluidized bed ethylene copolymerization reactor from available online temperature and gas composition measurements.

Control structures for implementing advanced control of polymerization reactors can be divided into three major groups:

1. Multirate cascade control structure (Figure 12.2)
2. Multirate decentralized control structure (Figure 12.3)
3. Multirate control structure with multirate state estimation (Figure 12.4)

All these control structures reflect a key defining characteristic of advanced control of polymer reactors: the availability of measurements at different sampling rates and with different time delays. While "fast" measurements such as temperatures, pressures, and flow rates are available at high sampling rates and with almost no time delays, "slow" measurements, which are usually directly related to product quality, are measured at low sampling rates and with considerable time delays (time delays as long as 24 h between the time a sample is taken and the time the sample analysis becomes available are not unusual). In the subsequent discussion, the vector of fast output measurements is denoted by y, and the vector of slow output measurements by Y.

12.3.3.1 Multirate Cascade Control Structure

As depicted in Figure 12.2, this control structure consists of two loops (or levels, or layers), for the case where measurements are available at two different sampling frequencies—one loop for each sampling frequency. The inner loop responsible for controlling the fast outputs is executed at the higher sampling frequency of these fast measurements, while the outer loop (primary controller) is executed at the lower sampling frequency of the slow measurements. The primary controller periodically (and infrequently) adjusts the set-point values of the secondary (fast) controlled outputs. In general, the number of different sampling frequencies at which the measurements are available determines the total number of distinct feedback loops.

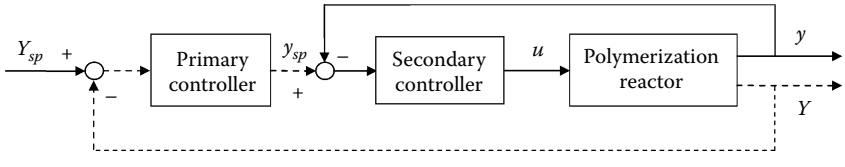

FIGURE 12.2 Multirate cascade control structure.

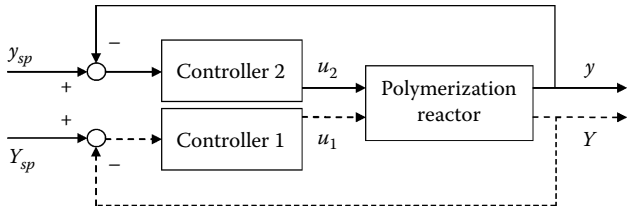

FIGURE 12.3 Multirate decentralized control structure.

In the polymerization industry, the primary controller is often an operator or a process engineer who, based on his/her experience and on data available from laboratory sample analysis, adjusts the set-point values of the secondary controlled outputs in order to achieve the desired polymer product quality. In this structure, the set-point values of the secondary controlled outputs are updated whenever measurements of the primary controlled outputs are available. The primary and secondary controllers can be of any type—from classical PID controllers to model predictive controllers.

As a simple illustrative example, consider a polymerization reactor in which reactor temperature, measured online every second without delay, is y, and the rate of thermal energy added to or removed from the reactor is the manipulated input, u. Let the polymer number-average molecular weight that is measured offline in a laboratory once a day with a time delay of one day, be the secondary controlled output, Y. Under the configuration in Figure 12.2, the reactor is under "continuous" temperature control, and the temperature set-point is adjusted by the primary controller (possibly an operator or a process engineer) once a day, depending on the difference between the measured number-average molecular weight and its desired value.

This control structure is quite common in the polymerization industry. For such industrial systems, at the "lower level," the secondary loop is configured and implemented in the distributed control system (DCS) for controlling pressure, temperature, level, and flow, whereas, at the "higher level," the primary control loop is configured in the supervisory computer for advanced control of polymer properties [16].

12.3.3.2 Multirate Completely Decentralized Control Structure

This control structure, depicted in Figure 12.3, consists of two "independent" feedback loops, for the case in which measurements are available at two different sampling frequencies. Note that, when the measurements are available at $n > 2$ different sampling frequencies, there will be a commensurate number of distinct feedback loops, n, one for each measurement frequency.

As with the structure in Figure 12.2, the secondary (fast) loop, which uses Controller 2 to control the fast outputs y, operates at the higher sampling frequency of the fast output measurements. On the other hand, the primary Controller 1 regulates the slow outputs (product quality variables) Y and operates at the lower sampling frequency of the slow output measurements. However, unlike the structure in Figure 12.2, Controller 1 directly adjusts (infrequently) its own set of manipulated inputs, u_1, that are paired with the primary (slow) controlled outputs. As with the primary (master) controller in the cascade structure of Figure 12.2, Controller 1 can be an operator or a process engineer. Both controllers (1 and 2) can also be standard automatic controllers—from classical PID controllers to model predictive controllers. The decentralized nature of this multirate control structure endows the control system with a measure of robustness in the presence of extremely low-frequency measurements with very long delays. Compared to the cascade control structure of Figure 12.2, implementing this control structure requires a larger number of manipulated inputs.

As a simple illustrative example, consider a reactor for which the reactor temperature is measured online every second without delay, while the polymer number-average molecular weight is measured offline in a laboratory once a day with a time delay of one day. In this case, the latter is the primary (slow) controlled output Y, while temperature is the secondary (fast) controlled output y. For this process, the

manipulated inputs u_1 and u_2 are, respectively, the flow rate of a chain transfer agent (or a thermal initiator), and the rate of thermal energy added to or removed from the reactor: The latter is used to control temperature, while the former is used to control the number-average molecular weight. As in the control structure of Figure 12.2, the reactor is under "continuous" temperature control; however, unlike that earlier structure, the flow rate of the chain transfer agent or the thermal initiator stream is adjusted by the primary controller (possibly an operator or a process engineer) only once a day, based on the observed deviation between the measured number-average molecular weight and its desired value, almost independent of the temperature control loop [16].

12.3.3.3 Multirate Control Structure with Multirate State Estimation

This control structure, depicted in Figure 12.4, is more general than the preceding structures, since it is suitable for cases where measurements are available over a broad range of sampling frequencies. Because it includes a multirate state estimator, the entire control scheme must be model-based, unlike the previously discussed schemes where the controllers can take any form. The heart of the scheme is the multirate state estimator which uses the infrequent measurements of the primary outputs Y, the frequent measurements of the secondary outputs y, and information about the vector of manipulated inputs u, to calculate the frequent, delay-free estimates $\hat{x}$ of all the polymerization reactor state variables. Such estimates are then used in a suitable control scheme to control the primary and secondary outputs simultaneously.

The estimator can be Kalman filter-based [17], or Luenberger observer-based [18]. However, because of the nonlinear dynamics associated with polymer reactors, these filters and observers must be nonlinear; in addition, sufficiently accurate dynamic models of the polymer reactors are required in order to obtain sufficiently accurate estimates of the unmeasured states. In recognition of the fact that accurate process models are not easy to develop (especially for industrial processes), it is important that the estimator be robust to plant-model mismatch and unmeasured disturbances. One of the more straightforward means of ensuring such robustness is to estimate a set of model parameters along with the state variables. This makes the estimator adaptive, at the cost of solving a larger estimation problem. Because of the coupling among variables and the heavy reliance on the estimation of the states of a complex process, this control scheme is more likely to be less robust to the late arrival of measurements than the other two schemes.

To illustrate, let us consider the same simple polymerization reactor example used earlier, in which the reactor temperature is measured online at a high sampling frequency and without delay; furthermore, the polymer number-average molecular weight is measured offline in a laboratory once a day, with a time delay of one day. The manipulated inputs u_1 and u_2 are, respectively, the flow rate of a chain transfer agent (or a thermal initiator), and the rate of thermal energy added to or removed from the reactor. In this case, a multirate state estimator uses all the available information to obtain a high-frequency, delay-free estimate of the number-average molecular weight, which the controller (a multivariable or a fully decentralized controller) subsequently uses to determine appropriate control action. Note that the control action is determined and implemented at the high sampling rate of the secondary output y. In this

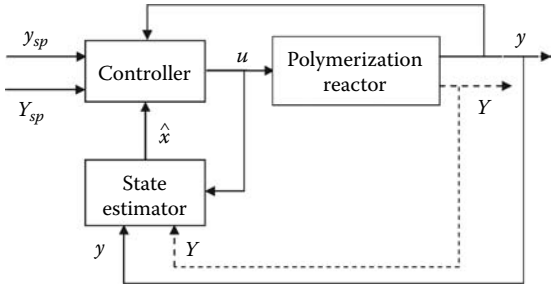

FIGURE 12.4 Multirate control structure with multirate state estimation.

control structure, "continuous" control of number-average molecular weight is achieved using the high-frequency, delay-free *estimates* of number-average molecular weight produced by the state estimator. The control system performance therefore depends heavily on the performance of this multirate state estimator.

Note that one can also use a multirate estimator with the control structures shown in Figures 12.2 and 12.3, calculate delay-free frequent estimates of the primary outputs, Y, and use the delay-free frequent estimates in the feedback loops instead of the delayed infrequent measurements of the primary outputs. In such control schemes, both controllers (the primary and secondary controllers in Figure 12.2; Controllers 1 and 2 in Figure 12.3) operate at the higher sampling frequency of the fast output measurements. Nevertheless, "continuous" control for both feedback loops is achieved at the cost of requiring a robust estimator.

12.3.4 Advanced Control Strategies II: Grade Transition

A single continuous polymerization process is often used in industry to produce several different grades of the same polymer product. For such processes, each "campaign" involves operating at a particular steady-state condition until the desired amount of the corresponding product grade has been manufactured; thereafter, the next "campaign" starts by "transitioning" to the new operating conditions required to make the next product type in the production cycle. Operating such processes effectively, clearly requires that transitions between grades be carried out smoothly and as quickly as possible. Slow transitions lead to the production of considerable amounts of off-specification polymer, and the resultant waste of energy and reactants.

The intrinsic nature of the control problem associated with grade transitions (driving process outputs from an initial state to a different final desired state in minimum time) makes transition control of continuous polymer reactors an ideal dynamic optimization problem. Such problems may be solved in several different ways.

One approach is to compute, offline, the optimal input (feed flow rates and temperature) profiles via numerical optimization using a nominal process model, and then implement such input profiles online, either in an open-loop fashion (as computed with no feedback or mid-course correction), or in a closed-loop fashion, where the optimal input profiles are enforced using temperature and feed flow rate feedback controllers. The primary disadvantage with this dynamic optimization approach is that the grade transition ceases to be optimal in the presence of plant-model mismatch and unmodeled process disturbances. Nevertheless, it is still possible to obtain good performance even under such nonideal conditions. An example application is contained in [19]. Optimal open-loop policies/trajectories were determined for reactor temperature, bleed stream flow, catalyst feed rate, and bed level, via offline dynamic optimization studies. A differential geometric controller was used to regulate instantaneous melt index and density, and to provide servo control during grade changeovers. Hydrogen and butene feed rates were manipulated to force the "measured" product properties onto the desired trajectories.

An alternative approach involves formulating the problem as a model predictive control (MPC) problem designed to minimize a desired cost function (typically the amount of off-spec material, the transition time, or both). The resulting optimization problem is typically nonconvex, because of the nonlinear dynamics of the reactors. An example application to the problem of grade transition in a continuous methyl methacrylate polymerization process, and also in a gas-phase polyethylene process, using nonlinear MPC and a Luenberger observer, can be found in [20].

A third approach is a hybrid of the first two. It involves the offline numerical computation of the optimal transition using a nominal model of the process. The computed profiles are by no means optimal for the plant due to uncertainty in the form of plant-model mismatch and disturbances. However, the resulting optimal transition can be specified in terms of the succession and the types of arcs—arcs that are either in the interior or on the boundaries of the feasible region, reflecting respectively, the fact that they either force a sensitivity to zero or keep a constraint active. The result is a "solution model" that expresses

in very practical terms the necessary conditions of optimality (NCO) that need to be satisfied by the plant. These NCOs are enforced using feedback and appropriate online measurements. In other words, these arcs and the corresponding switching times between them are adapted online via feedback control using typically a multiloop PID control structure. The approach has been referred to as NCO-tracking and falls within the class of self-optimizing controllers. Applications of this third approach to industrial polymerization processes can be found in [21–23].

12.4 Discontinuous Processes

12.4.1 Characteristics of Discontinuous Processes

A significant number of polymers are low-volume specialty materials manufactured for tailored applications. Such products are manufactured most efficiently using discontinuous (batch and semibatch) processes because of the intrinsic flexibility that these processes afford: batch and semibatch processes can be operated over relatively short periods of time, repeatedly, making them convenient for manufacturing a wide variety of low-volume products. The repetitive nature also permits batch-to-batch adjustments, facilitating quick adaptation to changes in quality specifications.

Although batch and semibatch polymerization processes share the common characteristics that they are *not* operated continuously, there are some important differences between them:

- In a batch process, the reactants are loaded into the reactor vessel at the beginning; the reactions then proceed without the further addition or removal of material until a prespecified reaction time has elapsed, after which the product is removed.
- In a semibatch process, one or several reactants can be added progressively to the reaction mixture or, in rare cases, products can be progressively removed from the reactor during the course of the reaction.

Nevertheless, within the context of this discussion, the overriding distinguishing characteristic of this class of processes is the repetitive, discontinuous operating policy, that is, the fact that they are *not* operated continuously. Hence, from this point on, the term "batch" will be used in a general sense that also encompasses semibatch processes, except when it is necessary to distinguish one from the other.

While continuous polymerization processes are predominantly operated at economically desirable steady states, batch polymerization processes operate permanently in "transient" mode, with process conditions and product characteristics constantly evolving from start to finish, so that there is no "steady-state" to speak of. Therefore, in addition to ensuring safe and economic operation, the main objective of batch polymerization reactor design and operation is to obtain, at the end of each batch cycle, material with acceptable product quality characteristics. As such, while under continuous operation the control system is designed to drive the polymerization process to the desired steady state operating point as quickly as possible—and maintain it there, the objective for batch polymerization is to follow time-varying policies/trajectories designed to produce, at the end of the batch, polymers with desired properties. Any effective batch control system must contend with this characteristic nonsteady-state process operation. The implications are as follows: with continuous processes, linear controllers have a reasonable chance of being effective precisely because of the operating objective of maintaining the process within a small neighborhood of the steady-state operating condition, where linear approximations are reasonably valid. With no such steady state for batch operation, linear approximations will be largely invalid and linear controllers largely ineffective.

In addition to the absence of a steady state, some other key features of batch polymerization processes that influence control system effectiveness include:

1. *Broad operating ranges:* batch operation spans conditions that extend from the beginning of the batch, with only reactants in the reactor, to the end of the batch, with mostly finished products.

2. *Single equipment, multiple products:* the same batch or semibatch reactor is usually used for the production of many different polymer products.
3. *Repetitiveness:* batch processes are used for low-volume production, necessitating frequent repetitions of batch runs.

The first two features pose challenges to effective control by demanding the use of sensors capable of covering a broad range of values, and also by placing stringent robustness and performance demands on controllers that must function over such broad ranges of operation. Because polymerization reactor dynamics depend strongly on the chemical composition and type of the feed (reactants), the second feature requires the use of control schemes and/or controller tunable parameters that can be modified appropriately for each set of reactants, even when the same equipment (reactor) is used. The third feature, on the other hand, can be advantageous because the results from prior runs can be used to improve the operation of subsequent ones (through "run-to-run" control and optimization schemes to be discussed subsequently).

The issue of lack of online measurement and its negative effect on control system design and implementation is also commonly found in batch processing.

12.4.2 Control of Batch Polymerization Processes I: Feedback Control

As noted previously, the diversity of reaction mechanisms and polymerization processes creates a wide variety of problem-specific issues. Nevertheless, there is a fundamental, defining characteristic of the control of batch polymerization processes: the requirement to track desired trajectories from the beginning to the end of the run, with the objective that the final product has desired properties. Consequently, as discussed in greater detail, effective control strategies involve the following two-steps:

1. Determine, offline, the input and output profiles (trajectories) required to manufacture a product with the desired quality.
2. Design a control scheme to track selected desired profiles as closely as possible.

What distinguishes one strategy from another is how each step is realized in practice. In general, the reference trajectories are determined either from accumulated experience or knowledge of the process and the product, or by solving a dynamic optimization problem. The computation of trajectories via optimization will be addressed in Section 12.4.3. Here, we focus on the control strategies that can be used to follow prescribed trajectories.

In general, trajectory tracking can be achieved in one of two ways: (1) via "online control" in an individual batch, whereby input adjustments are made online as the batch progresses, or (2) via "run-to-run" control over several batches, where the adjustments are *not* made online during a batch, but rather input trajectories are computed between batches, using information gathered from the previous batches to determine how to operate the next one.

In the context of these two control approaches, one needs to distinguish between *run-time outputs*, which are output variables that evolve with "run-time" during a batch, and *run-end outputs*, which consist of process values available at the end of the batch, such as the product quality (measured offline), batch time, and the maximum reactor temperature.

12.4.2.1 Online Control

For an individual batch, online control consists in tracking profiles that have been predetermined offline. Although the entire activity involved in implementing this strategy is repeated for each batch, what is done for one batch does not (formally) carry over to what is done for the next one. However, the objective is to track profiles that were pre-computed using nominal models, since the resulting control system performance will *not* be optimal in the presence of plant-model mismatch and unmodeled process disturbances. How the reference profiles can be adapted in the presence of such uncertainties is the focus of Section 12.4.3.

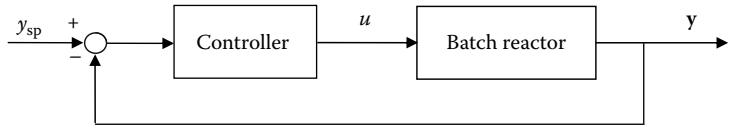

FIGURE 12.5 Online control of the run-time outputs y in a batch reactor.

With *online control of run-time outputs*, the most popular strategy consists primarily in tracking the pre-computed temperature profile by adjusting the reactor's heating/cooling rate. Sometimes, a second reference profile for the product average molecular weight (as an example) can also be tracked, typically by adjusting the feed rate of chain transfer agent. However, such a control strategy requires the use of an observer to reconstruct the average molecular weight of the product in the reactor from other (possibly infrequent) measurements. Figure 12.5 shows a generic block diagram representation of online control of the run-time outputs y in a batch reactor. Depending on the amount of process knowledge available, the controller can take several forms, from simple linear controllers to more elaborate nonlinear ones [24].

A few industrial applications of advanced control within the context of online control of run-time outputs are available in the literature. For example, a successful implementation of temperature control in an industrial 35 m^3 semibatch polymerization reactor using a flatness-based two-degree-of-freedom controller is reported in [25]. The performance of four different controllers (standard PI control, self-tuning PID control, and two nonlinear controllers) for regulating the reactor temperature in a 5 L jacketed batch suspension methyl methacrylate polymerization reactor is compared in [26]. As expected, the performance of the standard PI controller was the poorest since the controller parameters were fixed and not adapted to match the changing process characteristics. The self-tuning PID control, based on adaptive pole cancelation [27], performed better because available measurements were used to adapt the controller to the varying process characteristics. The two nonlinear controllers were based on differential-geometric techniques, which requires full-state measurement [28]; this necessitates the implementation of an extended Kalman filter to estimate the unavailable states from the available measurements. The two nonlinear controllers, which differ in the models on which each one is based, showed excellent performance despite significant uncertainty in the heat-transfer coefficient.

With a sufficiently accurate process model, and in the absence of disturbances, tracking the profiles determined offline is often sufficient to meet the batch-end product quality requirements. However, in the presence of disturbances, following prespecified profiles is unlikely to lead to the desired product quality. Hence the question arises: Is it possible to design an *online* control scheme for effective control of *run-end* outputs using *run-time* measurements? Since such an approach amounts to controlling a quantity that has not yet been measured, it is necessary to *predict* run-end outputs in order to compute the requisite corrective control action, using, for example, MPC. This approach to batch control may, therefore, be formulated as an MPC problem with a shrinking prediction horizon (equal to the time remaining to the end-of-batch), and an objective function that penalizes deviations from the desired product quality at batch end. With this strategy, at each sampling time during the batch, a piecewise-constant profile of future control moves is computed as the solution of the MPC optimization; and, in classic, MPC fashion, the first control move is implemented, and the states are reinitialized at the next sampling instant using process measurements or state estimates. The procedure is repeated until the end of the batch.

Within the context of batch process control, linear MPC is widely used in industry; not so with nonlinear MPC, but it remains an active area of research. For example, modifications of nonlinear MPC have been proposed for maintaining run-end molecular properties within bounds (in line with industry specifications) as opposed to controlling them at fixed values [29]. The recommended methodology has been applied in simulation to the control of batch styrene emulsion polymerization.

12.4.2.2 Run-to-Run Control

The objective of run-to-run control is to exploit one of the peculiarities of batch processes—their repetitive nature—in such a way that the procedure for controlling the current batch explicitly incorporates relevant information from previous batches. The input profiles are *not* adjusted online; instead, at the end of the kth batch, the input profiles $u_k[0, t_f]$ and the resulting product quality are used to determine the next input profiles $u_{k+1}[0, t_f]$, where t_f represents the final time.

The run-to-run adjustments serve to meet run-end targets and can be implemented in several different ways. For instance, the difference between predicted and actual run-end outputs can be used to refine the process model, and the updated model is subsequently used to adapt the controller parameters. Alternatively, the input profiles can be parameterized and the input parameters can be adapted from one run to the next in order to enforce run-end objectives. This latter run-to-run control approach, depicted schematically in Figure 12.6, can be expressed algorithmically as follows:

1. Parameterize the input profiles, $u_k[0, t_f] = \mathcal{U}(\pi_k)$, where π_k represents the vector of input parameters.
2. Start with the first run: set $k = 1$ and initialize π_k.
3. Implement the entire kth input profile, open loop; at the end of the batch, measure the run-end polymer properties z_k.
4. Determine the difference between the measured and desired run-end outputs and compute the $(k+1)$th values of the input parameters. For example, with integral run-to-run control, $\pi_{k+1} = \mathcal{I}(z_k, z_{sp})$, so that $u_{k+1}[0, t_f] = \mathcal{U}(\pi_{k+1})$.
5. Set $k := k + 1$ and return to (3).

Thus, while each batch is operated in open-loop fashion, feedback is introduced by updating the input profiles between the kth and $(k+1)$th batches. However, it is important to note that although a batch reactor is a highly dynamic process, the input parameters, π_k, have been related to the run-end outputs, z_k, in this control formulation via a *static map*. In actual fact, because the input parameters are fixed at the *beginning* of the batch, and the run-end outputs are evaluated at the *end* of the same batch, the plant dynamics, which are responsible for transforming the reactants available at the beginning of the batch to the products obtained at the end of the batch, have been incorporated into this static map. The process dynamics are therefore implicit in this apparently static relation. Note also that, because of its recursive nature, an important issue for this strategy is the convergence of the run-to-run control law.

Run-to-run control of run-end outputs has the drawback that it does not use any online information that is otherwise available through the run-time outputs. One way by which run-time information from previous batches can be used to control the run-time outputs of the current batch is through active and progressive "learning" of the run-time characteristics of the process. Such an approach, known as *iterative learning control* (ILC), is depicted in Figure 12.7. With ILC, the input profiles are adapted from one run to the next on the basis of the error between the reference output trajectories $y_{sp}[0, t_f]$ and each observed trajectory $y_k[0, t_f]$ with the objective of reducing this error at each iteration. Although the processing objective is still to meet run-end product requirements, ILC focuses on the run-time outputs as an *indirect* means of achieving run-end objectives.

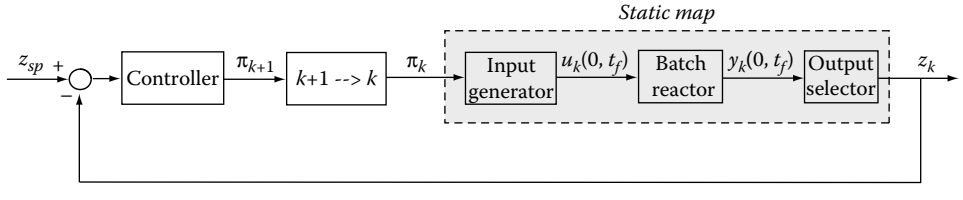

FIGURE 12.6 Run-to-run control of the run-end outputs z in a batch reactor.

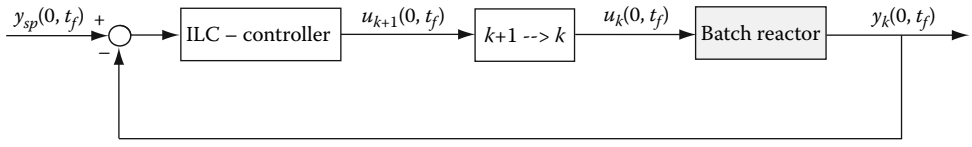

FIGURE 12.7 Iterative learning control of the run-time output profiles $y[0, t_f]$ in a batch reactor.

An example of successful industrial application of ILC can be found in [30], where an adaptive mechanism is used to learn about and compensate for variations in the reactor heat-transfer coefficient, resulting in a shorter heat-up phase for subsequent batches. This approach has now become standard in many industrial applications [31].

The main drawback of all run-to-run approaches is that they are unable to reject true run-time disturbances because the run-to-run strategy makes no provision for online feedback correction during the batch run. The obvious implication is that a combination of online and run-to run control approaches should provide improved performance, especially in the presence of significant run-time disturbances.

12.4.2.3 Combined Online and Run-to-Run Control

The fact that the control strategies discussed in the previous two sections complement each other well seems to suggest that one should be able to combine them judiciously in order to meet *both* run-time and run-end output objectives. A word of caution is necessary here, however: since both schemes attempt to adjust the same input variables, one must be careful to ensure that the online and between-run corrections do not oppose each other. Such a hybrid strategy is discussed in [32], where online MPC and run-to-run control are combined to control particle-size distribution of an emulsion polymerization product, both in simulation and experimentally.

Another example of the successful combination of online and run-to-run control is presented in [33]. The temperature profile of a batch polymerization reactor is separated into two sequential arcs tracked using online control, while the switching time between the two arcs, and the final time of the batch, are each adjusted run-to-run.

12.4.3 Control of Batch Polymerization Processes II: Optimal Control

As an alternative to "feedback" approaches discussed in Section 12.4.2, both within-batch and batch-end objectives can be met *simultaneously* and directly by formulating—and solving—the batch polymerization control problem as an optimization problem. This approach takes full advantage of the rich literature on general optimal control, appropriately adapted for batch polymerization processes. A comprehensive survey is given in [34], which contains nearly 140 references dealing with various aspects of polymerization modeling, control, and optimization. The main conceptual difference between the objectives discussed in Section 12.4.2 and those of the current section lies in the introduction of an economic performance criterion to be optimized. Run-time and run-end objectives are now seen as *constraints*. While optimal control strategies for batch polymerization processes have been widely studied in the literature, only a handful of successful applications to industrial reactors have been reported. One such application is presented later in this section as an illustrative example.

12.4.3.1 Problem Formulation

The optimization of a batch polymerization process can be formulated mathematically as follows:

$$\min_{u_k[0,t_f],\rho} \quad J_k = \phi(x_k(t_f)) + \int_0^{t_f} L(x_k(t), u_k(t), t)\, dt \tag{12.1}$$

$$\text{s.t.:} \quad \dot{x}_k(t) = F(x_k(t), u_k(t), \rho); \quad x_k(0) = x_{k,0}(\rho) \tag{12.2}$$

$$S(x_k(t), u_k(t), \rho) \leq 0 \tag{12.3}$$

$$P(x_k(t_f), \rho) \leq 0 \tag{12.4}$$

where J_k is the scalar cost to be minimized for the kth batch, S, the run-time constraints, P, the run-end constraints, ρ, the vector of time-invariant decision variables, and t_f, the "fixed" or "free" final time. If t_f is free, in the sense that it is to be determined as part of the optimization (rather than fixed, in the sense of having been determined *a priori*), then it is included in ρ. ϕ is the scalar cost associated with the final states, and L is the Lagrangian function. Note that the initial conditions can also be considered as decision variables.

For a specific process, P will represent a set of bounds on, for example, the batch-end weight-average molecular weight (the polymer product property of interest), while S will represent bounds on the manipulated variables and operational constraints such as physical limits on the heat-removal capacity of the reactor. For the purpose of illustration, a specific optimization problem could be formulated as follows:

$$\min_{t_f, T_{j,in}(t)} t_f \tag{12.5}$$

$$\text{s.t.} \quad \dot{x}(t) = F(x(t), T_{j,in}(t)), \quad x(0) = x_0 \tag{12.6}$$

$$X(t_f) \geq X_{min} \tag{12.7}$$

$$\bar{M}_w(t_f) \geq \bar{M}_{w,min} \tag{12.8}$$

$$T_{j,in}(t) \geq T_{j,in,min} \tag{12.9}$$

$$T_r(t) \leq T_{r,max} \tag{12.10}$$

where T_r is the reactor temperature and $T_{j,in}$, the jacket inlet temperature. F represents the process model equations, with the n-dimensional state vector x and the associated initial conditions x_0. It is usual to divide the system constraints into two categories:

- *Terminal constraints* (Equations 12.7 and 12.8): These are constraints on the final values taken by certain variables at the end of the batch. X_{min} is the lower bound on the final conversion, $X(t_f)$, and $\bar{M}_{w,min}$ is the lower bound on the final weight-average molecular weight, $\bar{M}_w(t_f)$. The lower bound on conversion is to ensure the production of an adequate amount of polymer in addition to preventing the accumulation of a toxic monomer; the lower bound on the average molecular weight guarantees the quality of the polymer.
- *Path constraints* (Equations 12.9 and 12.10): These are constraints on the values of the process variables during the course of the batch. In this specific example, $T_{j,in,min}$ is the minimal value that the jacket inlet temperature can take, and $T_{r,max}$ is the maximal allowed value of the reactor temperature $T(t)$.

Solving this specific problem will produce a jacket inlet temperature profile which, in the absence of disturbances or process-model mismatch, minimizes batch time, while guaranteeing that the conversion and average molecular weight specifications are met.

12.4.3.2 Solving Dynamic Optimization Problems

There exist several approaches for solving dynamic optimization problems, each one with its own peculiar features. For example, for low-order dynamical systems, variational approaches such as Pontryagin's maximum principle, or differential-geometric techniques, lead to analytical expressions for the various arcs that constitute the optimal control profiles. In practice, however, optimization problems are generally solved numerically using a variety of techniques that include sequential quadratic programming (SQP), genetic algorithms and stochastic optimization.

Regardless of the specific solution technique employed, it is important to keep in mind that the resulting solution, because it is based on a nominal model, will be valid and effective only to the extent that the model idealization matches reality. In practice, there is plenty of uncertainty in the form of plant-model mismatch and disturbances. Under these conditions, for the optimal control approach to be effective, appropriate corrective measures must be incorporated into the implementation strategies. These issues, which are central to the practice of optimal control of batch polymerization processes, are discussed next.

12.4.3.3 Implementation Strategies

Once the model-based optimization problem (Equations 12.1 through 12.4) has been solved numerically offline, *open-loop implementation* of the computed input trajectories will only be "optimal" if there are no disturbances and the process model is perfect. In actual practice, in the face of disturbances, it will be necessary to augment this basic strategy with active feedback and track the computed optimal output trajectories. An application of this approach to an emulsion copolymerization process is given in [35].

On the other hand, the repercussions of plant-model mismatch are that the input and output trajectories, computed via optimization based on a nominal model, will no longer be optimal for the plant. Even worse, these trajectories may correspond to infeasible paths that violate safety or operational constraints. Thus, one either needs to consider the effect of uncertainty explicitly in the optimization problem (robust optimization) or else adapt the trajectories online using process measurements (online optimization).

12.4.3.3.1 Robust Optimization

To prevent constraint violation, it may be necessary to account for uncertainty explicitly in the computation of the optimal profiles. For example, by formulating the optimization problem with a set of possible values for the uncertain parameters, a robust optimal solution may be determined that guarantees that the constraints are satisfied over the entire set of values specified for the uncertain parameters. This approach, known as *robust optimization*, obviously endows the solution with robustness and can lead to satisfactory results when the uncertainty region is small. With significant model uncertainty, the robust optimal solution is often unnecessarily conservative, thus leading to poor performance. Because polymerization processes are difficult to model, it is not unusual that the uncertainties associated with the model structure and the parameter values will be significant and, therefore, robust optimization will enjoy only limited effectiveness.

12.4.3.3.2 Online Optimization

An alternative strategy for avoiding the inevitable conservatism of robust optimization is to deal with the uncertainty actively by incorporating measurements into the optimization framework, with the premise that, when available, measurements provide the best up-to-date information about the real process behavior. This strategy can be implemented in different ways, as described next.

1. *Repeated, updated optimization:* In this two-step approach, the idea is to use measurements to update the optimization problem and repeat the optimization. There are different ways of updating the optimization problem:
   - Update the initial conditions of the *subsequent* optimization problem—as with MPC, when the output measurements or state estimates serve as new initial conditions for the subsequent optimization problem.
   - Identify the uncertain model parameters and update the process model.
   - Identify specific deviations between the plant and the model prediction (e.g., the constraint values), and correct the optimization problem formulation accordingly.

None of these methods is a panacea; they all have strengths and weaknesses [36]. Nevertheless, with careful consideration of the problem at hand, it is often possible to apply these repeated optimization approaches successfully to actual batch polymerization reactors, as reported in [37]. As was the case with feedback control, the repetitive nature of batch processes can also be exploited in the context of repeated optimization. The optimization follows essentially the same steps as discussed above, the main difference being that the update step is performed between consecutive batches instead of at the sampling times during a batch. This way, much more data are available (e.g., from the previous completed batches).

2. *Self-optimizing control:* In self-optimizing-control, optimality is achieved via feedback control, not by repeatedly solving an optimization problem. The key feature is the design of a feedback control structure that enforces optimal *plant* performance; and a primary means of enforcing optimality through feedback is by choosing the set-points as the NCO of the optimization problem.

   The optimal solution of a dynamic optimization problem consists of one or several arcs, with each one either enforcing an active constraint or else forcing a sensitivity to zero. The control structure is constructed as follows: (i) use a plant model to compute, offline, the nominal optimal solution (which will not be optimal for the plant because of uncertainty); (ii) design a multiloop control system such that each loop regulates a specific element of the NCO to zero; and (iii) implement this feedback control using measurements or estimates of the NCO elements. This approach has been labeled *NCO tracking* because it is designed to satisfy the NCO of the plant [38].

   The main difficulty with the implementation of NCO tracking is the real-time computation of the NCO elements. However, this difficulty is offset by the following facts: (i) constrained quantities are typically measured and thus directly available for feedback control, and (ii) accurate estimation of sensitivities is often not needed as there is generally much more to gain in terms of cost improvement by enforcing the set of active constraints than by forcing the sensitivities to zero [39].

   An alternative to *online* self-optimizing control is *run-to-run* self-optimizing control, which involves reformulating the optimization problem as a control problem. With this run-to-run alternative, there is the additional advantage of being able to wait until the end of the batch and thus accumulate more data. By nature, the run-to-run self-optimizing control strategy mimics the way the performance of batch processes is improved in industry. For instance, in the case of an isothermal batch polymerization process, one typically tries to increase the reactor temperature gradually from batch to batch in order to reduce the final time. This procedure stops when the operator ascertains that the process is sufficiently close to its constraints, and that increasing the reactor temperature any further will lead to off-spec products.

   It must be clear from the above that, although very attractive conceptually, self-optimizing control can be limited by the lack of appropriate measurements. Hence, state estimation is essential for successful implementation of this strategy. One such successful industrial application of self-optimizing control is discussed next.

12.4.3.4 Illustrative Example: Optimization of an Industrial Batch Polymerization Reactor

The problem involves a batch inverse-emulsion copolymerization process, where the objective is to minimize the reaction time [33]. (An inverse-emulsion polymerization process is so-called because, contrary to standard emulsion polymerization processes, which involves oil-soluble monomers and water-soluble initiators, here the monomers are water-soluble and emulsified in the oil-phase, while the polymerization is initiated with an oil-soluble initiator.) The reaction is highly exothermic; as such, higher reactor temperatures will speed up the reaction and hence reduce reaction time. However, the resulting heat generation will significantly increase the risk of thermal runaway, in addition to being potentially detrimental to the final product quality (especially average molecular weight). As a result, the prevalent

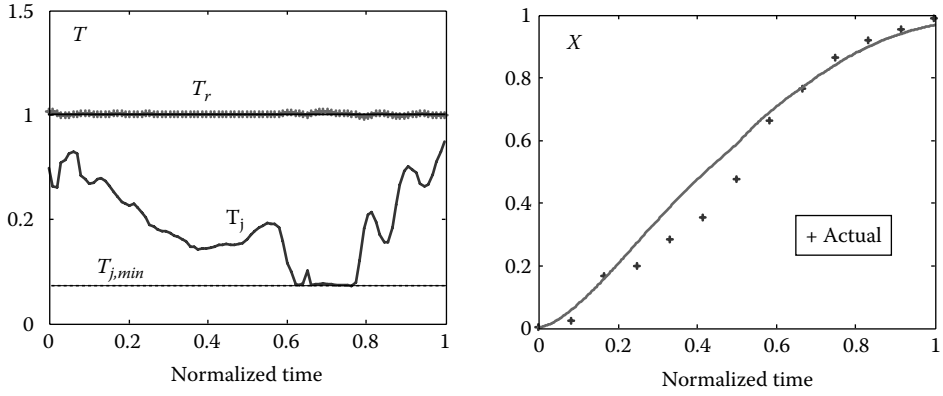

FIGURE 12.8 Normalized measured profiles in an industrial batch reactor. T_r is the reactor temperature, T_j, the jacket temperature, and X, the molar conversion. The solid conversion line corresponds to the prediction of a simple model of the polymerization reactor (Adapted from Francois, G., et al., *Ind. Eng. Chem. Res.*, 43, 7238–7242, 2004.). Note the loss of controllability in the interval [0.6, 0.8] when the manipulated variable T_j is at its lower bound. This loss of controllability was compensated for manually by reducing the initiator feed.

practice in industry has been to operate isothermally at a safe (relatively low) temperature. Reactor performance can be improved by determining the reactor temperature profile that provides the best possible tradeoff between productivity on the one hand, and safety and quality on the other. Figure 12.8 shows a set of temperature and molar conversion profiles representative of industrial practice in this 1 m^3 reactor.

While a detailed discussion of the design and implementation of a run-to-run measurement-based optimization strategy for this problem can be found in [33], the main points are summarized here.

1. Numerical optimization based on a simple process model indicates qualitatively that the first phase of the reaction must be nearly isothermal so as not to violate the heat removal constraint, while the temperature is allowed to increase in the latter part of the reaction in order to reduce reaction time. Indeed, once conversion has reached a certain value, the reaction rate decreases naturally and the reactor temperature can be increased (to reduce the batch time) while still meeting the terminal specification on the average molecular weight. The resulting temperature policy is therefore approximately semiadiabatic, whereby an initial isothermal phase is followed by an adiabatic phase.
2. The temperature at the final time, t_f, must respect a prescribed limit, above which the polymer starts to coagulate. Since the maximum temperature that will be reached depends on the amount of unreacted monomers remaining in the reactor when the control policy switches from isothermal to adiabatic operation, the switching time between the two phases is adjusted to meet this temperature constraint.

Consequently, the run-to-run optimization task amounts to specifying two scalar parameters:

i. t_s, the switching time between isothermal and adiabatic operations
ii. t_f, the final time

in order to meet two terminal constraints: (a) the desired conversion of monomer X_{des}, and (b) the maximum reactor temperature $T_{r,max}$. Since, for this process, X_{des} is nearly 100%, the desired conversion and the maximum temperature occur simultaneously. Therefore, the optimization problem reduces to a run-to-run adaptation of only the switching time, t_s, to satisfy $T_r(t_f) = T_{r,max}$. Such an adaptation scheme

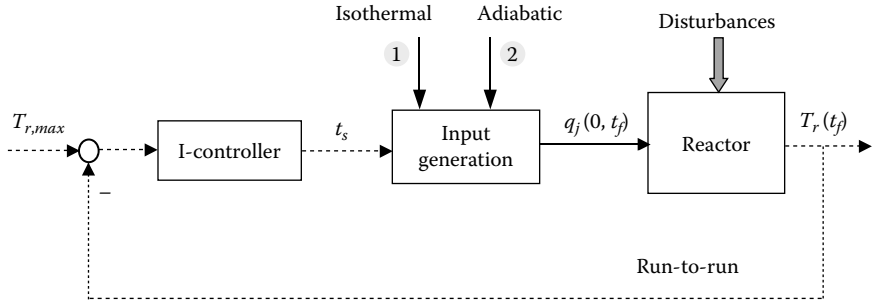

FIGURE 12.9 Run-to-run optimization scheme showing how the final reactor temperature target can be met by adjusting the switching time t_s. The reactor is initially operated isothermally at the normalized temperature $T_{r,ref} = 1$. Adiabatic operation starts at t_s, a value that is adjusted on a run-to-run basis to achieve $T_r(t_f) = T_{r,max}$. The true manipulated input is the flow rate of cooling medium in the jacket $q_j[0, t_f]$, which is determined in the isothermal phase by regulation of $T_r(t)$ around $T_{r,ref} = 1$ for $t < t_s$; in the adiabatic phase, $q_j[t_s, t_f]$ is simply set to 0.

is conveniently implemented with the discrete integral control law:

$$t_{s,k+1} = t_{s,k} + K(T_{r,max} - T_{r,k}(t_f)),$$

where K is the gain of the run-to-run integral controller. Figure 12.9 shows a block diagrammatic representation of the overall run-to-run scheme.

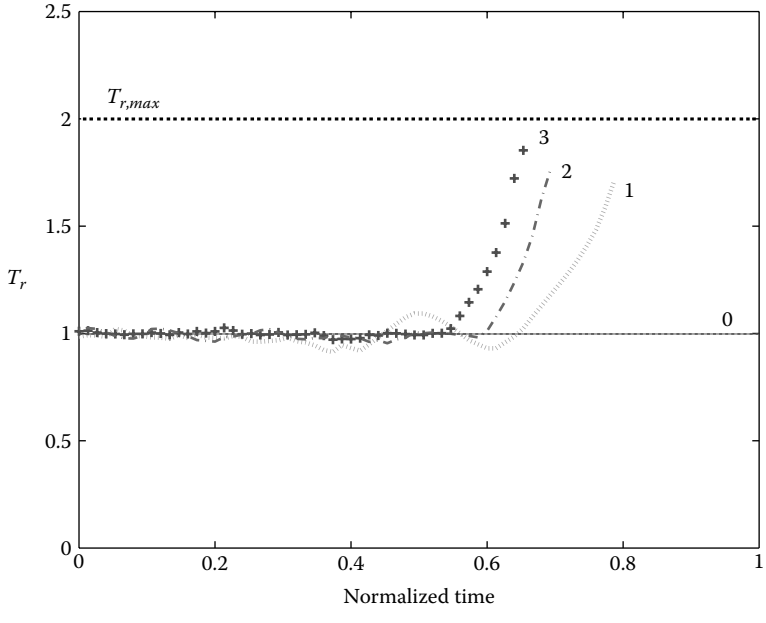

FIGURE 12.10 Normalized temperature profiles in the industrial reactor obtained with the run-to-run optimization scheme that adjusts the switching time between isothermal and adiabatic phases. Compared to the normalized reaction time of 1 for the current-practice isothermal operation, the reaction time is successively reduced to 0.78, 0.72, and 0.65 in three consecutive batches. Note that a slight "back-off" from $T_{r,max}$ is implemented for safety purposes, mainly to account for run-time disturbances that cannot be handled by this approach.

The improvement obtained with this scheme in three consecutive batches in the 1 m^3 industrial reactor is shown in Figure 12.10. The batch time is reduced by about 35%, while the final average molecular weight specifications are satisfied for all batches.

12.5 Summary and Conclusions

Process control continues to play an increasingly central role in the manufacture of high quality polymer products, safely and in an environmentally friendly fashion. In particular, because end-use properties of a polymer product strongly depend on the molecular and/or macroscopic architecture of the polymer—characteristics that are determined during polymer synthesis in a polymerization reactor and which cannot be altered in downstream processes or by blending—effective control of polymerization reactors is of great importance to the modern economic production of polymers. However, effective control of polymerization reactors is difficult because these reactors are typically highly exothermic, exhibit complex and highly nonlinear—and poorly understood—dynamics that are strongly dependent on the chemical type and composition of the reactants. Furthermore, the molecular properties that determine the polymer product performance in end-use are usually measured offline with very long time delays, so that these measurements are rarely available for online control.

Many of the modern methodologies employed for controlling polymer processes have been reviewed in this chapter. While the various techniques are different in specific details, their development, use, and implementation share the following three common steps, irrespective of the polymerization reactor type, whether continuous, batch, or semibatch.

1. Given a set of end-use properties desired of a polymer product, the first step is to determine the molecular and/or macroscopic architecture of the polymer (i.e., average molecular weights, functional group distribution, and copolymer composition—when two or more different monomers are involved) that will combine to yield the specified end-use properties.
2. As these molecular and macroscopic characteristics are rarely measured online, the second step is to determine the reactor temperature and feed flow rate profiles which, once implemented, will lead to the production of a polymer product with the desired molecular and macroscopic characteristics. These profiles can be piece-wise constant as in the case of continuous reactors, or entirely time-varying as in the case of semibatch or batch reactors. Of course, these profiles must be implemented in such a way that the reactor is not steered into unsafe operating regimes. If an accurate reactor model is available, the temperature and feed profiles can be calculated systematically offline and/or online using optimal control techniques and the model. The optimization methods reviewed in this chapter can be used to perform such calculations.
3. In the final step, feedback control is used to implement these profiles, which are presented as set-points for the temperature and feed flow rate feedback control system. If any infrequent measurements of polymer properties are available, they should be used in the feedback control system to improve performance.

While several of the techniques presented here have been implemented successfully in industrial practice, many challenging problems still remain. Here are a few of them. First, advances in optimization techniques in general, and in MPC in particular, have undoubtedly influenced industrial implementation; but these techniques depend on the availability of polymer reactor models of reasonably high fidelity *and* modest complexity. Achieving such an intrinsically contradictory balance in models of processes of practical importance remains difficult: industrial processes are, by nature, complex; capturing the essential components of such complex processes with models of sufficient high fidelity almost inevitably demands a minimum level of complexity. Methods of nonlinear model reduction capable of representing complex process dynamics effectively with reduced-order, control-relevant models will provide a significant boost to the industrial application of advanced control and optimization techniques. Next, while advances in

state estimation have enabled the estimation of infrequently measured product characteristics—and hence made online control of product characteristics possible—such estimates can never completely replace the actual measurements themselves. And as the manufacturing chain becomes inexorably more tightly integrated, each successive downstream customer will place increasingly stringent end-use performance demands on the polymer products they receive from each of their suppliers. Meeting these demands will ultimately require polymer process control system performance levels that are unattainable via unavoidably imperfect *inferred* product attributes. Advances in sensors, analyzers and ancillary measurement technology will be required in order to make available actual measurements of product attributes more frequently than is currently possible. Finally, as the polymer process control system structures acquire more complexity, theoretical analyses of model dynamics, overall control system stability, and achievable performance (both nominal and robust), will be essential, especially for providing guidance in selecting the best alternative for each problem.

References

1. J. MacGregor, Control of polymerization reactors, in *IFAC Symposium DYCORD 86*, Bournemouth, UK, pp. 21–36, 1986.
2. W. Ray, Modeling and control of polymerization reactors, in *3rd IFAC Symposium on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*, College Park, MD, p. 161, 1992.
3. F. Doyle III, M. Soroush, and C. Cordeiro, Control of product quality in polymerization processes, in *Chemical Process Control-VI*, J. B. Rawlings, B. A. Ogunnaike and J. W. Eaton (eds.), AIChE Symposium Series, New York, NY, pp. 290–306, 2002.
4. A. Schmidt and W. Ray, The dynamic behavior of continuous polymerization reactors-I: Isothermal solution polymerization in a CSTR, *Chem. Eng. Sci.*, vol. 36, no. 8, pp. 1401–1410, 1981.
5. J. Hamer, T. Akramov, and W. Ray, The dynamic behavior of continuous polymerization reactors-II: Nonisothermal solution homopolymerization and copolymerization in a CSTR, *Chem. Eng. Sci.*, vol. 36, no. 12, pp. 1897–1914, 1981.
6. A. Schmidt, A. Clinch, and W. Ray, The dynamic behavior of continuous polymerization reactors-III: An experimental study of multiple steady states in solution polymerization, *Chem. Eng. Sci.*, vol. 39, no. 3, pp. 419–432, 1981.
7. B. Bequette, *Process Control: Modeling, Design, and Simulation*. Upper Saddle River, NJ: Prentice Hall Press, 2002.
8. B. Ogunnaike and W. Ray, *Process Dynamics Modeling and Control*. Oxford University Press, New York, 1994.
9. D. Seborg, T. Edgar, and D. Mellichamp, *Process Dynamics and Control*, 2nd ed. John Wiley and Sons, New York, 2004.
10. D. Chien and A. Penlidis, On-line sensors for polymerization reactors, *Polymer Reviews*, vol. 30, no. 1, pp. 1–42, 1990.
11. W. Ray, Polymerization reactor control, *IEEE Contr. Syst. Mag.*, vol. 6, no. 4, pp. 3–8, 1986.
12. S. Ponnuswamy, S. Shah, and C. Kiparissides, On-line monitoring of polymer quality in a batch polymerization reactor, *J. Appl. Polymer Sci.*, vol. 32, no. 1, pp. 3239–3253, 1986.
13. F. Schork and W. H. Ray, On-line measurement of surface tension and density with application to emulsion polymerization, *J. Appl. Polymer Sci* , vol. 28, no. 1, pp. 407–430, 1983.
14. M. Soroush and C. Kravaris, Multivariable nonlinear control of a continuous polymerization reactor: An experimental study, *AIChE J.*, vol. 32, no. 12, pp. 1920–1937, 1993.
15. K. B. McAuley and J. MacGregor, On-line inference of polymer properties in an industrial polyethylene reactor, *AIChE J.*, vol. 37, no. 6, pp. 825–835, 1991.
16. N. Zambare, M. Soroush, and B. A. Ogunnaike, Multirate control of a polymerization reactor: A comparative study, in *Proc. of American Contr. Conf.*, San Diego, CA, USA, pp. 2553–2557, 1999.
17. B. Ogunnaike, On-line modeling and predictive control of an industrial terpolymerization reactor, *Int. of Control*, vol. 59, no. 3, pp. 711–729, 1994.
18. N. Zambare, M. Soroush, and B. Ogunnaike, A method of robust multi-rate state estimation, *J. Process Contr.*, vol. 13, no. 4, pp. 337–355, 2003.
19. K. B. McAuley and J. MacGregor, Nonlinear product gas-phase property control in industrial polyethylene reactors, *AlChE J.*, vol. 39, no. 5, pp. 855–866, 1993.

20. S. BenAmor, F. Doyle III, and R. MacFarlane, Polymer grade transition control using advanced real-time optimization software, *J. Process Contr.*, vol. 14, no. 4, pp. 349–364, 2004.
21. D. Bonvin, L. Bodizs, and B. Srinivasan, Optimal grade transition for polyethylene reactors via NCO tracking, *Trans IChemE Part A: Chemical Engineering Research and Design*, vol. 83, no. A6, pp. 692–697, 2005.
22. C. Chatzidoukas, C. Kiparissides, B. Srinivasan, and D. Bonvin, Optimization of grade transitions in an industrial gas-phase olefin polymerization fluidized bed reactor via NCO-tracking, in *16th IFAC World Congress*, Prague, Czech Republic, 2005.
23. J. Kadam, W. Marquardt, B. Srinivasan, and D. Bonvin, Optimal grade transition in industrial polymerization processes via NCO tracking, *AIChE J.*, vol. 53, no. 3, pp. 627–639, 2007.
24. M. Soroush and C. Kravaris, Nonlinear control of a batch polymerization reactor: An experimental study, *AIChE J.*, vol. 38, no. 9, pp. 1429–1448, 1992.
25. V. Hagenmeyer and M. Nohr, Flatness-based two-degree-of-freedom control of industrial semi-batch reactors using a new observation model for an extended Kalman filter approach, *Int. J. Control*, vol. 81, no. 3, pp. 428–438, 2008.
26. M. Shahrokhi and M. Ali Fanaei, Nonlinear temperature control of a batch suspension polymerization reactor, *Polymer Engineering and Science*, vol. 42, no. 6, pp. 1296–1308, 2002.
27. R. Ortega and R. Kelly, PID self-tuners: Some theoretical and practical aspects, *IEEE Trans. Ind. Electron.*, vol. IE-31, no. 4, pp. 332–338, 1984.
28. M. Soroush and C. Kravaris, Discrete-time nonlinear controller synthesis by input/output linearization, *AIChE J.*, vol. 38, no. 12, pp. 1923–1945, 1992.
29. J. Valappil and C. Georgakis, Nonlinear model predictive control of end-use properties in batch reactors, *AIChE J.*, vol. 48, no. 9, pp. 2006–2021, 2002.
30. K. Lee, S. Bang, J. Son, and S. Yoon, Iterative learning control of heat-up phase for a batch polymerization reactor, *J. Process Contr.*, vol. 6, no. 4, pp. 255–262, 1996.
31. Y. Wang, F. Gao, and F. J. Doyle, Survey on iterative learning control, repetitive control, and run-to-run control, *J. Process Contr* , vol. 19, no. 10, pp. 1589–1600, 2009.
32. M. Dokucu and F. Doyle III, Batch-to-batch control of characteristic points on the psd in experimental emulsion polymerization, *AIChE J* , vol. 54, no. 12, pp. 3171–3187, 2008.
33. G. Francois, B. Srinivasan, D. Bonvin, J. Hernandez Barajas, and D. Hunkeler, Run-to-run adaptation of a semi-adiabatic policy for the optimization of an industrial batch polymerization process, *Ind. Eng. Chem. Res.*, vol. 43, no. 23, pp. 7238–7242, 2004.
34. C. Kiparissides, Challenges in particulate polymerization reactor modeling and optimization: A population balance perspective, *J. Process Contr.*, vol. 16, no. 3, pp. 205–224, 2006.
35. C. Gentric, F. Pla, M. A. Latifi, and J. P. Corriou, Optimization and nonlinear control of a batch emulsion polymerization, *Chem. Eng. J.*, vol. 75, no. 1, pp. 31–46, 1999.
36. B. Chachuat, B. Srinivasan, and D. Bonvin, Adaptation strategies for real-time optimization, *Comput. Chem. Eng.*, vol. 33, pp. 1557–1567, 2009.
37. C. Kiparissides, P. Seferlis, G. Mourikas, and A. Morris, Online optimizing control of molecular weight properties in batch free-radical polymerization reactors, *Ind. Eng. Chem. Res.*, vol. 41, no. 24, pp. 31–46, 2002.
38. B. Srinivasan and D. Bonvin, Real-time optimization of batch processes by tracking the necessary conditions of optimality, *Ind. Eng. Chem. Res.*, vol. 46, no. 2, pp. 492–504, 2007.
39. S. Deshpande, B. Chachuat, and D. Bonvin, Parametric sensitivity of path-constrained optimal control: Towards selective input adaptation, in *American Control Conference 2009*, 2009, pp. 349–354. Available at http://infoscience.epfl.ch/record/128131.

13

Multiscale Modeling and Control of Porous Thin Film Growth

Gangshi Hu
University of California, Los Angeles

Xinyu Zhang
University of California, Los Angeles

Gerassimos Orkoulas
University of California, Los Angeles

Panagiotis D. Christofides
University of California, Los Angeles

13.1 Introduction

Modeling and control of thin film microstructure in thin film deposition processes has attracted significant research attention in recent years. Specifically, kinetic Monte Carlo (kMC) models based on a square lattice and utilizing the solid-on-solid (SOS) approximation for deposition were initially employed to describe the evolution of film microstructure and design feedback control laws for thin film surface roughness [1,2]. Furthermore, a method that couples partial differential equation (PDE) models and kMC models was developed for computationally efficient multiscale optimization of thin film growth [3]. However, kMC models are not available in closed form and this limitation restricts the use of kMC models for system-level analysis and design of model-based feedback control systems.

Stochastic differential equations (SDEs) arise naturally in the modeling of surface morphology of ultra-thin films in a variety of thin film preparation processes [4–8]. Advanced control methods based on SDEs have been developed to address the need of model-based feedback control of thin film microstructure. Specifically, methods for state/output feedback control of surface roughness based on linear [9–11] and nonlinear [12,13] SDE models have been developed.

In the context of modeling of thin film porosity, kMC models have been widely used to model the evolution of porous thin films in many deposition processes [14–17]. Deterministic and stochastic ordinary differential equation (ODE) models of film porosity were recently developed [18] to model the evolution of film porosity and its fluctuation and design model predictive control (MPC) algorithms to control film porosity to a desired level and reduce run-to-run porosity variability. More recently, simultaneous control of film thickness, surface roughness, and porosity within a unified control framework was addressed on the basis of a kMC thin film growth model using the deposition rate as the manipulated input [19]. However, in a practical thin film growth setting, the surface deposition rate cannot be manipulated directly but indirectly through manipulation of the inlet precursor concentration.

The present work addresses this practical consideration and focuses on simultaneous regulation of film thickness, surface roughness, and porosity in a multiscale model of a thin film growth process using the inlet precursor concentration as the manipulated input. Specifically, a continuous macroscopic PDE model is used to describe the dynamics of the gas phase. The thin film growth process is modeled via a microscopic kMC simulation model on a triangular lattice with vacancies and overhangs allowed to develop inside the film. The macroscopic and microscopic models are connected through boundary conditions. Distributed parameter and lumped dynamic models are developed to describe the evolution of the film surface profile and porosity. The developed dynamic models are then used as the basis for the design of a MPC algorithm that includes penalty on the deviation of film thickness, surface roughness, and film porosity from their respective set-point values. Simulation results demonstrate the applicability and effectiveness of the proposed modeling and control approach by applying the proposed controller to the multiscale process model.

13.2 Preliminaries

We consider a silicon thin film growth process in a low-pressure chemical vapor deposition (LPCVD) reactor, which is shown in Figure 13.1. Due to the large discrepancies of the time and length scales between the gas phase and the thin film growth, two different models are employed to describe the evolutions of the gas phase and of the thin film. Under the hypothesis of continuum, a PDE model derived from a mass balance is used to describe the species concentration in the gas phase. The thin film growth model is simulated through an on-lattice kMC model that uses a triangular lattice and allows overhangs and vacancies to develop inside the film. The two models are connected through boundary conditions, that is, the adsorption rate in the kMC model depends on the reactant concentration at surface following the reaction rate law.

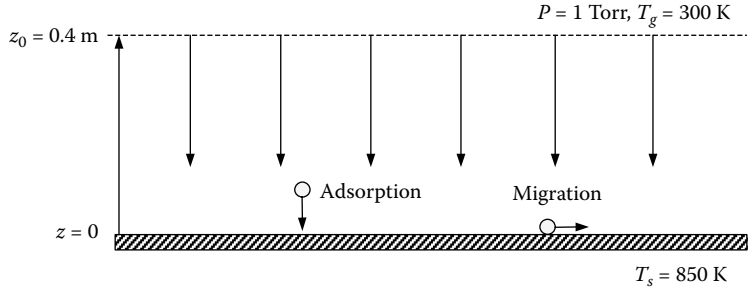

FIGURE 13.1 The multiscale model of a thin film growth process in a LPCVD reactor.

13.2.1 Gas-Phase Model

For the gas-phase model, a vertical, one-dimensional, stagnant flow geometry is considered. The inlet flow consists of two components, hydrogen and silane. Silane diffuses through a stagnant gas film of hydrogen. The temperature is constant throughout the gas phase. Thus, under the assumption of continuum, the silane concentration in the gas phase can be modeled via the following parabolic PDE:

$$\frac{\partial X}{\partial t} = D\frac{\partial^2 X}{\partial z^2} - KX \tag{13.1}$$

where X is the molar fraction of silane, D is the diffusivity of silane, the term $-KX$ accounts for the consumption of silane in the gas phase, that is, the gas-phase reaction and undesired sediments on reactor walls (we assume that this term is a first-order term).

The diffusivity, D, is calculated using a second-order polynomial of temperature as follows [20]:

$$D = c_0 + c_1 T_g + c_2 T_g^2, \tag{13.2}$$

where T_g is the gas-phase temperature set at 300 K, c_0, c_1, and c_2 are the coefficients of the polynomial.

The mass balance equation of Equation 13.1 is subject to the initial condition:

$$X(z, 0) = 0, \tag{13.3}$$

and the boundary condition at the inlet ($z = z_0 = 0.4$ m):

$$X(z_0, t) = X_{in}, \tag{13.4}$$

where X_{in} is the inlet concentration of silane, and the boundary condition at the wafer surface ($z = 0$):

$$CD\frac{\partial X}{\partial z}(0, t) = R_W, \tag{13.5}$$

where C is the molar concentration of the gas phase right above the surface and R_W is the reaction rate on the wafer surface. Under the assumption of ideal gas, $C = P/(RT_g)$, where P is the gas-phase pressure and R is the ideal gas constant.

When silane diffuses to the wafer surface, it decomposes into silicon and hydrogen as follows:

$$SiH_4 \rightarrow Si + 2H_2. \tag{13.6}$$

Then the silicon atoms are deposited onto the thin film. The reaction rate law on the surface is given as follows [20]:

$$R_W = \frac{kPX_s}{1 + K_H(P(1 - X_s))^{1/2} + K_s P X_s}, \tag{13.7}$$

where X_s is the silane concentration at wafer surface, k, K_H, and K_s are coefficients in the rate law. The coefficient k follows an Arrhenius-type law as follows [20]:

$$k = 1.6 \times 10^4 \exp(-18{,}500/T_s) \quad \text{mole m}^{-2}\,\text{s}^{-1}\,\text{Pa}^{-1}, \tag{13.8}$$

where T_s is the temperature of the wafer surface. The values of the parameters and coefficients of the gas-phase model can be found in Table 13.1.

13.2.2 On-Lattice kMC Model of Film Growth

The film growth model used in this work is an on-lattice kMC model in which all particles occupy discrete lattice sites [19,21]. The on-lattice kMC model is valid for a low-temperature region, $T < 0.5T_m$ (T_m is

TABLE 13.1 Gas-Phase Model Parameters

T_g	300 K	P	1 Torr
T_s	850 K	z_0	0.4 m
c_0	−2.90	K	0.5
c_1	2.06×10^{-2}	K_H	0.19 $\text{Pa}^{-1/2}$
c_2	2.81×10^{-5}	K_s	0.70 Pa^{-1}

the melting point of the crystal). A triangular lattice is selected to represent the crystalline structure of the film, as shown in Figure 13.2. The new particles are always deposited from the top side of the lattice where the gas phase is located. The number of sites in the lateral direction is defined as the lattice size and is denoted by L. In the triangular lattice, a bottom layer in the lattice is initially set to be fully packed and fixed, as shown in Figure 13.2. There are no vacancies in this layer and the particles in this layer cannot migrate. This layer acts as the substrate for the deposition and is not counted in the computation of the number of the deposited particles, that is, this fixed layer does not influence the film porosity. Two types of microscopic processes (Monte Carlo events) are considered: an adsorption process, in which particles are incorporated into the film from the gas phase, and a migration process, in which surface particles move to adjacent sites [14–16,22].

In an adsorption process, an incident particle comes in contact with the film and is incorporated onto the film. The microscopic adsorption rate, W, which is in units of layers per unit time, is equal to the surface reaction rate in the gas phase, R_W (i.e., $W = R_W$). The incident particles are initially placed at random positions above the film lattice and move toward the lattice in the vertical direction until contacting the first particle on the film. Upon contact, the particle moves (relaxes) to the nearest vacant site. Surface relaxation is conducted if this site is unstable, that is, with only one neighboring particle. When a particle is subject to surface relaxation, the particle moves to its most stable neighboring vacant site and is finally incorporated into the film.

In a migration process, a particle overcomes the energy barrier of the site and jumps to its vacant neighboring site. The migration rate (probability) of a particle follows an Arrhenius-type law with a precalculated activation energy barrier that depends on the local environment of the particle and the

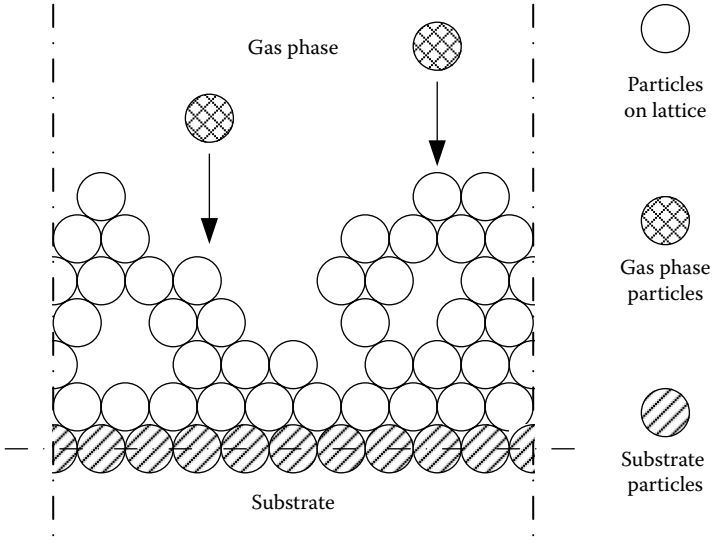

FIGURE 13.2 Thin film growth process on a triangular lattice.

substrate temperature. Since the film is thin, the temperature is assumed to be uniform throughout the film. The interior particles (the particles fully surrounded by six nearest neighbors) and the substrate layers cannot migrate.

When a particle is subject to migration, it can jump to either of its vacant neighboring sites with equal probability, unless the vacant neighboring site has no nearest neighbors, that is, the surface particle cannot jump off the film and it can only migrate on the surface. The deposition process is simulated using the continuous-time Monte Carlo (CTMC) method (see [18] for details).

13.2.3 Definitions of Surface Height Profile and Film Site Occupancy Ratio

Utilizing the CTMC algorithm, simulations of the kMC model of a porous silicon thin film growth process can be carried out. Snapshots of film microstructure, that is, the configurations of particles within the triangular lattice, are obtained from the kMC model at various time instants during process evolution. To quantitatively evaluate the thin film microstructure, two variables, surface roughness, and film porosity, are introduced in this subsection.

Surface roughness, which measures the texture of thin film surface, is represented by the root mean square (RMS) of the surface height profile of the thin film. Determination of surface height profile is slightly different in the triangular lattice model compared to a SOS model. In the SOS model, the surface of thin film is naturally described by the positions of the top particles of each column. In the triangular lattice model, however, due to the existence of vacancies and overhangs, the definition of film surface needs further clarification. Specifically, taking into account practical considerations of surface roughness measurements, the surface height profile of a triangular lattice model is defined based on the particles that can be reached in the vertical direction, as shown in Figure 13.3. In this definition, a particle is considered as a surface particle only if it is not blocked by the particles in the neighboring columns. Therefore, the surface height profile of a porous thin film is the line that connects the sites that are occupied by the surface particles. With this definition, the surface height profile can be treated as a function of the spatial coordinate. Surface roughness, as a measurement of the surface texture, is defined as the standard deviation of the surface height profile from its average height as follows [21]:

$$r = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(h_i - \bar{h})^2}, \tag{13.9}$$

where r denotes the surface roughness and $\bar{h} = \frac{1}{L}\sum_{i=1}^{L} h_i$ is the average surface height. Note that $r \geq 0$ and $r = 0$ corresponds to a flat surface.

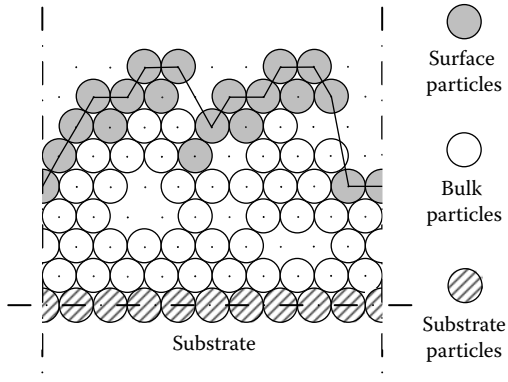

FIGURE 13.3 Definition of surface height profile. A surface particle is a particle that is not blocked by particles from both of its neighboring columns in the vertical direction.

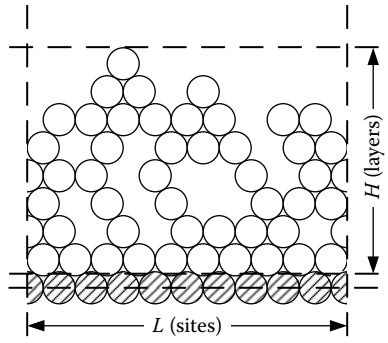

FIGURE 13.4 Illustration of the definition of film SOR of Equation 13.10.

In addition to film surface roughness, the film site occupancy ratio (SOR) is introduced to represent the extent of the porosity inside the thin film. The mathematical expression of film SOR is defined as follows:

$$\rho = \frac{N}{LH}, \tag{13.10}$$

where ρ denotes the film SOR, N is the total number of deposited particles on the lattice, L is the lattice size, and H denotes the number of deposited layers. Note that the deposited layers are the layers that contain only deposited particles and do not include the initial substrate layers. The variables in the definition expression of Equation 13.10 can be found in Figure 13.4. Since each layer contains L sites, the total number of sites in the film that can be contained within the H layers is LH. Thus, film SOR is the ratio of the occupied lattice sites, N, over the total number of available sites, LH. Film SOR ranges from 0 to 1. Specifically, $\rho = 1$ denotes a fully occupied film with a flat surface. The value of zero is assigned to ρ at the beginning of the deposition process since there are no particles deposited on the lattice.

13.2.4 Lattice Size Dependence of Film Surface Roughness and SOR

After defining the film surface roughness and SOR in Section 13.2.3, the dependence of film surface roughness and SOR on lattice size is investigated [23]. To investigate the lattice size dependence, kMC simulations of the thin film deposition process with different lattice sizes are carried out. The substrate temperature and the adsorption rate remain fixed throughout the entire simulation. The simulation time is set to be sufficiently long so that the steady-state values of surface roughness and SOR can be estimated. The expected values are averaged from multiple independent simulation runs.

The steady-state values of the expected surface roughness square exhibit strong dependence on the lattice size. This dependence can be addressed by plotting the steady-state values, $\langle r^2 \rangle_{ss}$, with respect to lattice size at $T = 300$ K and $W = 1$ layer/s; see Figure 13.5. The error bars in Figure 13.5 represent the range $\langle r^2 \rangle_{ss} \pm \sigma_r$, where σ_r denotes the standard deviation of $\langle r^2 \rangle_{ss}$ and is calculated from 10 averages of evenly divided groups of all simulation runs. The steady-state values of the expected surface roughness square are determined from the evolution profiles by averaging over the last 1000 points where steady state has been clearly reached. A linear dependence on the lattice size is clearly shown in Figure 13.5 for large lattice sizes, where the linear regression is obtained from the data points of lattice sizes $L \geq 100$ and the regression coefficient is 0.9997. The saturation time, that is, the time required for the expected surface roughness square to reach very close to its steady-state value, is proportional to the square of the lattice size.

Figure 13.6 shows the steady-state values of film SOR, ρ^{ss}, for different lattice sizes at $T = 400$ K and $W = 1$ layer/s. Figure 13.6 also includes the values of film SOR at $t = 1000$ s and $t = 2000$ s. The error bars of the film SOR are calculated from five averages of evenly divided groups of all simulation runs. A least-squares method is used to estimate the steady-state value of ρ_d^{ss} by fitting $\rho(t)$ data at finite times

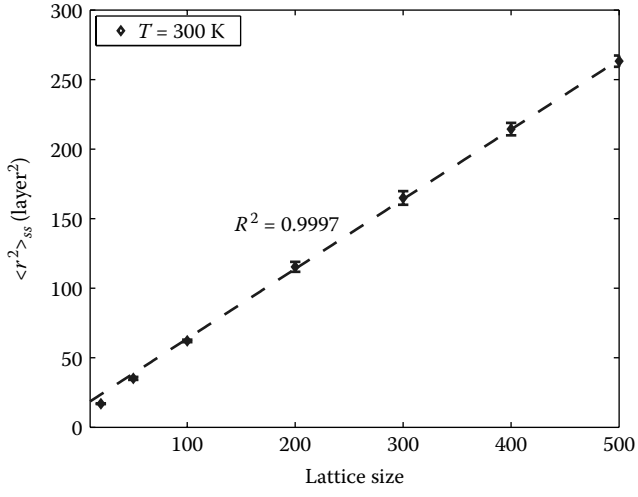

FIGURE 13.5 Dependence of the steady-state values of the expected surface roughness square, $\langle r^2 \rangle_{ss}$ (symbols with error bars), on the lattice size, and the linear regression, $\langle r^2 \rangle_{ss} = kL + b$, of the points corresponding to $L \geq 100$ (dashed line); $W = 1$ layer/s and $T = 300$ K.

obtained from the kMC simulation to an integral model of film SOR; see Section 13.3.2 for details of the integral model. Since the steady-state values are estimates from the evolution profiles, no error bars are presented with ρ^{ss}. For small lattice sizes, ρ^{ss} is very close to the values of film SOR at 1000 s and 2000 s, which indicates that the film SOR has reached its steady state. However, for large lattice sizes, the time $t = 2000$ s is not long enough for the film SOR to approach its steady state. A weak dependence of the steady-state film SOR is observed in Figure 13.6 (a logarithmic scale is used to show the large range of lattice sizes investigated in this work), where it can be seen that the steady-state value of the expected film SOR decays as the lattice size increases. The dynamics of film SOR also depends on the lattice size: large lattice sizes result in slow evolutions of the film SOR.

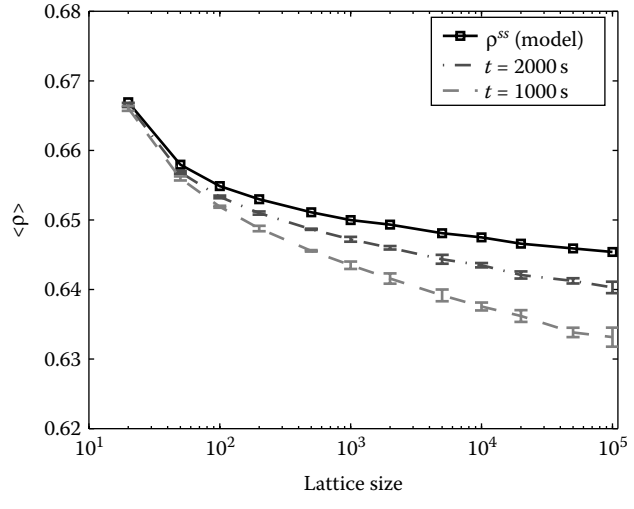

FIGURE 13.6 Profiles of the predicted steady-state values, the values at $t = 1000$ s and $t = 2000$ s of the expected film SOR for different lattice sizes; $W = 1$ layer/s and $T = 400$ K.

13.3 Dynamic Model Construction

13.3.1 Edwards–Wilkinson-Type Equation of Surface Height

An Edwards–Wilkinson (EW)-type equation, a second-order stochastic PDE, can be used to describe the surface height evolution in many microscopic processes that involve thermal balance between adsorption (deposition) and migration (diffusion). In this work, an EW-type equation is chosen to describe the dynamics of the fluctuation of surface height (the rigorous validation of this choice can be found in [23] and its validation in the context of control will be made clear below):

$$\frac{\partial h}{\partial t} = r_h + \nu \frac{\partial^2 h}{\partial x^2} + \xi(x,t) \tag{13.11}$$

subject to periodic boundary conditions:

$$h(-\pi,t) = h(\pi,t), \quad \frac{\partial h}{\partial x}(-\pi,t) = \frac{\partial h}{\partial x}(\pi,t) \tag{13.12}$$

and the initial condition:

$$h(x,0) = h_0(x), \tag{13.13}$$

where $x \in [-\pi, \pi]$ is the spatial coordinate, t is the time, r_h and ν are the model parameters, and $\xi(x,t)$ is a Gaussian white noise with the following mean and covariance:

$$\begin{aligned} \langle \xi(x,t) \rangle &= 0, \\ \left\langle \xi(x,t)\xi(x',t') \right\rangle &= \sigma^2 \delta(x-x')\delta(t-t'), \end{aligned} \tag{13.14}$$

where σ^2 is a parameter which measures the intensity of the Gaussian white noise and $\delta(\cdot)$ denotes the standard Dirac delta function.

To proceed with model parameter estimation and control design, a stochastic ODE approximation of Equation 13.11 is first derived using modal decomposition. Consider the eigenvalue problem of the linear operator of Equation 13.11, which takes the form:

$$\begin{aligned} A\bar{\phi}_n(x) &= \nu \frac{d^2 \bar{\phi}_n(x)}{dx^2} = \lambda_n \bar{\phi}_n(x), \\ \bar{\phi}_n(-\pi) = \bar{\phi}_n(\pi), &\quad \frac{d\bar{\phi}_n}{dx}(-\pi) = \frac{d\bar{\phi}_n}{dx}(\pi), \end{aligned} \tag{13.15}$$

where λ_n denotes an eigenvalue and $\bar{\phi}_n$ denotes an eigenfunction. A direct computation of the solution of the above eigenvalue problem yields $\lambda_0 = 0$ with $\psi_0 = 1/\sqrt{2\pi}$, and $\lambda_n = -\nu n^2$ (λ_n is an eigenvalue of multiplicity two) with eigenfunctions $\phi_n = (1/\sqrt{\pi})\sin(nx)$ and $\psi_n = (1/\sqrt{\pi})\cos(nx)$ for $n = 1, \ldots, \infty$. Note that the $\bar{\phi}_n$ in Equation 13.15 denotes either ϕ_n or ψ_n. For fixed positive value of ν, all eigenvalues (except the zeroth eigenvalue) are negative and the distance between two consecutive eigenvalues (i.e., λ_n and λ_{n+1}) increases as n increases.

The solution of Equation 13.11 is expanded in an infinite series in terms of the eigenfunctions of the operator of Equation 13.15 as follows:

$$h(x,t) = \sum_{n=1}^{\infty} \alpha_n(t)\phi_n(x) + \sum_{n=0}^{\infty} \beta_n(t)\psi_n(x), \tag{13.16}$$

where $\alpha_n(t), \beta_n(t)$ are time-varying coefficients. Substituting the above expansion for the solution, $h(x,t)$, into Equation 13.11 and taking the inner product with the adjoint eigenfunctions, $\phi_n^*(x) = (1/\sqrt{\pi})\sin(nx)$

and $\psi_n^*(x) = (1/\sqrt{\pi})\cos(nx)$, the following system of infinite stochastic ODEs is obtained:

$$\begin{aligned} \frac{d\beta_0}{dt} &= \sqrt{2\pi} r_h + \xi_\beta^0(t), \\ \frac{d\alpha_n}{dt} &= \lambda_n \alpha_n + \xi_\alpha^n(t), \quad n = 1, \ldots, \infty, \\ \frac{d\beta_n}{dt} &= \lambda_n \beta_n + \xi_\beta^n(t), \quad n = 1, \ldots, \infty, \end{aligned} \tag{13.17}$$

where

$$\xi_\alpha^n(t) = \int_{-\pi}^{\pi} \xi(x,t)\phi_n^*(x)\,dx, \quad \xi_\beta^n(t) = \int_{-\pi}^{\pi} \xi(x,t)\psi_n^*(x)\,dx. \tag{13.18}$$

The covariances of $\xi_\alpha^n(t)$ and $\xi_\beta^n(t)$ can be obtained: $\langle \xi_\alpha^n(t)\xi_\alpha^n(t')\rangle = \sigma^2\delta(t-t')$ and $\langle \xi_\beta^n(t)\xi_\beta^n(t')\rangle = \sigma^2\delta(t-t')$. Due to the orthogonality of the eigenfunctions of the operator in the EW equation of Equation 13.11, $\xi_\alpha^n(t)$ and $\xi_\beta^n(t)$, $n = 0, 1, \ldots$, are stochastically independent.

Since the stochastic ODE system is linear, the analytical solution of state variance can be obtained from a direct computation as follows:

$$\begin{aligned} \langle \alpha_n^2(t)\rangle &= \frac{\sigma^2}{2\nu n^2} + \left(\langle \alpha_n^2(t_0)\rangle - \frac{\sigma^2 s}{2\nu n^2}\right) e^{-2\nu n^2(t-t_0)}, \quad n = 1, \ldots, \infty, \\ \langle \beta_n^2(t)\rangle &= \frac{\sigma^2}{2\nu n^2} + \left(\langle \beta_n^2(t_0)\rangle - \frac{\sigma^2}{2\nu n^2}\right) e^{-2\nu n^2(t-t_0)}, \quad n = 1, \ldots, \infty, \end{aligned} \tag{13.19}$$

where $\langle \alpha_n^2(t_0)\rangle$ and $\langle \beta_n^2(t_0)\rangle$ are the state variances at time t_0. The analytical solution of state variance of Equation 13.19 will be used in the parameter estimation and the MPC design.

When the dynamic model of surface height profile is determined, surface roughness of the thin film is defined as the standard deviation of the surface height profile from its average height and is computed as follows:

$$r(t) = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} [h(x,t) - \bar{h}(t)]^2\,dx}, \tag{13.20}$$

where $\bar{h}(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} h(x,t)\,dx$ is the average surface height. According to Equation 13.16, we have $\bar{h}(t) = \beta_0(t)\psi_0$. Therefore, $\langle r^2(t)\rangle$ can be rewritten in terms of $\langle \alpha_n^2(t)\rangle$ and $\langle \beta_n^2(t)\rangle$ as follows:

$$\begin{aligned} \langle r^2(t)\rangle &= \frac{1}{2\pi}\left\langle \int_{-\pi}^{\pi} (h(x,t) - \bar{h}(t))^2\,dx \right\rangle \\ &= \frac{1}{2\pi}\left\langle \sum_{i=1}^{\infty} (\alpha_i^2(t) + \beta_i^2(t)) \right\rangle \\ &= \frac{1}{2\pi}\sum_{i=1}^{\infty} \left[\langle \alpha_i^2(t)\rangle + \langle \beta_i^2(t)\rangle\right], \end{aligned} \tag{13.21}$$

where $\bar{h} = \frac{1}{2\pi}\int_{-\pi}^{\pi} h(x,t)\,dx = \beta_0(t)\psi_0$ is the average of surface height. Thus, Equation 13.21 provides a direct link between the state variance of the infinite stochastic ODEs of Equation 13.17 and the expected surface roughness of the thin film. Note that the parameter r_h does not appear in the expression of surface roughness, since only the zeroth state, β_0, is affected by r_h but this state is not included in the computation of the expected surface roughness square of Equation 13.21.

Film thickness, which is represented by the average of surface height, $\bar{h}$, is another objective under consideration in this work. The dynamics of the expected value of averaged surface height can be obtained

from the analytical solution of the zeroth state, β_0, from Equation 13.17, as follows:

$$\frac{d\langle\bar{h}\rangle}{dt} = r_h. \tag{13.22}$$

The analytical solution of expected value of film thickness, $\langle\bar{h}\rangle$, can be obtained directly from Equation 13.22 as follows:

$$\left\langle\bar{h}(t)\right\rangle = \left\langle\bar{h}(t_0)\right\rangle + r_h(t - t_0). \tag{13.23}$$

13.3.2 Dynamic Model of Film SOR

The concept of film SOR is used to characterize film porosity. According to the definition of film SOR of Equation 13.10, film SOR accounts for all deposited layers during the entire deposition process. Thus, film SOR is a cumulative property, the evolution of which can be characterized by an integral form. Before further derivation of the dynamic model of film SOR, a concept of instantaneous film SOR of the film layers deposited between time t and $t + dt$, denoted by ρ_d, is first introduced as the spatial derivative of the number of deposited particles in the growing direction as follows:

$$\rho_d = \frac{dN}{d(HL)}. \tag{13.24}$$

In Equation 13.24, the lattice size L is a constant and the derivative dH can be written as a linear function of time derivative dt as follows:

$$dH = r_H\, dt. \tag{13.25}$$

where r_H is the growth rate of the thin film from the top layer point of view. Note that r_H is different from the model coefficient r_h in Equation 13.11. Thus, the expressions of N and H can be obtained by integrating Equations 13.24 and 13.25 as follows:

$$\begin{aligned} N(t) &= L\int_0^t \rho_d r_H\, ds, \\ H(t) &= \int_0^t \rho_d\, ds. \end{aligned} \tag{13.26}$$

With the definition of ρ of Equation 13.10 and the expressions of N and H of Equation 13.26, the film SOR of Equation 13.10 can be rewritten in an integral form as follows:

$$\rho = \frac{\int_0^t \rho_d r_H\, ds}{\int_0^t r_H\, ds}. \tag{13.27}$$

To simplify the subsequent development and develop an SOR model that is suitable for control purposes, we assume (this assumption will be verified in the closed-loop simulation results below where the performance of the controller will be evaluated) that the dynamics of the instantaneous film SOR, ρ_d, can be approximated by a linear first-order process, that is:

$$\tau\frac{d\rho_d(t)}{dt} = \rho_d^{ss} - \rho_d(t), \tag{13.28}$$

where τ is the time constant and ρ_d^{ss} is the steady-state value of the instantaneous film SOR. We note that the first-order ODE model of Equation 13.28 was introduced and justified with numerical results in [18] for the modeling of the partial film SOR, which is defined to characterize the evolution of the film porosity of layers that are close to the film surface. In this work, the instantaneous film SOR is a similar concept to the partial film SOR, because it also describes the contribution to the bulk film porosity of the newly

deposited layers. Therefore, the first-order ODE model is a suitable choice to describe the evolution of the instantaneous film SOR.

From Equation 13.27, it follows that at large times as ρ_d approaches ρ_d^{ss}, the steady-state film SOR (ρ^{ss}) approaches the steady-state value of the instantaneous film SOR (i.e., $\rho^{ss} = \rho_d^{ss}$). The deterministic ODE system of Equation 13.28 is subject to the following initial condition:

$$\rho_d(t_0) = \rho_{d0}, \tag{13.29}$$

where t_0 is the initial time and ρ_{d0} is the initial value of the instantaneous film SOR. From Equations 13.28 and 13.29 and the fact that $\rho^{ss} = \rho_d^{ss}$ at large times, it follows that

$$\rho_d(t) = \rho^{ss} + \left(\rho_{d0} - \rho^{ss}\right) e^{-(t-t_0)/\tau}. \tag{13.30}$$

For controller implementation purposes, the expression of the film SOR can be derived as follows:

$$\begin{aligned} \rho(t) &= \frac{\int_0^{t_0} \rho_d r_H \, ds + \int_{t_0}^{t} \rho_d r_H \, ds}{\int_0^{t_0} r_H \, ds + \int_{t_0}^{t} r_H \, ds} \\ &= \frac{\rho_0 H_0 + \int_{t_0}^{t} \rho_d r_H \, ds}{H_0 + \int_{t_0}^{t} r_H \, ds}, \end{aligned} \tag{13.31}$$

where t_0 is the current time, ρ_0 and H_0 are film SOR and film height at time t_0, respectively.

Substituting the solution of ρ_d of Equation 13.30 into Equation 13.31 and assuming that r_H is constant for $t > \tau > t_0$, which is taken to be the case in the parameter estimation and the MPC formulations below, the analytical solution of film SOR at time t can be obtained as follows:

$$\rho = \frac{\rho_0 H_0 + r_H \left[\rho^{ss}(t - t_0) + (\rho^{ss} - \rho_0)\tau(e^{-(t-t_0)/\tau} - 1)\right]}{H_0 + r_H(t - t_0)} \tag{13.32}$$

which is directly utilized in the MPC formulation of Equation 13.35 below.

13.4 Model Predictive Controller Design

In this section, a model predictive controller is designed based on the dynamic models of surface height and film SOR. The control objective is to regulate the expected values of roughness square and film SOR to desired levels by manipulating the inlet silane concentration. A desired value of film thickness is also included in the cost function in the MPC formulation. A reduced-order model of EW equation is used in the MPC formulation to approximate the dynamics of the surface roughness. State feedback control is considered in this work, that is, the surface height profile and the value of film SOR are assumed to be available to the controller.

13.4.1 Reduced-Order Model for Surface Roughness

In the MPC formulation, the expected surface roughness is computed from the EW equation (Equation 13.11). The EW equation, which is a distributed parameter dynamic model, contains infinite-dimensional stochastic states. Therefore, it leads to a model predictive controller of infinite order that cannot be realized in practice (i.e., the practical implementation of a control algorithm based on such a system will require the computation of infinite sums which cannot be done by a computer). To this end, a reduced-order model of infinite-dimensional ODE model of Equation 13.17 is instead derived and used to calculate the prediction of expected surface roughness in the model predictive controller.

Due to the structure of the eigenspectrum of the linear operator of the EW equation (Equation 13.11), the dynamics of the EW equation are characterized by a finite number of dominant modes. By neglecting the high-order modes ($n \geq m+1$), the system of Equation 13.17 can be approximated by a finite-dimensional system as follows:

$$\frac{d\alpha_n}{dt} = \lambda_n \alpha_n + \xi_\alpha^n(t), \quad \frac{d\beta_n}{dt} = \lambda_n \beta_n + \xi_\beta^n(t) \quad n = 1, \ldots, m. \tag{13.33}$$

Note that the ODE for the zeroth state is also neglected, since the zeroth state does not contribute to surface roughness.

Using the finite-dimensional system of Equation 13.33, the expected surface roughness square, $\langle r^2(t) \rangle$, can be approximated with the finite-dimensional state variance as follows:

$$\langle \tilde{r}^2(t) \rangle = \frac{1}{2\pi} \sum_{i=1}^{m} \left[\langle \alpha_i^2(t) \rangle + \langle \beta_i^2(t) \rangle \right] \tag{13.34}$$

where the tilde symbol in $\langle \tilde{r}^2(t) \rangle$ denotes its association with a finite-dimensional system.

13.4.2 MPC Formulation

We consider the control problem of film surface roughness, porosity, and thickness regulation by using a MPC design. The expected values, $\langle r^2 \rangle$, $\langle \rho \rangle$, and $\langle \bar{h} \rangle$, are chosen as the control objectives. The adsorption rate is computed by the controller, which, in turn, is used to calculate the inlet silane concentration via Equation 13.8 (i.e., the presence of the gas phase is neglected in the calculation of the control action, X_{in}, but it is accounted for in the multiscale process model, where the control action is applied). The substrate temperature is fixed at 850 K during the entire closed-loop simulation. The control action is obtained by solving a finite-horizon optimal control problem.

The cost function in the optimal control problem (Equation 13.35 below) includes penalty on the deviation of $\langle r^2 \rangle$ and $\langle \rho \rangle$ from their respective set-point values. However, since the manipulated input variable is the adsorption rate and the film deposition process is a batch operation (i.e., the film growth process is terminated within a certain time), a desired value of the film thickness is also required to prevent an undergrown thin film at the end of the deposition process. Therefore, in the MPC shown in Equation 13.35, the desired film thickness is regarded as the set-point value of the film thickness, that is, the deviation of the film thickness from the desired value is included in the cost function. However, only the negative deviation (when the film thickness is less than the desired value) is penalized; no penalty is imposed on the deviation when the thin film thickness exceeds the desired thickness. Different weighting factors are assigned to the penalties on the deviations of the expected values of film surface roughness, SOR, and thickness from their desired values. Relative deviations are used in the formulation of the cost function to make the magnitude of the different terms comparable. The optimization problem is subject to the dynamics of the reduced-order model of surface roughness of Equation 13.33, the dynamics of the film thickness of Equation 13.22, and the dynamics of the film SOR of Equation 13.27. The optimal profile of the adsorption rate is calculated by solving a finite-dimensional optimization problem in a receding horizon fashion. Specifically, the MPC problem is formulated as follows:

$$\min_{W_1,\ldots,W_i,\ldots,W_p} J = \sum_{i=1}^{p} \left\{ q_{r^2,i} F_{r^2,i} + q_{\rho,i} F_{\rho,i} + q_{h,i} F_{h,i} \right) \} \quad \text{subject to}$$

$$F_{r^2,i} = \left[\frac{r_{set}^2 - \langle r^2(t_i) \rangle}{r_{set}^2} \right]^2, \quad F_{\rho,i} = \left[\frac{\rho_{set} - \langle \rho(t_i) \rangle}{\rho_{set}} \right]^2$$

$$
\begin{aligned}
F_{h,i} &= \begin{cases} \left[\dfrac{h_{min}-\left\langle \bar{h}(t_i)\right\rangle}{h_{min}}\right]^2, & h_{min} > \left\langle \bar{h}(t_i)\right\rangle \\ 0, & h_{min} \le \left\langle \bar{h}(t_i)\right\rangle \end{cases} \\
\left\langle \alpha_n^2(t_i)\right\rangle &= \frac{\sigma^2}{2\nu n^2} + \left(\left\langle \alpha_n^2(t_{i-1})\right\rangle - \frac{\sigma^2}{2\nu n^2}\right) e^{-2\nu n^2 \Delta} \\
\left\langle \beta_n^2(t_i)\right\rangle &= \frac{\sigma^2}{2\nu n^2} + \left(\left\langle \beta_n^2(t_{i-1})\right\rangle - \frac{\sigma^2}{2\nu n^2}\right) e^{-2\nu n^2 \Delta} \\
\left\langle \bar{h}(t_i)\right\rangle &= \left\langle \bar{h}(t_{i-1})\right\rangle + r_h \Delta \\
\rho(t_i) &= \frac{1}{\left\langle \bar{h}(t_{i-1})\right\rangle + r_h \Delta} \cdot \left\{ \rho(t_{i-1}) \left\langle \bar{h}(t_{i-1})\right\rangle \right. \\
&\quad \left. + r_h \left[\rho^{ss} \Delta + (\rho^{ss} - \rho(t_{i-1}))\tau_p (e^{-\Delta/\tau_p} - 1)\right]\right\} \\
W_{min} &< W_i < W_{max}, \quad i = 1, 2, \ldots, p
\end{aligned}
\tag{13.35}
$$

where t is the current time, Δ is the sampling time, p is the number of prediction steps, $p\Delta$ is the specified prediction horizon, t_i, $i = 1, 2, \ldots, p$, is the time of the ith prediction step ($t_i = t + i\Delta$), respectively, W_i, $i = 1, 2, \ldots, p$, is the adsorption rate at the ith step ($W_i = W(t + i\Delta)$), respectively, $q_{r^2,i}$, $q_{h,i}$, and $q_{\rho,i}$, $i = 1, 2, \ldots, p$, are the weighting penalty factors for the deviations of $\langle r^2 \rangle$ and $\langle \rho \rangle$ from their respective set-points r_{set}^2 and ρ_{set}, $\langle \bar{h} \rangle$ from its desired h_{min}, at the ith prediction step, and W_{min} and W_{max} are the lower and upper bounds on the deposition rate, respectively. Note that we choose $\langle \bar{h} \rangle$, r_h, and $\rho(t_0)$ to replace H, r_H, and ρ_{d0} in the MPC formulation of Equation 13.35, respectively.

The optimal set of (W_1, W_2, ..., W_p), is obtained from the solution of the multi-variable optimization problem of Equation 13.35, and only the first value of the manipulated input trajectory, W_1, is used to compute the inlet silane concentration and is applied to the deposition process from time t until the next sampling time, when new measurements are received and the MPC problem of Equation 13.35 is solved for the computation of the next optimal input trajectory.

The dependence of the model coefficients, r_h, ν, σ^2, ρ^{ss}, and τ, on adsorption rate is used in the formulation of the model predictive controller of Equation 13.35. Thus, parameter estimation from open-loop kMC simulation results of the thin film growth process for a variety of operation conditions is performed to obtain the dependence of the model coefficients on adsorption rate using least-squares methods [21].

13.5 Simulation Results

In this section, the proposed model predictive controller of Equation 13.35 is applied to the multiscale model of the thin film growth process described in Section 13.2. The value of the adsorption rate is obtained from the solution of the problem of Equation 13.35 at each sampling time. The corresponding inlet concentration of silane is calculated from the adsorption rate based on the rate law of Equation 13.8 and is applied to the closed-loop system until the next sampling time. The optimization problem in the MPC formulation of Equation 13.35 is solved using a local constrained minimization algorithm using a broad set of initial guesses.

The desired values (set-point values) in the closed-loop simulations are $r_{set}^2 = 50$ layer^2 and $\rho_{set} = 0.985$, with a desired film thickness of $h_{min} = 800$ layers. The substrate temperature is fixed at 850 K. The variation of adsorption rate in the MPC formulation of Equation 13.35 is from 0.1 layer/s to 0.45 layer/s (0.45 layer/s is the maximum adsorption rate that can be obtained according to the rate law of Equation 13.8 at $X = 1$ and the given conditions of the gas phase in Table 13.1). The number of prediction

steps is set to be $p = 5$. The prediction horizon of each step is fixed at $\Delta = 5$ s. The closed-loop simulation duration is 3000 s. All expected values are obtained from 1000 independent simulation runs.

13.5.1 Regulation of Surface Roughness and Film Thickness

Closed-loop simulations of regulating film surface roughness and film thickness are first carried out. In these control problems, the control objective is to regulate the expected surface roughness square and expected film thickness to desired values. Thus, the cost functions of these problems contain penalties on the deviations of the expected surface roughness square from the set-point value and of the expected film thickness of the desire value. The weighting factors are $q_{r^2,i} = 0.1$, $q_{h,i} = 1$, and $q_{\rho,i} = 0$ for all i.

Figure 13.7 shows the closed-loop simulation results of the roughness–thickness control problem. From Figure 13.7, it can be seen that the model predictive controller drives the expected film thickness close to the desired value, at the end of the simulation. However, due to the requirement of achieving a desired film thickness value, which includes a higher penalty factor, the controller computes a higher adsorption rate, and thus, it results in a higher expected surface roughness square at the end of the closed-loop simulation. The effect of the penalty on film thickness can be observed by comparing Figure 13.7 with Figure 13.8, which shows the closed-loop simulation results without penalty on film thickness. It can

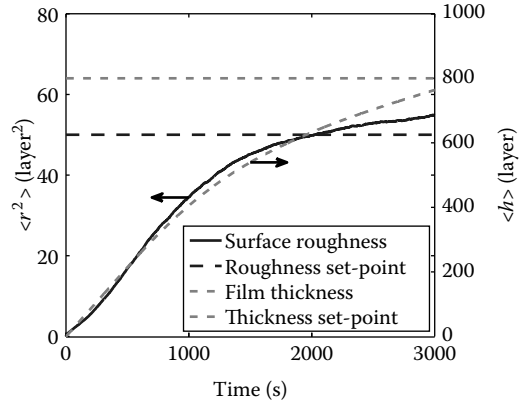

FIGURE 13.7 Profiles of the expected values of surface roughness square (solid line) and of the film thickness (dash-dotted line) under closed-loop operation; regulation of surface roughness and film thickness.

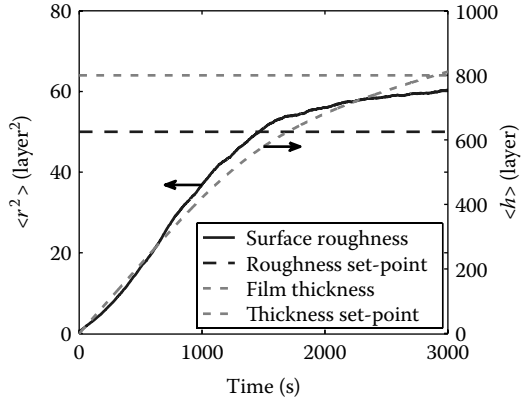

FIGURE 13.8 Profiles of the expected values of surface roughness square (solid line) and of the film thickness (dash-dotted line) under closed-loop operation; surface roughness-only control.

be clearly seen that, without penalty on the deviation of film thickness from its desired value, the expected surface roughness square approaches closer to the set-point value at the end of the simulation, while the expected film thickness is lower than the desired value.

13.5.2 Regulation of Film Porosity

In this subsection, it is demonstrated that the precise regulation of SOR to its set-point can be achieved. Figure 13.9 shows the closed-loop simulation results of the porosity control problem where the cost function includes only penalty on the deviation of film SOR from the desired value, 0.985. We conclude from these two figures that the model predictive controller successfully drives the expected film SOR to the set-point value.

13.5.3 Simultaneous Regulation of Surface Roughness, Film Porosity, and Film Thickness

Finally, closed-loop simulations of simultaneous regulation of film thickness, surface roughness, and film SOR are carried out with the same weighting factors. Since the inlet silane concentration is the

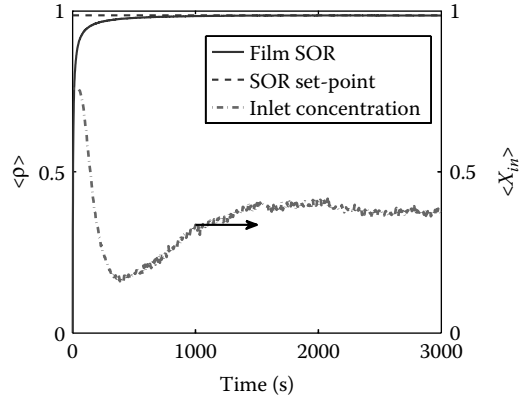

FIGURE 13.9 Profiles of the expected values of film SOR (solid line) and of the inlet silane concentration (dash-dotted line) under closed-loop operation; porosity-only control.

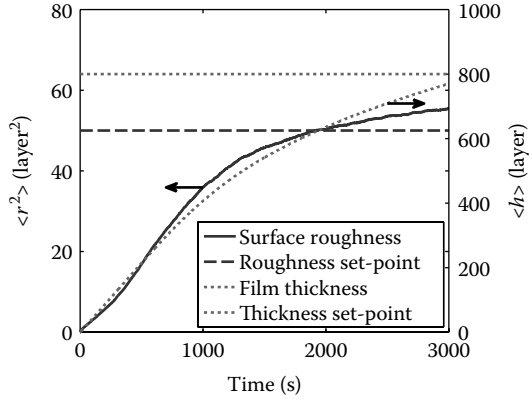

FIGURE 13.10 Profiles of the expected values of surface roughness square (solid line) and of the film thickness (dash-dotted line) under closed-loop operation; simultaneous regulation of film thickness, roughness, and porosity.

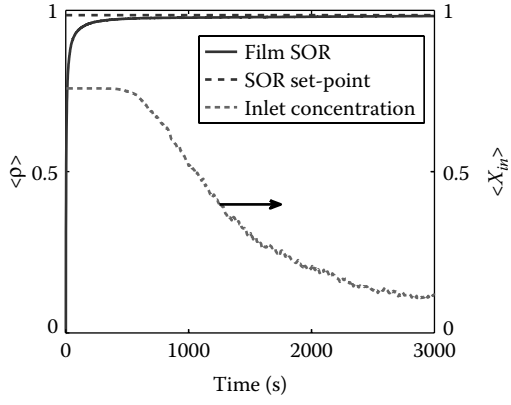

FIGURE 13.11 Profiles of the expected values of film SOR (solid line) and of the inlet silane concentration (dash-dotted line) under closed-loop operation; simultaneous regulation of film thickness, roughness, and porosity.

only manipulated input, the desired values of r_{set}^2 and ρ_{set} cannot be achieved simultaneously, that is, the corresponding inlet silane concentration needed to achieve the desired surface roughness and film thickness are not the same. Therefore, a trade-off between the two set-points is made by the controller. Figures 13.10 and 13.11 show the simulation results for this scenario. The expected values of both surface roughness square and film SOR approach their corresponding set-points with the desired film thickness being achieved.

13.6 Conclusions

Simultaneous regulation of film thickness, surface roughness, and porosity was developed in a multiscale model of a thin film growth process using the inlet precursor concentration as the manipulated input. Specifically, a continuous macroscopic PDE model was used to describe the dynamics of the gas phase. The thin film growth process was modeled via a microscopic kMC simulation model on a triangular lattice with vacancies and overhangs allowed to develop inside the film. Closed-form dynamic models of thin film surface profile and porosity were developed and used as the basis for the design of a MPC algorithm to simultaneously regulate film thickness, surface roughness, and film porosity. Simulation results demonstrated the applicability and effectiveness of the proposed modeling and control approach by applying the proposed controller to the multiscale model.

Acknowledgment

Financial support from NSF, CBET-0652131, is gratefully acknowledged.

References

1. P. D. Christofides, A. Armaou, Y. Lou, and A. Varshney. *Control and Optimization of Multiscale Process Systems*. Birkhäuser, Boston, 2008.
2. Y. Lou and P. D. Christofides. Estimation and control of surface roughness in thin film growth using kinetic Monte-Carlo models. *Chemical Engineering Science*, 58:3115–3129, 2003.
3. A. Varshney and A. Armaou. Multiscale optimization using hybrid PDE/kMC process systems with application to thin film growth. *Chemical Engineering Science*, 60:6780–6794, 2005.

4. R. Cuerno, H. A. Makse, S. Tomassone, S. T. Harrington, and H. E. Stanley. Stochastic model for surface erosion via ion sputtering: Dynamical evolution from ripple morphology to rough morphology. *Physical Review Letters*, 75:4464–4467, 1995.
5. S. F. Edwards and D. R. Wilkinson. The surface statistics of a granular aggregate. *Proceedings of the Royal Society of London Series A – Mathematical Physical and Engineering Sciences*, 381:17–31, 1982.
6. K. B. Lauritsen, R. Cuerno, and H. A. Makse. Noisy Kuramote–Sivashinsky equation for an erosion model. *Physical Review E*, 54:3577–3580, 1996.
7. J. Villain. Continuum models of crystal growth from atomic beams with and without desorption. *Journal de Physique I*, 1:19–42, 1991.
8. D. D. Vvedensky, A. Zangwill, C. N. Luse, and M. R. Wilby. Stochastic equations of motion for epitaxial growth. *Physical Review E*, 48:852–862, 1993.
9. G. Hu, Y. Lou, and P. D. Christofides. Dynamic output feedback covariance control of stochastic dissipative partial differential equations. *Chemical Engineering Science*, 63:4531–4542, 2008.
10. Y. Lou and P. D. Christofides. Feedback control of surface roughness using stochastic PDEs. *AIChE Journal*, 51:345–352, 2005.
11. D. Ni and P. D. Christofides. Multivariable predictive control of thin film deposition using a stochastic PDE model. *Industrial and Engineering Chemistry Research*, 44:2416–2427, 2005.
12. Y. Lou and P. D. Christofides. Nonlinear feedback control of surface roughness using a stochastic PDE: Design and application to a sputtering process. *Industrial and Engineering Chemistry Research*, 45:7177–7189, 2008.
13. Y. Lou, G. Hu, and P. D. Christofides. Model predictive control of nonlinear stochastic partial differential equations with application to a sputtering process. *AIChE Journal*, 54:2065–2081, 2008.
14. S. W. Levine and P. Clancy. A simple model for the growth of polycrystalline Si using the kinetic Monte Carlo simulation. *Modelling and Simulation in Materials Science and Engineering*, 8:751–762, 2000.
15. L. Wang and P. Clancy. A kinetic Monte Carlo study of the growth of Si on Si(100) at varying angles of incident deposition. *Surface Science*, 401:112–123, 1998.
16. L. Wang and P. Clancy. Kinetic Monte Carlo simulation of the growth of polycrystalline Cu films. *Surface Science*, 473:25–38, 2001.
17. P. Zhang, X. Zheng, S. Wu, J. Liu, and D. He. Kinetic Monte Carlo simulation of Cu thin film growth. *Vacuum*, 72:405–410, 2004.
18. G. Hu, G. Orkoulas, and P. D. Christofides. Modeling and control of film porosity in thin film deposition. *Chemical Engineering Science*, 64:3668–3682, 2009.
19. G. Hu, G. Orkoulas, and P. D. Christofides. Regulation of film thickness, surface roughness and porosity in thin film growth using deposition rate. *Chemical Engineering Science*, 64:3903–3913, 2009.
20. C. R. Kleijn, Th. H. van der Meer, and C. J. Hoogendoorn. A mathematical model for lpcvd in a single wafer reactor. *Journal of the Electrochemical Society*, 11:3423–3433, 1989.
21. G. Hu, G. Orkoulas, and P. D. Christofides. Stochastic modeling and simultaneous regulation of surface roughness and porosity in thin film deposition. *Industrial and Engineering Chemistry Research*, 48:6690–6700, 2009.
22. Y. G. Yang, R. A. Johnson, and H. N. Wadley. A Monte Carlo simulation of the physical vapor deposition of nickel. *Acta Materialia*, 45:1455–1468, 1997.
23. G. Hu, J. Huang, G. Orkoulas, and P. D. Christofides. Investigation of film surface roughness and porosity dependence on lattice size in a porous thin film deposition process. *Physical Review E*, 80:041122, 2009.

14

Control of Particulate Processes

Mingheng Li
California State Polytechnic University

Panagiotis D. Christofides
University of California, Los Angeles

14.1 Introduction

Particulate processes (also known as dispersed-phase processes) are characterized by the co-presence of and strong interaction between a continuous (gas or liquid) phase and a particulate (dispersed) phase and are essential in making many high-value industrial products. Particulate processes play a prominent role in a number of process industries, since about 60% of the products in the chemical industry are manufactured as particulates with an additional 20% using powders as ingredients. Representative examples of industrial particulate processes include the crystallization of proteins for pharmaceutical applications, the emulsion polymerization for the production of latex, the fluidized-bed production of solar-grade silicon particles through thermal decomposition of silane gas, the aerosol synthesis of titania powder used in the production of white pigments, and the thermal spray processing of functional thermal barrier and wear-resistant coatings. The industrial importance of particulate processes and the realization that the physico-chemical and mechanical properties of materials made with particulates depend heavily on the characteristics of the underlying particle-size distribution (PSD) have motivated significant research attention over the last 10 years on model-based control of particulate processes. These efforts have also been complemented by recent and ongoing developments in measurement technology, which allow the accurate and fast online measurement of key process variables including important characteristics of PSDs [1–3]. The recent efforts on model-based control of particulate processes have also been motivated by significant advances in the modeling of particulate processes. Specifically, population balances have provided a natural framework for the mathematical modeling of PSDs in broad classes of particulate processes (see, e.g., the tutorial article [4] and the review article [5]), and have been successfully used to describe PSDs in emulsion polymerization reactors [6,7], crystallizers [2,8], aerosol reactors, [9], and cell cultures [10]. Three representative examples will be studied to illustrate the structure of the mathematical models that arise in the population balance modeling of particulate processes: continuous crystallization, batch crystallization, and aerosol synthesis.

14.1.1 Continuous Crystallization

Crystallization is a particulate process, which is widely used in industry for the production of many products including fertilizers, proteins, and pesticides. A typical continuous crystallization process is shown in Figure 14.1. Under the assumptions of isothermal operation, constant volume, well-mixed suspension, nucleation of crystals of infinitesimal size, and mixed product removal, a dynamic model for the crystallizer can be derived from a population balance for the particle phase and a mass balance for the solute concentration, and has the following mathematical form [11,12]:

$$\begin{aligned}\frac{\partial n(r,t)}{\partial t} &= -\frac{\partial (R(t)n(r,t))}{\partial r} - \frac{n(r,t)}{\tau} + \delta(r-0)Q(t)\\ \frac{dc(t)}{dt} &= \frac{(c_0-\rho)}{\epsilon(t)\tau} + \frac{(\rho - c(t))}{\tau} + \frac{(\rho - c(t))}{\epsilon(t)}\frac{d\epsilon(t)}{dt}\end{aligned} \tag{14.1}$$

where $n(r,t)dr$ is the number of crystals in the size range of $[r, r+dr]$ at time t per unit volume of suspension, τ is the residence time, ρ is the density of the crystal, $c(t)$ is the solute concentration in the crystallizer, c_0 is the solute concentration in the feed, and

$$\epsilon(t) = 1 - \int_0^\infty n(r,t)\frac{4}{3}\pi r^3 dr$$

is the volume of liquid per unit volume of suspension. $R(t)$ is the crystal growth rate, $\delta(r-0)$ is the standard Dirac function, and $Q(t)$ is the crystal nucleation rate. The term $\delta(r-0)Q(t)$ accounts for the production of crystals of infinitesimal (zero) size via nucleation. An example of the expressions of $R(t)$ and $Q(t)$ is the following:

$$R(t) = k_1(c(t) - c_s), \quad Q(t) = \epsilon(t)k_2 e^{-\frac{k_3}{(c(t)/c_s - 1)^2}} \tag{14.2}$$

where k_1, k_2, and k_3 are constants and c_s is the concentration of solute at saturation. For a variety of operating conditions ([13] for model parameters and detailed studies), the continuous crystallizer model of Equation 14.1 exhibits highly oscillatory behavior (the main reason for this behavior is that the nucleation rate is much more sensitive to supersaturation relative to the growth rate, that is, [compare—the dependence of $R(t)$ and $Q(t)$ on the values of $c(t)$ and c_s]) which suggests the use of feedback control to ensure stable operation and attain a crystal size distribution (CSD) with desired characteristics. To

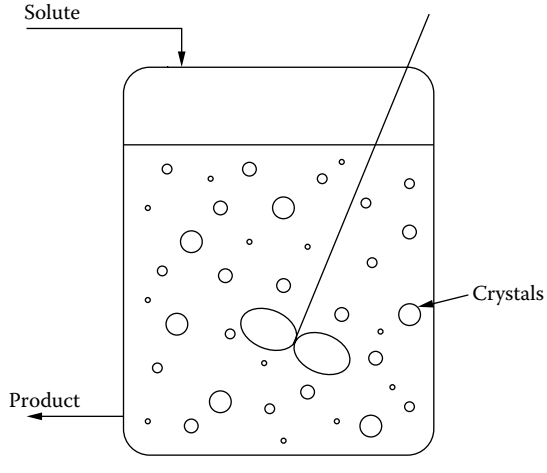

FIGURE 14.1 Schematic representation of a continuous crystallizer.

achieve this control objective, the inlet solute concentration can be used as the manipulated input and the crystal concentration as the controlled and measured output.

14.1.2 Batch Protein Crystallization

Batch crystallization plays an important role in the pharmaceutical industry. A batch crystallizer which is used to produce tetragonal HEW (hen-egg-white) lysozyme crystals from a supersaturated solution [14] is considered here. A schematic representation of the batch crystallizer is shown in Figure 14.2. Applying population, mass, and energy balances to the process, the following mathematical model is obtained:

$$\begin{aligned}
&\frac{\partial n(r,t)}{\partial t} + G(t)\frac{\partial n(r,t)}{\partial r} = 0, \quad n(0,t) = \frac{B(t)}{G(t)} \\
&\frac{dC(t)}{dt} = -24\rho k_v G(t)\mu_2(t) \\
&\frac{dT(t)}{dt} = -\frac{UA}{MC_p}(T(t) - T_j(t))
\end{aligned} \tag{14.3}$$

where $n(r,t)$ is the CSD, $B(t)$ the nucleation rate, $G(t)$ the growth rate, $C(t)$ the solute concentration, $T(t)$ the crystallizer temperature, $T_j(t)$ the jacket temperature, ρ the density of crystals, k_v the volumetric shape factor, U the overall heat-transfer coefficient, A the total heat-transfer surface area, M the mass of solvent in the crystallizer, C_p the heat capacity of the solution, and $\mu_2(t) = \int_0^\infty r^2 n(r,t)\,dr$ the second moment of the CSD. The nucleation rate, $B(t)$, and the growth rate, $G(t)$, are given by [14]:

$$B(t) = k_a C(t) \exp\left(-\frac{k_b}{\sigma^2(t)}\right), \quad G(t) = k_g \sigma^g(t) \tag{14.4}$$

where $\sigma(t)$, the supersaturation, is a dimensionless variable and is defined as $\sigma(t) = \ln(C(t)/C_s(T(t)))$, $C(t)$ is the solute concentration, g is the exponent relating growth rate to the supersaturation, and $C_s(T)$ is the saturation concentration of the solute which is a nonlinear function of the temperature of the form:

$$C_s(T) = 1.0036 \times 10^{-3} T^3 + 1.4059 \times 10^{-2} T^2 - 0.12835T + 3.4613 \tag{14.5}$$

The existing experimental results [15] show that the growth condition of the tetragonal HEW lysozyme crystal is significantly affected by supersaturation. Low supersaturation leads to the cessation of the crystal

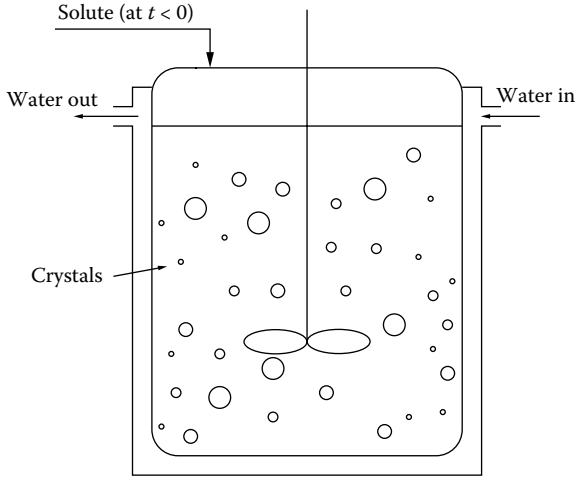

FIGURE 14.2 Schematic representation of a batch cooling crystallizer.

growth. On the other hand, rather than forming tetragonal crystals, a large amount of needle-like crystals are formed when the supersaturation is too high. Therefore, a proper range of supersaturation is necessary to guarantee the product's quality. The jacket temperature, T_j, is manipulated to achieve the desired crystal shape and size distribution.

14.1.3 Aerosol Synthesis

Aerosol processes are increasingly being used for the large-scale production of nanosized and micronsized particles. A typical aerosol flow reactor for the synthesis of titania aerosol with simultaneous chemical reaction, nucleation, condensation, coagulation, and convective transport is shown in Figure 14.3. A general mathematical model which describes the spatio-temporal evolution of the PSD in such aerosol processes can be obtained from a population balance and consists of the following nonlinear partial integro-differential equation [16,17]:

$$\frac{\partial n(v,z,t)}{\partial t} + v_z \frac{\partial n(v,z,t)}{\partial z} + \frac{\partial (G(\bar{x},v,z)n(v,z,t))}{\partial v} - I(v^*)\delta(v-v^*)$$
$$= \frac{1}{2}\int_0^v \beta(v-\bar{v},\bar{v},\bar{x})n(v-\bar{v},t)n(\bar{v},z,t)\,d\bar{v} - n(v,z,t)\int_0^\infty \beta(v,\bar{v},\bar{x})n(\bar{v},z,t)\,d\bar{v} \qquad (14.6)$$

where $n(v,z,t)$ denotes the PSD function, v is the particle volume, t is the time, $z \in [0,L]$ is the spatial coordinate, L is the length scale of the process, v^* is the size of the nucleated aerosol particles, v_z is the velocity of the fluid, $\bar{x}$ is the vector of the state variables of the continuous phase, $G(\cdot,\cdot,\cdot), I(\cdot), \beta(\cdot,\cdot,\cdot)$ are nonlinear scalar functions, which represent the growth, nucleation and coagulation rates, and $\delta(\cdot)$ is the standard Dirac function. The model of Equation 14.6 is coupled with a mathematical model which describes the spatio-temporal evolution of the concentration of species and temperature of the gas phase ($\bar{x}$) that can be obtained from the mass and energy balances. The control problem is to regulate process variables like inlet flow rates and wall temperature to produce aerosol products with desired size distribution characteristics.

The mathematical models of Equations 14.1, 14.3, and 14.6 demonstrate that particulate process models are nonlinear and distributed parameter in nature. These properties have motivated extensive research on the development of efficient numerical methods for the accurate computation of their solution see, for example, [5,9,10,18–21]. However, in spite of the rich literature on population balance modeling, numerical solution, and dynamical analysis of particulate processes, till about 10 years ago, research on model-based control of particulate processes had been very limited. Specifically, early research efforts had mainly focused on the understanding of fundamental control-theoretic properties (controllability and observability) of population balance models [22] and the application of conventional control schemes

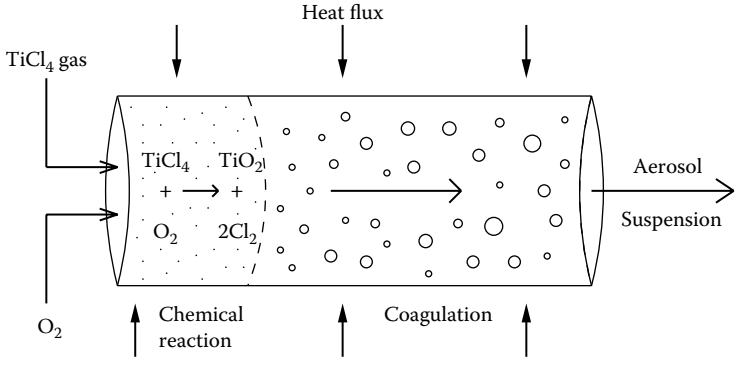

FIGURE 14.3 Schematic representation of a titania aerosol reactor.

(such as proportional-integral and proportional-integral-derivative control, self-tuning control) to crystallizers and emulsion polymerization processes [6,23,24]. The main difficulty in synthesizing nonlinear model-based feedback controllers for particulate processes is the distributed parameter nature of the population balance models, which does not allow their direct use for the synthesis of low-order (and therefore, practically implementable) model-based feedback controllers. Furthermore, a direct application of the aforementioned solution methods to particulate process models leads to finite-dimensional approximations of the population balance models (i.e., nonlinear ordinary differential equation (ODE) systems in time) which are of very high order and thus inappropriate for the synthesis of model-based feedback controllers that can be implemented in real time. This limitation had been the bottleneck for model-based synthesis and real-time implementation of model-based feedback controllers on particulate processes.

14.2 Model-Based Control of Particulate Processes

14.2.1 Overview

Motivated by the lack of population balance-based control methods for particulate processes and the need to achieve tight size distribution control in many particulate processes, over the last 10 years, a general framework for the synthesis of nonlinear, robust, and predictive controllers for particulate processes based on population balance models [13,14,16,25–30] has been developed. Specifically, within the developed framework, nonlinear low-order approximations of the particulate process models are initially derived using order reduction techniques and are used for controller synthesis. Subsequently, the infinite-dimensional closed-loop system stability, performance, and robustness properties were precisely characterized in terms of the accuracy of the approximation of the low-order models. Furthermore, controller designs were proposed that deal directly with the key practical issues of uncertainty in model parameters, unmodeled actuator/sensor dynamics and constraints in the capacity of control actuators and the magnitude of the process state variables. It is also important to note that owing to the low-dimensional structure of the controllers, the computation of the control action involves the solution of a small set of ODEs, and thus, the developed controllers can be readily implemented in real-time with reasonable computing power, thereby resolving the main issue on model-based control of particulate processes. In addition to theoretical developments, the application of the proposed methods to size distribution control in continuous and batch crystallization, aerosol, and thermal spray processes has also been successfully demonstrated and their effectiveness and advantages with respect to conventional control methods has been docmented. Figure 14.4 summarizes these efforts. The reader may refer to [8–10] for reviews of results on simulation and control of particulate processes. The remainder of this chapter is based on results published in [13,14,16,17,25,26,28–32].

14.2.2 Particulate Process Model

To present the main elements of the approach to model-based control of particulate processes, a general class of spatially homogeneous particulate processes with simultaneous particle growth, nucleation, agglomeration, and breakage will be considered. Examples of such processes have been introduced in the previous section. Assuming that particle size is the only internal particle coordinate and applying a dynamic material balance on the number of particles of size r to $r + dr$ (population balance), one can obtain the following general nonlinear partial integro-differential equation which describes the rate of change of the PSD, $n(r, t)$:

$$\frac{\partial n}{\partial t} = -\frac{\partial (G(x, r)n)}{\partial r} + w(n, x, r) \tag{14.7}$$

where $n(r, t)$ is the particle number size distribution, $r \in [0, r_{max}]$ is the particle size, and r_{max} is the maximum particle size (which may be infinity), t is the time, and $x \in \mathbb{R}^n$ is the vector of state variables, which describe properties of the continuous phase (e.g., solute concentration, temperature, and pH in

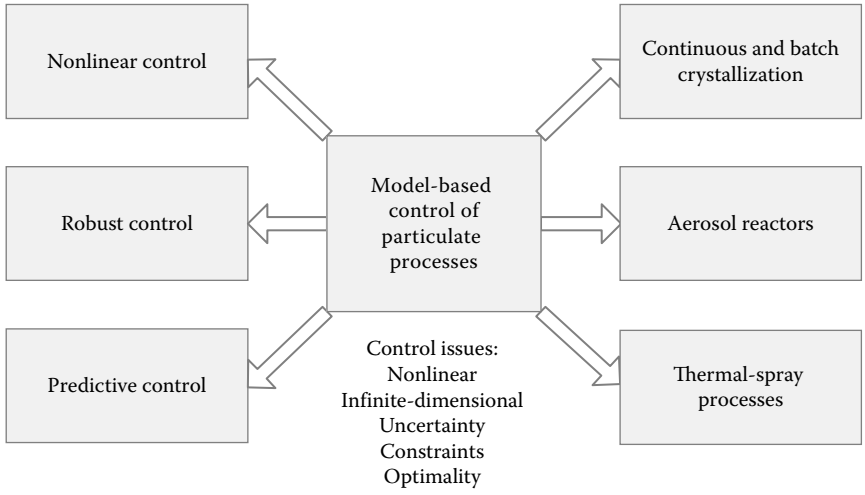

FIGURE 14.4 Summary of research on model-based control of particulate processes.

a crystallizer); see Equation 14.8 for the system that describes the dynamics of x. $G(x,r)$ and $w(n,x,r)$ are nonlinear scalar functions whose physical meaning can be explained as follows: $G(x,r)$ accounts for particle growth through condensation and is usually referred to as growth rate. It usually depends on the concentrations of the various species present in the continuous phase, the temperature of the process, and the particle size. On the other hand, $w(n,x,r)$ represents the net rate of introduction of new particles into the system. It includes all the means by which particles appear or disappear within the system including particle agglomeration (merging of two particles into one), breakage (division of one particle to two), as well as nucleation of particles of size $r \geq 0$ and particle feed and removal. The rate of change of the continuous-phase variables x can be derived by a direct application of mass and energy balances to the continuous phase and is given by a nonlinear integro-differential equation system of the general form:

$$\dot{x} = f(x) + g(x)u(t) + A\int_0^{r_{max}} a(n,r,x)\,dr \tag{14.8}$$

where $f(x)$ and $a(n,r,x)$ are nonlinear vector functions, $g(x)$ is a nonlinear matrix function, A is a constant matrix, and $u(t) = [u_1\ u_2 \ldots u_m] \in \mathbb{R}^m$ is the vector of manipulated inputs. The term $A\int_0^{r_{max}} a(n,r,x)dr$ accounts for mass and heat transfer from the continuous phase to all the particles in the population.

14.2.3 Model Reduction of Particulate Process Models

While the population balance models are infinite-dimensional systems, the dominant dynamic behavior of many particulate process models has been shown to be low-dimensional. Manifestations of this fundamental property include the occurrence of oscillatory behavior in continuous crystallizers [11] and the ability to capture the long-term behavior of aerosol systems with self-similar solutions [9]. Motivated by this, a general methodology for deriving low-order ODE systems, was introduced [13] which accurately reproduces the dominant dynamics of the nonlinear integro-differential equation system of Equations 14.7 and 14.8. The proposed model reduction methodology exploits the low-dimensional behavior of the dominant dynamics of the system of Equations 14.7 and 14.8 and is based on a combination of the method of weighted residuals with the concept of approximate inertial manifold.

Specifically, the proposed approach initially employs the method of weighted residuals (see [5] for a comprehensive review of results on the use of this method for solving population balance equations) to construct a nonlinear, possibly high-order, ODE system that accurately reproduces the solutions and

dynamics of the distributed parameter system of Equations 14.7 and 14.8. Specifically, one can first consider an orthogonal set of basis functions $\phi_k(r)$, where $r \in [0, r_{max})$, $k = 1, \ldots, \infty$, and expand the PSD function $n(r,t)$ in an infinite series in terms of $\phi_k(r)$ as follows:

$$n(r,t) = \sum_{k=1}^{\infty} a_k(t)\phi_k(r) \tag{14.9}$$

where $a_k(t)$ are time-varying coefficients. In order to approximate the system of Equations 14.7 and 14.8 with a finite set of ODEs, one can obtain a set of N equations by substituting Equation 14.9 into Equations 14.7 and 14.8, multiplying the population balance with N different weighting functions $\psi_\nu(r)$ (i.e. $\nu = 1, \ldots, N$), and integrating over the entire particle size spectrum. In order to obtain a finite-dimensional model, the series expansion of $n(r,t)$ is truncated up to order N. The infinite-dimensional system of Equation 14.7 reduces to the following finite set of ODEs:

$$\begin{aligned}\int_0^{r_{max}} \psi_\nu(r) \sum_{k=1}^{N} \phi_k(r) \frac{\partial a_{kN}(t)}{\partial t} dr = &-\sum_{k=1}^{N} a_{kN}(t) \int_0^{r_{max}} \psi_\nu(r) \frac{\partial (G(x_N, r)\phi_k(r))}{\partial r} dr \\ &+ \int_0^{r_{max}} \psi_\nu(r) w\left(\sum_{k=1}^{N} a_{kN}(t)\phi_k(r), x_N, r\right) dr, \quad \nu = 1, \ldots, N \\ \dot{x}_N = &f(x_N) + g(x_N)u(t) + A\int_0^{r_{max}} a\left(\sum_{k=1}^{N} a_{kN}(t)\phi_k(r), r, x_N\right) dr\end{aligned} \tag{14.10}$$

where x_N and a_{kN} are the approximations of x and a_k obtained by an Nth order truncation. From Equation 14.10, it is clear that the form of the ODEs that describe the rate of change of $a_{kN}(t)$ depends on the choice of the basis and weighting functions, as well as on N. The system of Equation 14.10 was obtained from a direct application of the method of weighted residuals (with arbitrary basis functions) to the system of Equations 14.7 and 14.8, and thus, may be of very high order in order to provide an accurate description of the dominant dynamics of the particulate process model. High-dimensionality of the system of Equation 14.10 leads to a complex controller design and high-order controllers, which cannot be readily implemented in practice. To circumvent these problems, the low-dimensional behavior of the dominant dynamics of particulate processes was exploited [13] and an approach based on the concept of inertial manifold to derive low-order ODE systems that accurately describe the dominant dynamics of the system of Equation 14.10 was proposed. This order reduction technique initially employs singular perturbation techniques to construct nonlinear approximations of the modes neglected in the derivation of the finite-dimensional model of Equation 14.10 (i.e., modes of order $N+1$ and higher) in terms of the first N modes. Subsequently, these steady-state expressions for the modes of order $N+1$ and higher (truncated up to appropriate order) are used in the model of Equation 14.10 (instead of setting them to zero) and significantly allow improving the accuracy of the model of Equation 14.10 without increasing its dimension; details on this procedure can be found in [13].

Referring to the method of weighted residuals, it is important to note that the basis and weighting functions determine the type of weighted residual method being used. In particular, the method of weighted residuals reduces to the method of moments when the basis functions are chosen to be Laguerre polynomials and the weighting functions are chosen as $\psi_\nu = r^\nu$. The moments of the PSD are defined as:

$$\mu_\nu = \int_0^{\infty} r^\nu n(r,t) dr, \quad \nu = 0, \ldots, \infty \tag{14.11}$$

and the moment equations can be directly generated from the population balance model by multiplying it by r^ν, $\nu = 0, \ldots, \infty$ and integrating from 0 to ∞. The procedure of forming moments of the population balance equation very often leads to terms that may not reduce to moments, terms that include fractional

moments, or to an unclosed set of moment equations. To overcome this problem, the PSD may be expanded in terms of Laguerre polynomials defined in $L_2[0,\infty)$ and the series solution using a finite number of terms may be used to close the set of moment equations (this procedure has been successfully used for models of crystallizers with fines trap [25]).

14.2.4 Model-Based Control Using Low-Order Models

14.2.4.1 Nonlinear Control

Low-order models can be constructed using the techniques described in the previous section. Specifically, based on the method of moments, the following infinite-order dimensionless system can be derived from Equation 14.1 for the continuous crystallization process [13]:

$$\begin{aligned}
\frac{d\tilde{x}_0}{dt} &= -\tilde{x}_0 + (1-\tilde{x}_3)Da e^{-F/\tilde{y}^2} \\
\frac{d\tilde{x}_1}{dt} &= -\tilde{x}_1 + \tilde{y}\tilde{x}_0 \\
\frac{d\tilde{x}_2}{dt} &= -\tilde{x}_2 + \tilde{y}\tilde{x}_1 \\
\frac{d\tilde{x}_3}{dt} &= -\tilde{x}_3 + \tilde{y}\tilde{x}_2 \\
\frac{d\tilde{x}_\nu}{dt} &= -\tilde{x}_\nu + \tilde{y}\tilde{x}_{\nu-1}, \quad \nu = 4, \ldots \\
\frac{d\tilde{y}}{dt} &= \frac{1-\tilde{y}-(\alpha-\tilde{y})\tilde{y}\tilde{x}_2}{1-\tilde{x}_3}
\end{aligned} \tag{14.12}$$

where $\tilde{x}_i$ and $\tilde{y}$ are the dimensionless ith moment and solute concentration, respectively, and Da and F are dimensionless parameters [13]. On the basis of the system of Equation 14.12, it is clear that the moments of order four and higher do not affect those of order three and lower, and moreover, the state of the infinite-dimensional system:

$$\frac{d\tilde{x}_\nu}{dt} = -\tilde{x}_\nu + \tilde{y}\tilde{x}_{\nu-1}, \quad \nu = 4, \ldots \tag{14.13}$$

is bounded when x_3 and y are bounded, and it converges to a globally exponentially stable equilibrium point when $\lim_{t\to\infty} x_3 = c_1$ and $\lim_{t\to\infty} \tilde{y} = c_2$, where c_1, c_2 are constants. This implies that the dominant dynamics (i.e., dynamics associated with eigenvalues that are close to the imaginary axis) of the process of Equation 14.1 can be adequately captured by the following fifth-order moment model:

$$\begin{aligned}
\frac{d\tilde{x}_0}{dt} &= -\tilde{x}_0 + (1-\tilde{x}_3)Da e^{-F/\tilde{y}^2} \\
\frac{d\tilde{x}_1}{dt} &= -\tilde{x}_1 + \tilde{y}\tilde{x}_0 \\
\frac{d\tilde{x}_2}{dt} &= -\tilde{x}_2 + \tilde{y}\tilde{x}_1 \\
\frac{d\tilde{x}_3}{dt} &= -\tilde{x}_3 + \tilde{y}\tilde{x}_2 \\
\frac{d\tilde{y}}{dt} &= \frac{1-\tilde{y}-(\alpha-\tilde{y})\tilde{y}\tilde{x}_2}{1-\tilde{x}_3}
\end{aligned} \tag{14.14}$$

The ability of the above fifth-order moment model to reproduce the dynamics, and to some extent the solutions, of the distributed parameter model of Equation 14.1 is shown in Figure 14.5, where the profiles of the total particle concentration generated by the two models are compared (both models start from the same initial conditions). Even though the discrepancy of the total particle concentration profiles predicted

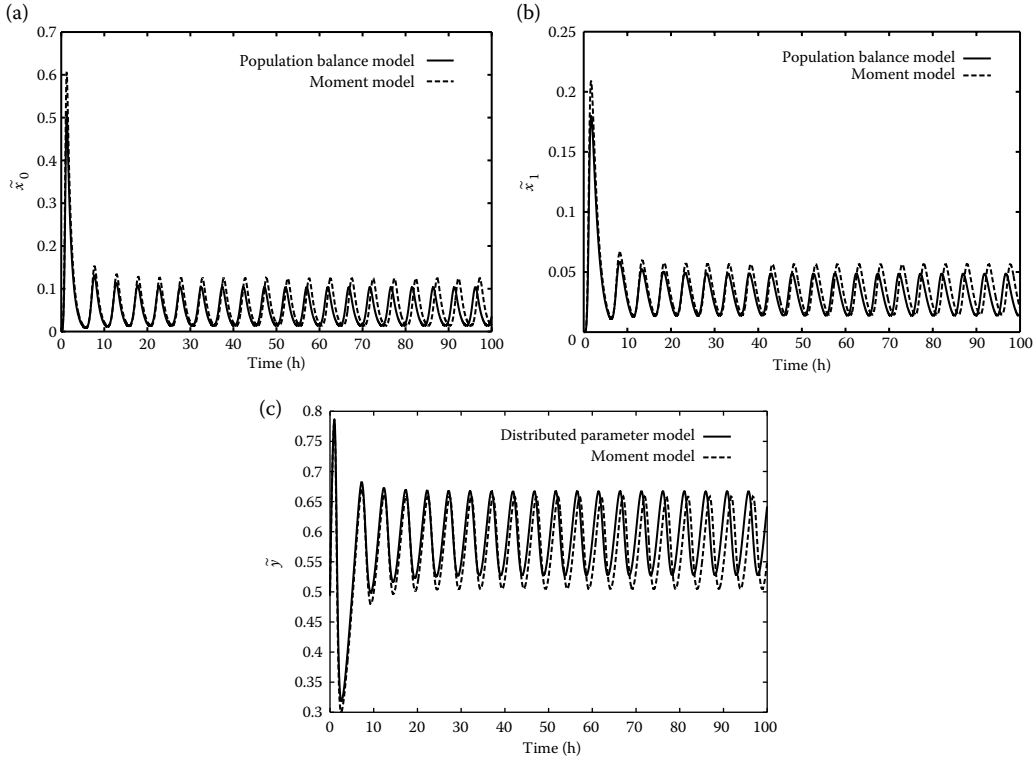

FIGURE 14.5 Comparison of open-loop profiles of (a) crystal concentration, (b) total crystal size, and (c) solute concentration obtained from the distributed parameter model and the moment model.

by the two models increases with time (this is expected due to the open-loop instability of the process), it is clear that the fifth-order moment model of Equation 14.14 provides a very good approximation of the distributed parameter model of Equation 14.1, thereby establishing that the dominant dynamics of the system of Equation 14.1 are low-dimensional and motivating the use of the moment model for nonlinear controller design.

For the batch crystallization process, the following low-order model can be derived from Equation 14.3 using the method of moments:

$$
\begin{aligned}
\frac{d\mu_0}{dt} &= \left(1 - \frac{4}{3}\pi\mu_3\right) k_2 e^{-\frac{k_3}{(c/c_s-1)^2}} e^{-\frac{E_b}{RT}} \\
\frac{d\mu_1}{dt} &= k_1(c - c_s)e^{-\frac{E_g}{RT}}\mu_0 \\
\frac{d\mu_2}{dt} &= 2k_1(c - c_s)e^{-\frac{E_g}{RT}}\mu_1 \\
\frac{d\mu_3}{dt} &= 3k_1(c - c_s)e^{-\frac{E_g}{RT}}\mu_2 \\
\frac{dc}{dt} &= \frac{-4\pi(c - c_s)\mu_2(\rho - c)}{\left(1 - \frac{4}{3}\pi\mu_3\right)} \\
\frac{dT}{dt} &= -\frac{\rho_c \Delta H_c}{\rho C_p} 4\pi k_1(c - c_s)e^{-\frac{E_g}{RT}}\mu_2 - \frac{UA_c}{\rho C_p V}(T - T_c)
\end{aligned}
\tag{14.15}
$$

Based on the low-order models, the nonlinear finite-dimensional state and output feedback controllers have been synthesized that guarantee stability and enforce output tracking in the closed-loop finite-dimensional system. It is also established that these controllers exponentially stabilize the closed-loop particulate process model. The output feedback controller is constructed through a standard combination of the state feedback controller with a state observer. Specifically, in the case of the continuous crystallization example, the nonlinear output feedback controller has the following form:

$$
\begin{aligned}
\frac{d\omega_0}{dt} &= -\omega_0 + (1-\omega_3)Dae^{-F/\omega_4^2} + L_0(\tilde{h}(\tilde{x}) - \tilde{h}(\omega)) \\
\frac{d\omega_1}{dt} &= -\omega_1 + \omega_4\omega_0 + L_1(\tilde{h}(\tilde{x}) - \tilde{h}(\omega)) \\
\frac{d\omega_2}{dt} &= -\omega_2 + \omega_4\omega_1 + L_2(\tilde{h}(\tilde{x}) - \tilde{h}(\omega)) \\
\frac{d\omega_3}{dt} &= -\omega_3 + \omega_4\omega_2 + L_3(\tilde{h}(\tilde{x}) - \tilde{h}(\omega)) \\
\frac{d\omega_4}{dt} &= \frac{1-\omega_4-(\alpha-\omega_4)\omega_4\omega_2}{1-\omega_3} + L_4(\tilde{h}(\tilde{x}) - \tilde{h}(\omega)) \\
&\quad + \frac{[\beta_2 L_{\tilde{g}} L_{\tilde{f}}\tilde{h}(\omega)]^{-1}\left\{\nu - \beta_0\tilde{h}(\omega) - \beta_1 L_{\tilde{f}}\tilde{h}(\omega) - \beta_2 L_{\tilde{f}}^2\tilde{h}(\omega)\right\}}{1-\omega_3} \\
\bar{u}(t) &= [\beta_2 L_{\tilde{g}} L_{\tilde{f}}\tilde{h}(\omega)]^{-1}\left\{\nu - \beta_0\tilde{h}(\omega) - \beta_1 L_{\tilde{f}}\tilde{h}(\omega) - \beta_2 L_{\tilde{f}}^2\tilde{h}(\omega)\right\}
\end{aligned}
\tag{14.16}
$$

where ν is the set-point, $\beta_0, \beta_1, \beta_2$, and $L = [L_0\ L_1\ L_2\ L_3\ L_4]^T$ are controller parameters and $\tilde{h}(\omega) = \omega_0$ or $\tilde{h}(\omega) = \omega_1$.

The nonlinear controller of Equation 14.16 was also combined with a Proportional-integral (PI) controller (i.e., the term $\nu - \beta_0\tilde{h}(\omega)$ was substituted by $\nu - \beta_0\tilde{h}(\tilde{x}) + 1/\tau_i'\xi$, where $\dot{\xi} = \nu - \tilde{h}(\tilde{x})$, $\xi(0) = 0$, and τ_i' is the integral time constant) to ensure offsetless tracking in the presence of constant uncertainty in process parameters. The practical implementation of the nonlinear controller of Equation 14.16 requires online measurements of the controlled outputs $\tilde{x}_0$ or $\tilde{x}_1$; in practice, such measurements can be obtained by using, for example, light scattering [2,33]. In Equation 14.16, the state feedback controller is synthesized via geometric control methods and the state observer is an extended Luenberger-type observer [13].

Several simulations have been performed in the context of the continuous crystallizer process model presented before to evaluate the performance and robustness properties of the nonlinear controllers designed based on the reduced order models, and compared them with the ones of a PI controller. In all the simulation runs, the initial condition:

$$n(r,0) = 0.0, \quad c(0) = 990.0\ \text{kg/m}^3$$

was used for the process model of Equations 14.1 and 14.2 and the finite-difference method with 1000 discretization points was used for its simulation. The crystal concentration, $\tilde{x}_0$ was considered to be the controlled output and the inlet solute concentration was chosen to be the manipulated input. Initially, the set-point tracking capability of the nonlinear controller was evaluated under nominal conditions for a 0.5 increase in the value of the set-point.

Figure 14.6 shows the closed-loop output (left plot) and manipulated input (right plot) profiles obtained by using the nonlinear controller (solid lines). For the sake of comparison, the corresponding profiles under PI control are also included (dashed lines); the PI controller was tuned so that the closed-loop output response exhibits the same level of overshoot to the one of the closed-loop outputs under nonlinear control. Clearly, the nonlinear controller drives the controlled output to its new set-point value in significantly shorter time than the one required by the PI controller, while both controlled outputs exhibit very similar

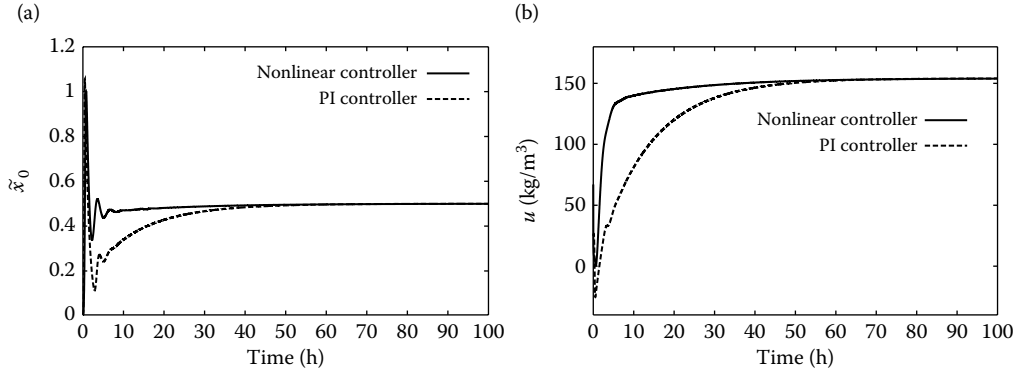

FIGURE 14.6 (a) Closed-loop output and (b) manipulated input profiles under nonlinear and PI control, for a 0.5 increase in the set-point ($\tilde{x}_0$ is the controlled output).

overshoots. For the same simulation run, the evolution of the closed-loop profile and the final steady-state profile of the CSD are shown in Figure 14.7. An exponentially decaying CSD is obtained at the steady state. The reader may refer to [13] for extensive simulation results.

14.2.4.2 Hybrid Predictive Control

In addition to handling nonlinear behavior, an important control problem is to stabilize the crystallizer at an unstable steady state (which corresponds to a desired PSD) using constrained control action. Currently, the achievement of high performance, under control and state constraints, relies to a large extent on the use of model predictive control (MPC) policies. In this approach, a model of the process is used to make predictions of the future process evolution and compute control actions, through repeated solution of constrained optimization problems, which ensure that the process state variables satisfy the imposed limitations. However, the ability of the available MPCs to guarantee closed-loop stability and enforce constraint satisfaction is dependent on the assumption of feasibility (i.e., existence of a solution) of the constrained optimization problem. This limitation strongly impacts the practical implementation of the MPC policies and limits the *a priori* (i.e., before controller implementation) characterization of the set of initial conditions starting from where the constrained optimization problem is feasible and closed-loop stability is guaranteed. This problem typically results in the need for extensive closed-loop

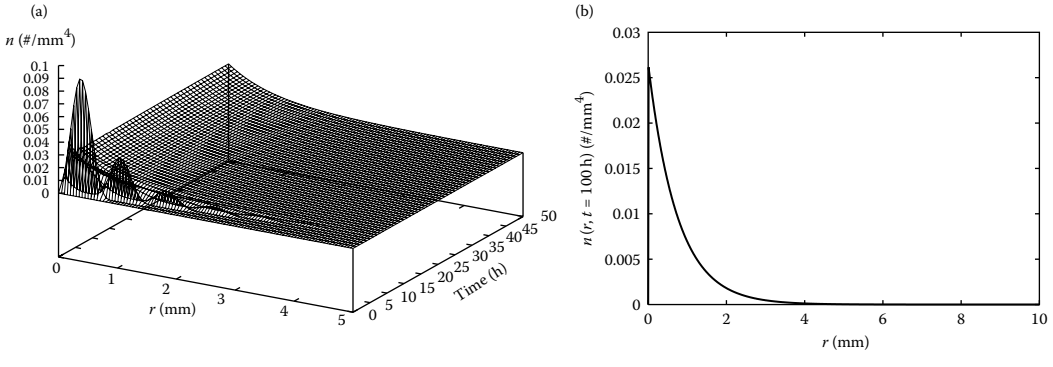

FIGURE 14.7 Profile of evolution of CSD (a) and final steady-state CSD (b) under nonlinear control ($\tilde{x}_0$ is the controlled output).

simulations and software verification (before online implementation) to search over the whole set of possible initial operating conditions that guarantee stability. This in turn can lead to prolonged periods for plant commissioning. Alternatively, the lack of *a priori* knowledge of the stabilizing initial conditions may necessitate a limiting process operation within a small conservative neighborhood of the desired set point in order to avoid extensive testing and simulations. Given the tight product quality specifications, however, both of the remedies can impact negatively on the efficiency and profitability of the process by limiting its operational flexibility. Lyapunov-based analytical control designs allow for an explicit characterization of the constrained stability region [34–36], however, their closed-loop performance properties cannot be transparently characterized.

To overcome these difficulties, a hybrid predictive control structure that provides a safety net for the implementation of predictive control algorithms has recently been developed [37]. The central idea is to embed the implementation of MPC within the stability region of a bounded controller and devise a set of switching rules that orchestrate the transition from MPC to the bounded controller in the event that MPC is unable to achieve closed-loop stability (e.g., due to inappropriate choice of the horizon length, infeasibility, or computational difficulties). Switching between the two controllers allows reconciling the tasks of optimal stabilization of the constrained closed-loop system (through MPC) with that of computing *a priori* the set of initial conditions for which closed-loop stability is guaranteed (through Lyapunov-based [34,35] bounded nonlinear control).

The application of the hybrid predictive control strategy to the continuous crystallizer of Equations 14.1 and 14.2 was demonstrated. The control objective was to suppress the oscillatory behavior of the crystallizer and stabilize it at an unstable steady state that corresponds to a desired PSD by manipulating the inlet solute concentration. To achieve this objective, measurements or estimates of the first four moments and of the solute concentration are assumed to be available. Subsequently, the proposed methodology was employed for the design of the controllers using a low-order model constructed by using the method of moments. A comparison was made between the hybrid predictive control scheme, with an MPC controller designed with a set of stabilizing constraints and a Lyapunov-based nonlinear controller.

In the first set of simulation runs, the ability of the MPC controller with the stability constraints to stabilize the crystallizer starting from the initial condition, $x(0) = [0.066\ \ 0.041\ \ 0.025\ \ 0.015\ \ 0.560]'$ was tested. The result is shown by the solid lines in Figure 14.8a–e where it is seen that the predictive controller, with a horizon length of $T = 0.25$, is able to stabilize the closed-loop system at the desired equilibrium point. Starting from the initial condition $x(0) = [0.033\ \ 0.020\ \ 0.013\ \ 0.0075\ \ 0.570]'$, however, the MPC controller with the stability constraints yields no feasible solution. If the stability constraints are relaxed to make the MPC feasible, one can see from the dashed lines in Figure 14.8a–e that the resulting control action cannot stabilize the closed-loop system, and leads to a stable limit cycle. On the other hand, the bounded controller is able to stabilize the system from both initial conditions (this was guaranteed because both initial conditions lied inside the stability region of the controller). The state trajectory starting from $x(0) = [0.033\ \ 0.020\ \ 0.013\ \ 0.0075\ \ 0.570]'$ is shown in Figure 14.8a–e with the dotted profile. This trajectory, although stable, presents slow convergence to the equilibrium as well as a damped oscillatory behavior that the MPC does not show when it is able to stabilize the system.

When the hybrid predictive controller is implemented from the initial condition $x(0) = [0.033\ \ 0.020\ \ 0.013\ \ 0.0075\ \ 0.570]'$, the supervisor detects initial infeasibility of MPC and implements the bounded controller in the closed loop. As the closed-loop states evolve under the bounded controller and get closer to the desired steady-state, the supervisor finds (at $t = 5.8$ h) that the MPC becomes feasible and, therefore, implements it for all future times. Note that despite the "jump" in the control action profile as one switches from the bounded controller to MPC at $t = 5.8$ h (see the difference between dotted and dash-dotted profiles in Figure 14.8f), the moments of the PSD in the crystallizer continue to evolve smoothly (dash-dotted lines in Figure 14.8a–e). The supervisor finds that MPC continues to be feasible and is implemented in the closed-loop to stabilize the closed-loop system at the desired steady state.

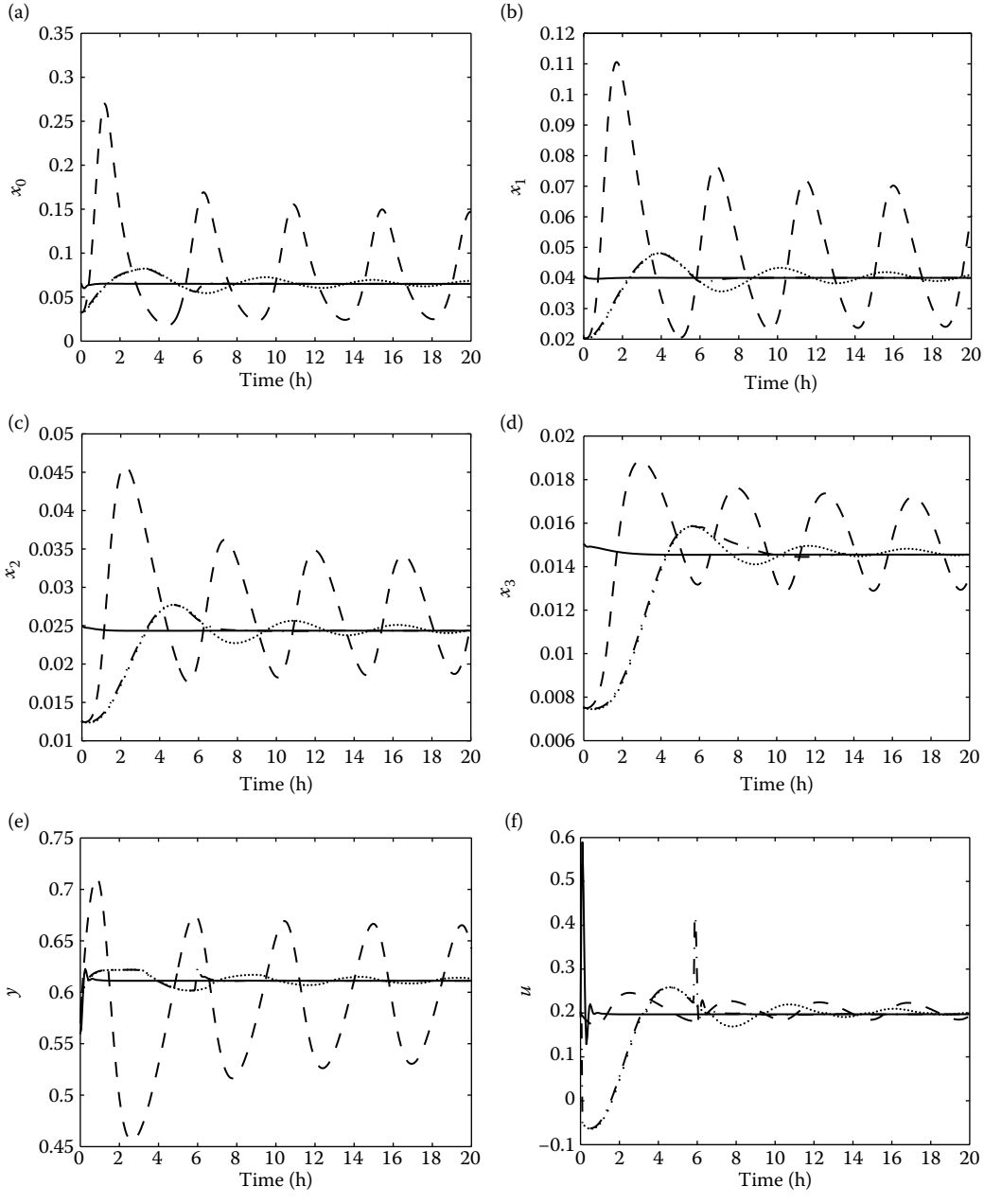

FIGURE 14.8 Continuous crystallizer example: closed-loop profiles of the dimensionless crystallizer moments (a)–(d), the solute concentration in the crystallizer (e), and the manipulated input (f) under MPC with stability constraints (solid lines), under MPC without stability constraints (dashed lines), under the bounded controller (dotted lines), and using the hybrid predictive controller (dash-dotted lines). Note the different initial states.

Compared with the simulation results under the bounded controller, the hybrid predictive controller (dash-dotted lines) stabilizes the system much faster, and achieves a better performance, reflected in a lower value of the performance index (0.1282 vs 0.1308). The manipulated input profiles for the three scenarios are shown in Figure 14.8f.

14.2.4.3 Predictive Control of Size Distribution in a Batch Protein Crystallizer

In batch crystallization, the main objective is to achieve a desired PSD at the end of the batch and satisfying state and control constraints during the whole batch run. Significant previous work has focused on CSD control in batch crystallizers [2,38]. In [39], a method was developed for assessing parameter uncertainty and studied its effects on the open-loop optimal control strategy, which maximized the weight mean size of the product. To improve the product quality expressed in terms of the mean size and the width of the distribution, an on-line optimal control methodology was developed for a seeded batch cooling crystallizer [40]. In these previous works, most efforts were focused on the open-loop optimal control of the batch crystallizer, that is, the optimal operating condition was calculated off-line based on mathematical models. The successful application of such a control strategy relies, to a large extent, on the accuracy of the models. Furthermore, an open-loop control strategy may not be able to manipulate the system to follow the optimal trajectory because of the ubiquitous existence of modeling error. Motivated by this, a predictive feedback control system to maximize the volume-averaged tetragonal lysozyme crystal size (i.e., μ_4/μ_3 where μ_3, μ_4 are the third and fourth moments of the CSD; see Equation 14.11) by manipulating the jacket temperature, T_j was developed [30]. The principle moments are calculated from the on-line measured CSD, n, which can be obtained by measurement techniques such as the laser light scattering method. The concentration and crystallizer temperature are also assumed to be measured in real time. In the closed-loop control structure, a reduced-order moments model was used within the predictive controller for the purpose of prediction. The main idea is to use this model to obtain a prediction of the state of the process at the end of the batch operation, t_f, from the current measurement at time t. Using this prediction, a cost function that depends on this value is minimized subject to a set of operating constraints. Manipulated input limitations and constraints on supersaturation and crystallizer temperature are incorporated as input and state constraints on the optimization problem. The optimization algorithm computes the profile of the manipulated input T_j from the current time until the end of the batch operation interval and then the current value of the computed input is implemented on the process, and the optimization problem is resolved and the input is updated each time a new measurement is available (receding horizon control strategy). The optimization problem that is solved at each sampling instant takes the following form:

$$
\begin{aligned}
&\min_{T_j} -\frac{\mu_4(t_f)}{\mu_3(t_f)} \\
&s.t. \frac{d\mu_0}{dt} = k_a C \ \exp\left(-\frac{k_b}{\sigma^2}\right) \\
&\frac{d\mu_i}{dt} = i k_g \sigma^g \mu_{i-1}(t), \quad i = 1, \ldots, 4 \\
&\frac{dC}{dt} = -24 \rho k_v k_g \sigma^g \mu_2(t) \\
&\frac{dT}{dt} = -\frac{UA}{MC_p}(T - T_j) \\
&T_{min} \le T \le T_{max} \\
&T_{j\,min} \le T_j \le T_{j\,max} \\
&\sigma_{min} \le \sigma \le \sigma_{max} \\
&\left\| \frac{dC_s}{dt} \right\| \le k_1
\end{aligned}
\tag{14.17}
$$

$$
n(0, t) \le n_{fine}, \quad \forall \ t \ge t_f/2 \tag{14.18}
$$

where T_{min} and T_{max} are the constraints on the crystallizer temperature, T, and are specified as 4°C and 22°C, respectively. $T_{j\,min}$ and $T_{j\,max}$ are the constraints on the manipulated variable, T_j, and are

specified as 3°C and 22°C, respectively. The constraints on the supersaturation σ are $\sigma_{min} = 1.73$ and $\sigma_{max} = 2.89$. The constant, k_1 (chosen to be 0.065 mg/mL min), specifies the maximum rate of change of the saturation concentration C_s. n_{fine} is the largest allowable number of nuclei at any time instant during the second half of the batch run, and is set to 5/μm mL. In the simulation, the sampling time is 5 min, while the batch process time t_f is 24 h. The optimization problem is solved using sequential quadratic programming (SQP). A second-order accurate finite-difference scheme with 3000 discretization points is used to obtain the solution of the population balance model of Equation 14.3. Referring to the predictive control formulation of Equation 14.18, it is important to note that previous work has shown that the objective of maximizing the volume-averaged crystal size can result in a large number of fines in the final product [41]. To enhance the ability of the predictive control strategy to maximize the performance objective while avoiding the formation of a large number of fines in the final product, the predictive controller of Equations 14.7 and 14.8 includes a constraint (Equation 14.18) on the number of fines present in the final product. Specifically, the constraint of Equation 14.18, by restricting the number of nuclei formed at any time instant during the second half of the batch run limits the fines in the final product. Note that predictive control without constraint on fines can result in a product with a large number of fines (see Figure 14.9a) which is undesirable. The implementation of the predictive controller with the constraint of Equation 14.18, designed to reduce the fines in the product, results in a product with much less fines while still maximizing the volume-averaged crystal size (see Figure 14.9b). The reader may refer to [14,30] for further results on the performance of the predictive controller and comparisons with the performance of two other open-loop control strategies, constant temperature control (CTC) and constant supersaturation control (CSC).

14.2.4.4 Fault-Tolerant Control of Particulate Processes

Compared with the significant and growing body of research work on the feedback control design of particulate processes, the problem of designing fault-tolerant control systems for particulate processes has not received much attention. This is an important problem given the vulnerability of automatic control systems to faults (e.g., malfunctions in the control actuators, measurement sensors or process equipment), and the detrimental effects that such faults can have on the process operating efficiency and product quality. Given that particulate processes play a key role in a wide range of industries (e.g., chemical, food, and pharmaceutical) where the ability to consistently meet stringent product specifications is critical to the product utility, it is imperative that systematic methods for the timely diagnosis and handling of faults be developed to minimize production losses that could result from operational failures.

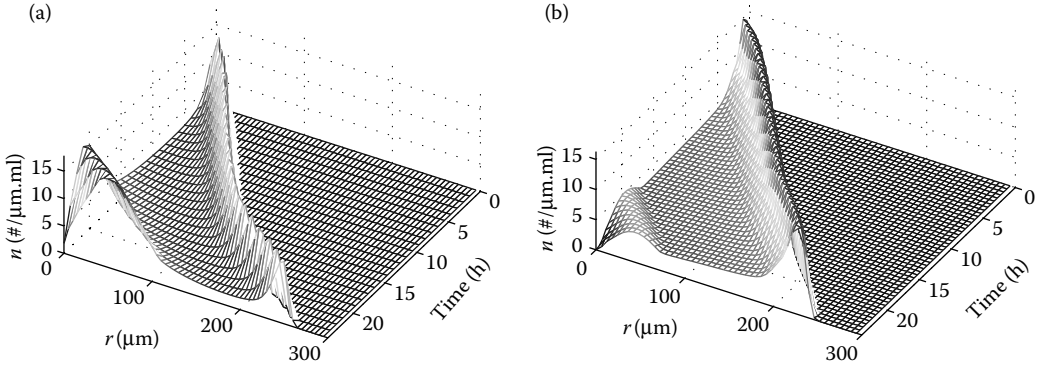

FIGURE 14.9 Evolution of particle size distribution under (a) Predictive control without constraint on fines, and (b) Predictive control with constraint on fines.

Motivated by these considerations, recent research efforts have started to tackle this problem by bringing together tools from model-based control, infinite-dimensional systems, fault diagnosis and hybrid systems theory. For particulate processes modeled by population balance equations with control constraints, actuator faults, and a limited number of process measurements, a fault-tolerant control architecture that integrates model-based fault detection, feedback, and supervisory control has recently been developed in [42]. The architecture, which is based on reduced-order models that capture the dominant dynamics of the particulate process, consists of a family of control configurations, together with a fault detection filter and a supervisor. For each configuration, a stabilizing output feedback controller with well-characterized stability properties is designed through the combination of a state feedback controller and a state observer that uses the available measurements of the principal moments of the PSD and the continuous-phase variables to provide appropriate state estimates. A fault detection filter that simulates the behavior of the fault-free, reduced-order model is then designed, and its discrepancy from the behavior of the actual process state estimates is used as a residual for fault detection. Finally, a switching law based on the stability regions of the constituent control configurations is derived to reconfigure the control system in a way that preserves closed-loop stability in the event of fault detection. Appropriate fault detection thresholds and control reconfiguration criteria that account for model reduction and state estimation errors were derived for the implementation of the control architecture on the particulate process. The methodology was successfully applied to a continuous crystallizer example where the control objective was to stabilize an unstable steady state and achieve a desired CSD in the presence of constraints and actuator faults.

In addition to the synthesis of actuator fault-tolerant control systems for particulate processes, recent research efforts have also investigated the problem of preserving closed-loop stability and performance of particulate processes in the presence of sensor data losses [32]. Typical sources of sensor data losses include measurement sampling losses, intermittent failures associated with measurement techniques, as well as data packet losses over transmission lines. In this work, two representative particulate process examples—a continuous crystallizer and a batch protein crystallizer—were considered. In both examples, feedback control systems were first designed on the basis of low-order models and applied to the population balance models to enforce closed-loop stability and constraint satisfaction. Subsequently, the robustness of the control systems in the presence of sensor data losses was investigated using a stochastic formulation developed in [43] that models sensor failures as a random Poisson process. In the case of the continuous crystallizer, a Lyapunov-based nonlinear output feedback controller was designed and shown to stabilize an open-loop unstable steady state of the population balance model in the presence of input constraints. Analysis of the closed-loop system under sensor malfunctions showed that the controller is robust with respect to significant sensor data losses, but cannot maintain closed-loop stability when the rate of data losses exceeds a certain threshold. In the case of the batch crystallizer, a predictive controller was designed to obtain a desired CSD at the end of the batch while satisfying state and input constraints. Simulation results showed how constraint modification in the predictive controller formulation can assist in achieving constraint satisfaction under sensor data losses.

14.2.4.5 Nonlinear Control of Aerosol Reactors

The crystallization process examples discussed in the previous section share the common characteristic of having two independent variables (time and particle size). In such a case, order reduction, for example, with the method of moments, leads to a set of ODEs in time as a reduced-order model. This is not the case, however, when three or more independent variables (time, particle size, and space) are used in the process model. An example of such a process is the aerosol flow reactor presented in the Introduction section. The complexity of the partial integro-differential equation model of Equation 14.6 does not allow its direct use for the synthesis of a practically implementable nonlinear model-based feedback controller for spatially inhomogeneous aerosol processes. Therefore, a model-based controller design method for spatially inhomogeneous aerosol processes was developed [16,17,29], which is based on the experimental

observation that many aerosol size distributions can be adequately approximated by lognormal functions. The proposed control method can be summarized as follows:

1. Initially, the aerosol size distribution is assumed to be described by a lognormal function and the method of moments is applied to the aerosol population balance model of Equation 14.6 to compute a hyperbolic partial differential equation (PDE) system (where the independent variables are time and space) that describes the spatio-temporal behavior of the three leading moments needed to exactly describe the evolution of the lognormal aerosol size distribution.
2. Then, nonlinear geometric control methods for hyperbolic PDEs [44] are applied to the resulting system to synthesize nonlinear distributed output feedback controllers that use process measurements at different locations along the length of the process to adjust the manipulated input (typically, wall temperature), in order to achieve an aerosol size distribution with desired characteristics (e.g., geometric average particle volume).

An application of this nonlinear control method was carried to an aerosol flow reactor, including nucleation, condensation, and coagulation, used to produce NH_4Cl particles [16] and a titania aerosol reactor [17]. Specifically, for an aerosol flow reactor used to produce NH_4Cl particles, the following chemical reaction takes place $NH_3 + HCl \rightarrow NH_4Cl$, where NH_3, HCl are the reactant species and NH_4Cl is the monomer product species. Under the assumption of lognormal aerosol size distribution, the mathematical model that describes the evolution of the first three moments of the distribution, together with the monomer (NH_4Cl) and reactant (NH_3, HCl) concentrations and reactor temperature takes the form:

$$
\begin{aligned}
\frac{\partial N}{\partial \theta} &= -v_{zl}\frac{\partial N}{\partial \bar{z}} + I' - \xi N^2 \\
\frac{\partial V}{\partial \theta} &= -v_{zl}\frac{\partial V}{\partial \bar{z}} + I'k^* + \eta(S-1)N \\
\frac{\partial V_2}{\partial \theta} &= -v_{zl}\frac{\partial V_2}{\partial \bar{z}} + I'k^{*2} + 2\epsilon(S-1)V + 2\zeta V^2 \\
\frac{\partial S}{\partial \theta} &= -v_{zl}\frac{\partial S}{\partial \bar{z}} + C\bar{C}_1\bar{C}_2 - I'k^* - \eta(S-1)N \\
\frac{\partial \bar{C}_1}{\partial \theta} &= -v_{zl}\frac{\partial \bar{C}_1}{\partial \bar{z}} - A_1\bar{C}_1\bar{C}_2 \\
\frac{\partial \bar{C}_2}{\partial \theta} &= -v_{zl}\frac{\partial \bar{C}_2}{\partial \bar{z}} - A_2\bar{C}_1\bar{C}_2 \\
\frac{\partial \bar{T}}{\partial \theta} &= -v_{zl}\frac{\partial \bar{T}}{\partial \bar{z}} + B\bar{C}_1\bar{C}_2\bar{T} + E\bar{T}(\bar{T}_w - \bar{T})
\end{aligned}
\tag{14.19}
$$

where $\bar{C}_1$ and $\bar{C}_2$ are the dimensionless concentrations of NH_3 and HCl, respectively, $\bar{T}, \bar{T}_w$ are the dimensionless reactor and wall temperatures, respectively, and A_1, A_2, B, C, E are dimensionless quantities [16].

Figure 14.10 displays the steady-state profile of the dimensionless total particle concentration, N, as a function of reactor length. As expected, N increases very fast close to the inlet of the reactor (approximately, the first 3% of the reactor), due to a nucleation burst, and then, it slowly decreases in the remaining part of the reactor due to coagulation. Even though coagulation decreases the total number of particles, it leads to the formation of bigger particles, and thus, increases the geometric average particle volume, v_g. The control problem is formulated as the one controlling the geometric average particle volume in the outlet of the reactor, $v_g(1, \theta)$, ($v_g(1, \theta)$ is directly related to the geometric average particle diameter, and hence, it is a key product characteristic of industrial aerosol processes) by manipulating the wall

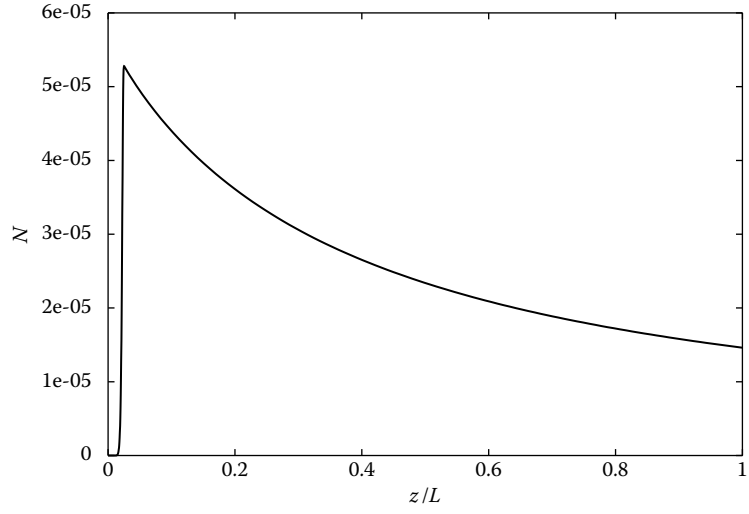

FIGURE 14.10 Steady-state profile of dimensionless particle concentration.

temperature, that is:

$$y(\theta) = \mathcal{C}v_g = v_g(1,\theta), \quad u(\theta) = \bar{T}_w(\theta) - \bar{T}_{ws} \tag{14.20}$$

where $\mathcal{C}(\cdot) = \int_0^1 \delta(\bar{z}-1)(\cdot)\,dz$ and $\bar{T}_{ws} = T_{ws}/T_o = 1$. Since coagulation is the main mechanism that determines the size of the aerosol particles, one may focus on controlling the part of the reactor where coagulation occurs. Therefore, the wall temperature is assumed to be equal to its steady-state value in the first 3.5% of the reactor (where nucleation mainly occurs), and it is adjusted by the controller in the remaining part of the reactor (where coagulation takes place).

The model of Equation 14.19 was used as the basis for the synthesis of a nonlinear controller utilizing the above-mentioned control method. For this model, σ was found to be equal to 2 and the necessary controller was synthesized using the nonlinear distributed state feedback formula developed in [44] and is of the form:

$$u = \left[\mathcal{C}\gamma_\sigma L_g \left(\sum_{j=1}^{n} \frac{\partial x_j}{\partial \bar{z}} L_{a_j} + L_f\right) h(x) b(\bar{z})\right]^{-1} \left\{y_{sp} - \mathcal{C}h(x) - \sum_{\nu=1}^{2} \mathcal{C}\gamma_\nu \left(\sum_{j=1}^{n} \frac{\partial x_j}{\partial \bar{z}} L_{a_j} + L_f\right)^{\nu} h(x)\right\} \tag{14.21}$$

where $\gamma_1 = 580$ and $\gamma_2 = 1.6 \times 10^5$, to enforce a slightly underdamped response.

Two simulation runs were performed to evaluate the set-point tracking capabilities of the nonlinear controller and its performance was compared with that of a PI controller. In both simulation runs, the aerosol reactor was initially assumed to be at steady state and a 5% increase in the set-point value of $v_g(1,0)$ was imposed at $t = 0$ s (i.e., $y_{sp} = 1.05 v_g(1,0)$). Figure 14.11 (left plot–solid line) shows the profile of the controlled output which is the mean particle volume at the outlet of the reactor $v_g(1,t)$, while Figure 14.11 (right plot–solid line) displays the corresponding profile of the manipulated input which is the wall temperature. The nonlinear controller of Equation 14.21 regulates successfully, $v_g(1,t)$ to its new set-point value. For the sake of comparison, it was also implemented on the process a PI controller; this controller was tuned so that the time which the closed-loop output needs to reach the final steady state is the same as that of the closed-loop outputs under nonlinear control. The profiles of the controlled output and manipulated input are shown in Figure 14.11 (dashed lines show the corresponding profiles for the PI controller). It is clear that the nonlinear controller outperforms the PI controller.

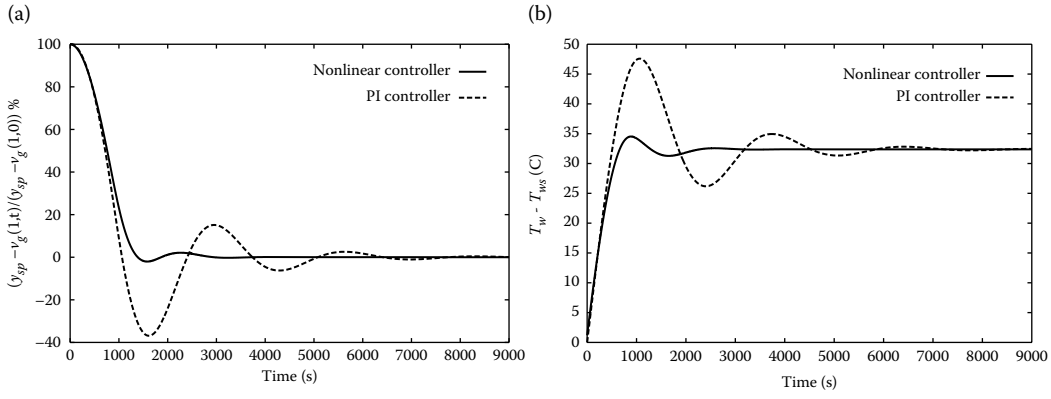

FIGURE 14.11 (a) Closed-loop profiles of scaled mean particle volume in the outlet of the reactor under PI and nonlinear controllers. (b) Manipulated input profiles for PI and nonlinear controllers.

14.3 Conclusions

Control of particulate processes systems is a cross-disciplinary and rapidly growing research area that brings together fundamental modeling, numerical simulation, nonlinear dynamics, and control theory. This chapter presents recent advances in systematic methods for the design of easy-to-implement nonlinear feedback controllers for broad classes of particulate processes. It is expected that the feedback control will play an important role in the synthesis and processing of nanosize and microsize particles with the everincreasing research and development in advanced materials and semiconductor manufacturing, nanotechnology, and biotechnology. The reader may refer to [31] for a detailed discussion on future problems on control of particulate processes.

References

1. P. A. Larsen, J. B. Rawlings, and N. J. Ferrier. An algorithm for analyzing noisy, *in situ* images of high-aspect-ratio crystals to monitor particle size distribution. *Chem. Eng. Sci.*, 61:5236–5248, 2006.
2. J. B. Rawlings, S. M. Miller, and W. R. Witkowski. Model identification and control of solution crystallizatin process—a review. *Ind. Eng. Chem. Res.*, 32:1275–1296, 1993.
3. J. B. Rawlings, C. W. Sink, and S. M. Miller. Control of crystallization processes. In *Industrial Crystallization—Theory and Practice*, pp. 179–207, Butterworth, Boston, 1992.
4. H. M. Hulburt and S. Katz. Some problems in particle technology: A statistical mechanical formulation. *Chem. Eng. Sci.*, 19:555–574, 1964.
5. D. Ramkrishna. The status of population balances. *Rev. Chem. Eng.*, 3:49–95, 1985.
6. J. Dimitratos, G. Elicabe, and C. Georgakis. Control of emulsion polymerization reactors. *AIChE J*, 40:1993–2021, 1994.
7. F. J. Doyle, M. Soroush, and C. Cordeiro. Control of product quality in polymerization processes. In *AIChE Symposium Series: Proceedings of 6th International Conference on Chemical Process Control*, Rawlings, J. B. et al. (Eds.), pp. 290–306, American Institute of Chemical Engineers, New York, NY, 2002.
8. R. D. Braatz and S. Hasebe. Particle size and shape control in crystallization processes. In *AIChE Symposium Series: Proceedings of 6th International Conference on Chemical Process Control*, Rawlings, J. B. et al. (Eds.), pp. 307–327, American Institute of Chemical Engineers, New York, NY, 2002.
9. S. K. Friendlander. *Smoke, Dust and Haze: Fundamentals of Aerosol Dynamics* (2nd Ed.). Oxford University Press, New York, NY, 2000.
10. P. Daoutidis and M. Henson. Dynamics and control of cell populations. In *Proceedings of 6th International Conference on Chemical Process Control*, pp. 308–325, Tucson, AZ, 2001.

11. G. R. Jerauld, Y. Vasatis, and M. F. Doherty. Simple conditions for the appearance of sustained oscillations in continuous crystallizers. *Chem. Eng. Sci.*, 38:1675–1681, 1983.
12. S. J. Lei, R. Shinnar, and S. Katz. The stability and dynamic behavior of a continuous crystallizer with a fines trap. *AIChE J.*, 17:1459–1470, 1971.
13. T. Chiu and P. D. Christofides. Nonlinear control of particulate processes. *AIChE J.*, 45:1279–1297, 1999.
14. D. Shi, P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of crystal size distribution in protein crystallization. *Nanotechnology*, 16:S562–S574, 2005.
15. P. G. Vekilov and F. Rosenberger. Dependence of lysozyme growth kinetics on step sources and impurities. *J. Cryst. Growth*, 158:540–551, 1996.
16. A. Kalani and P. D. Christofides. Nonlinear control of spatially-inhomogeneous aerosol processes. *Chem. Eng. Sci.*, 54:2669–2678, 1999.
17. A. Kalani and P. D. Christofides. Modeling and control of a titania aerosol reactor. *Aerosol Sci. Technol.*, 32:369–391, 2000.
18. F. Gelbard and J. H. Seinfeld. Numerical solution of the dynamic equation for particulate processes. *J. Comput. Phys.*, 28:357–375, 1978.
19. K. Lee and T. Matsoukas. Simultaneous coagulation and break-up using constant-number monte Carlo. *Powder Technol.*, 110:82–89, 2000.
20. Y. L. Lin, K. Lee, and T. Matsoukas. Solution of the population balance equation using constant-number Monte Carlo. *Chem. Eng. Sci.*, 57:2241–2252, 2002.
21. T. Smith and T. Matsoukas. Constant-number Monte Carlo simulation of population balances. *Chem. Eng. Sci.*, 53:1777–1786, 1998.
22. D. Semino and W. H. Ray. Control of systems described by population balance equations-I. Controllability analysis. *Chem. Eng. Sci.*, 50:1805–1824, 1995.
23. G. Hu, J. Huang, G. Orkoulas, and P. D. Christofides, Investigation of film surface roughness and porosity dependence on lattice size in a porous thin film deposition process, *Phys. Rev. E.*, 80:041122, 2009.
24. D. Semino and W. H. Ray. Control of systems described by population balance equations-II. Emulsion polymerization with constrained control action. *Chem. Eng. Sci.*, 50:1825–1839, 1995.
25. T. Chiu and P. D. Christofides. Robust control of particulate processes using uncertain population balances. *AIChE J.*, 46:266–280, 2000.
26. P. D. Christofides. *Model-Based Control of Particulate Processes*. Kluwer Academic Publishers, Particle Technology Series, Netherlands, 2002.
27. P. D. Christofides and T. Chiu. Nonlinear control of particulate processes. In *AIChE Annual Meeting, Paper 196a*, Los Angeles, CA, 1997.
28. N. H. El-Farra, T. Chiu, and P. D. Christofides. Analysis and control of particulate processes with input constraints. *AIChE J.*, 47:1849–1865, 2001.
29. A. Kalani and P. D. Christofides. Simulation, estimation and control of size distribution in aerosol processes with simultaneous reaction, nucleation, condensation and coagulation. *Comput. Chem. Eng.*, 26:1153–1169, 2002.
30. D. Shi, N. H. El-Farra, M. Li, P. Mhaskar, and P. D. Christofides. Predictive control of particle size distribution in particulate processes. *Chem. Eng. Sci.*, 61:268–281, 2006.
31. P. D. Christofides, M. Li, and L. Mädler. Control of particulate processes: Recent results and future challenges. *Powder Technol.*, 175:1–7, 2007.
32. A. Gani, P. Mhaskar, and P. D. Christofides. Handling sensor malfunctions in control of particulate processes. *Chem. Eng. Sci.*, 63:1217–1229, 2008.
33. C. F. Bohren and D. R. Huffman. *Absorption and Scattering of Light by Small Particles*. Wiley, New York, 1983.
34. N. H. El-Farra and P. D. Christofides. Integrating robustness, optimality, and constraints in control of nonlinear processes. *Chem. Eng. Sci.*, 56:1–28, 2001.
35. N. H. El-Farra and P. D. Christofides. Bounded robust control of constrained multivariable nonlinear processes. *Chem. Eng. Sci.*, 58:3025–3047, 2003.
36. Y. Lin and E. D. Sontag. A universal formula for stabilization with bounded controls. *Syst. Contr. Lett.*, 16:393–397, 1991.
37. N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid predictive control of nonlinear systems: Method and applications to chemical processes. *Int. J. Robust Nonlinear Control*, 14:199–225, 2004.
38. W. Xie, S. Rohani, and A. Phoenix. Dynamic modeling and operation of a seeded batch cooling crystallizer. *Chem. Eng. Commun.*, 187:229–249, 2001.

39. S. M. Miller and J. B. Rawlings. Model identification and control strategies for batch cooling crystallizers. *AIChE J.*, 40:1312–1327, 1994.
40. G. P. Zhang and S. Rohani. On-line optimal control of a seeded batch cooling crystallizer. *Chem. Eng Sci.*, 58:1887–1896, 2003.
41. D. L. Ma, D. K. Tafti, and R. D. Braatz. Optimal control and simulation of multidimensional crystallization processes. *Comput. Chem. Eng.*, 26:1103–1116, 2002.
42. N. H. El-Farra and A. Giridhar. Detection and management of actuator faults in controlled particulate processes using population balance models. *Chem. Eng. Sci.*, 63:1185–1204, 2008.
43. P. Mhaskar, A. Gani, C. McFall, P. D. Christofides, and J. F. Davis. Fault-tolerant control of nonlinear process systems subject to sensor faults. *AIChE J.*, 53:654–668, 2007.
44. P. D. Christofides and P. Daoutidis. Feedback control of hyperbolic PDE systems. *AIChE J.*, 42:3063–3086, 1996.

15

Nonlinear Model Predictive Control for Batch Processes

Zoltan K. Nagy
Loughborough University

Richard D. Braatz
University of Illinois at Urbana-Champaign

15.1 Introduction

Batch processes are well suited for the manufacture of low-volume high-value products and are often used for the flexible manufacturing of multiple related products in the same facility. Batch processes are usually preferred for production volumes below 10,000 Mt/year, whereas continuous processes are predominantly used when the production volume is in the order of 100,000 Mt/year. Batch processes are

the production scheme of choice for the pharmaceutical, biotechnology, specialty chemical, consumer products, and microelectronics industries. The production of these high-value-added chemicals contributes a significant and growing portion of the revenue and earnings of the chemical process industries. Batch processes are heavily used in the pharmaceutical industry where isolation and product consistency are required for reasons of safety and sterility. Additionally, the shutdown and startup of high-volume production continuous processes under economic and safety constraints can be treated as a batch control problem. Despite their widespread application, batch recipes and control strategies are often designed on an empirical basis in industrial practice due to costs for developing process models and the limited quantity of online instrumentation available. Over the last decade, increased availability of suitable models and measurement devices has lead to an increased interest in the dynamic (online) optimization, which has the potential of yielding significant performance and quality improvements. A series of potential issues must be addressed for a successful application of optimal batch control strategies. Usually the initial conditions are only roughly known, most states are unmeasured, and disturbances and model uncertainties are present. These problems must be taken into account during control systems design, because even small disturbances and model uncertainties can lead to significant performance degradation in batch processes.

A particular characteristic of batch processes is the absence of steady state. Batch processes develop from an initial state to a generally very different final stage. Consequently, it is not possible to identify a single operating point around which a control system could be designed using well-established linear control design approaches commonly used for continuous processes. Another feature of batch processes is the very high importance of constraints. Due to the wide operating range during the batch, generally batch processes are operated under active constraints, especially in the case of optimal designs.

Another distinguishing feature of batch processes is their irreversible behavior. For many batch processes such as polymerization or microelectronics processes (e.g., manufacture of wafers), there is a very small tolerance to processing mistakes, with no possibilities of recovering products that are outside the tight specification limits through further processing. This is in contrast to continuous processes where a control input generally can bring back the process to the original operating point. This characteristic reinforces the need for advanced control strategies for batch processes, which can provide consistent end-product properties.

Additionally, batch processes are usually characterized by repetition of the batch runs. This repetitive nature offers the possibility to use the information from previous batch runs to improve the performance of subsequent runs within the framework of *batch-to-batch* or *iterative learning control* approaches.

The aforementioned particular features of batch processes define the control challenges for this class of systems, but they also offer a ground of opportunities for many novel control and design approaches with very high economical and social impact, governed by the highly competitive and profit driven nature of today's process industries.

15.2 Overview of Batch Process Control

Often batch process control involves the offline model-based optimization of the process and then the implementation of the optimal trajectories/recipes by the use of classical feedback control approaches such as proportional-integral-derivative (PID) control. Process optimization has the potential to reduce production costs, improve product quality, reduce product variability, and ease scaleup. The benefits of batch optimization can occur during initial product design, process operation, and scaleup stages. At the pilot plant and production stages, batch and semibatch optimization can contribute significantly to the achievement of consistency of production and the minimization of batch production times. In these cases, the model uncertainties should be taken into account to ensure that the desired process performance improvements are achieved. A brief discussion on robust optimal control strategies in the context of batch nonlinear predictive control, which is also valid for open-loop control, is provided in

Section 15.3.5. Since the open-loop optimal control of batch processes can be considered as a particular case of the batch nonlinear predictive control approach presented in detail in the next section, it will not be discussed further here.

To accommodate changing process conditions and confer the inherent robustness coming from feedback to the batch optimal control problem, the optimizations can be repeated online, whenever a new set of measurements becomes available, leading to real-time explicit optimization-based schemes such as *nonlinear model predictive control* (NMPC). NMPC techniques are becoming increasingly accepted as theoretically being the best choice for advanced batch process control, due to their ability to cope with process constraints, nonlinearities, and different objectives derived from economical or environmental considerations. However, despite the significant and continuously increasing importance of batch processes, the number of NMPC applications is significantly lower than in the case of continuous processes (Qin and Badgwell, 2003). Although the inherent nonlinearity of batch processes suggests NMPC as a natural choice for the advanced control of these systems, most industrial NMPC vendors do not support typical batch NMPC problems. Few vendor companies provide solutions for *batch nonlinear model predictive control* (BNMPC) applications (e.g., Cybernetica and IPCOS). This can be explained mainly by the special features of batch processes that make their control very challenging. Batch processes are characterized by strongly nonlinear time-varying dynamics and are often described by complicated process models, leading to strongly nonlinear optimization problems that can be difficult to solve online to provide real-time feasibility and the stability and robustness needed in an industrial environment. The industrial implementations of BNMPC have been mostly for high-value-added processes where the economic potential for improvement was large.

To decrease the computational complexity, another category of control approaches has been proposed based on real-time implicit optimization schemes that are often referred to as *measurement-based optimization* approaches. One category of implicit optimization schemes employs an update law that approximates the optimal solution by minimizing the second-order variation of the cost function subject to the linearized dynamical constraints. Another category of implicit optimization approaches is based on simple representation of the process/controller model, and the application of Pontryagin's Maximum Principle to derive the necessary conditions of optimality, which are tracked using current measurements. Although these approaches significantly reduce the online computational burden, have been applied to real applications, and provide an alluring robust control framework, the approaches require significant offline effort to derive the solution model and the optimality conditions. Additionally, current trends in the chemical industries are for companies to increasingly derive detailed first-principles models for most of their important processes, for simulation and operator-training purposes. NMPC is the focus of this chapter because it offers a generic framework to incorporate existing models in a uniform and flexible framework.

The repetitive nature of batch processes provides scope for the development of batch-to-batch (also called *run-to-run*) control approaches, where the main objective is to provide control over a number of batches rather than within a single batch, and to achieve convergence of the product quality over several batch runs, by using the information from previous batches in order to design the control law for the next batch. Batch-to-batch control can cope with a low quantity or quality of measurements during the batch. Generally, substantially more data and computational time are available at the end of the batch to refine the model by adapting its parameters and structure, and to redesign the control input and recipe for the next run. A drawback of batch-to-batch control is that convergence of the control performance occurs only over several batches; failure of batches may occur during the initial few numbers of runs and due to unexpected disturbances within the batch. Also the approach may provide poor performance in the case of uncertain initial conditions and errors in the recipe implementation. Batch-to-batch control approaches can be implemented in the framework of both explicit (model-based) or implicit (measurement-based) optimization. Combined batch-to-batch and within-batch approaches have also been reported, which provide control on the two timescales of batch processes to merge the advantages of both approaches. Although many aspects of the explicit batch-to-batch control approaches can be discussed in the generic framework of batch NMPC, this chapter focuses on the within-batch NMPC approach.

15.3 Computationally Efficient Real-Time Output Feedback NMPC for Batch Processes

15.3.1 Problem Formulation of BNMPC

NMPC is an optimization-based multivariable constrained control technique that incorporates a nonlinear dynamic model for the prediction of the process outputs. At each sampling time, the model is updated on the basis of new measurements and state variable estimates. Then the open-loop optimal manipulated variable moves are calculated over a finite prediction horizon with respect to some cost function, and the manipulated variables for the subsequent prediction horizon are implemented. Then the prediction horizon is shifted or shrunk by usually one sampling time into the future and the previous steps are repeated. The optimal control problem to be solved online in each sampling time in the BNMPC algorithm is usually formulated as

$$\textit{Problem } P_1(t_k): \qquad \min_{u(t)\in\mathcal{U}, t_k^F} \ \mathcal{H}(x(t), u(t); \theta), \tag{15.1}$$

subject to:

$$\dot{x}(t) = f(x(t), u(t); \theta), \tag{15.2}$$

$$y(t) = g(x(t), u(t); \theta), \tag{15.3}$$

$$x(t_k) = \hat{x}(t_k), \tag{15.4}$$

$$h(x(t), u(t); \theta) \le 0, \quad t \in [t_k, t_k^F], \tag{15.5}$$

where $\mathcal{H}$ is the performance objective, t is the time, t_k is the time at sampling time k, $t_0 = 0$ is the initial time at the beginning of the batch, $x(t) \in \mathbb{R}^{n_x}$ is the n_x vector of states, $u(t) \in \mathcal{U}$ is the n_u set of input vectors, $y(t) \in \mathbb{R}^{n_y}$ is the n_y vector of measured variables used to compute the estimated states $\hat{x}(t_k)$ with initial values $\hat{x}_0$, $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ is the n_θ vector of possible uncertain parameters, where the set Θ can be either defined by hard bounds or probabilistically (characterized by a multivariate probability density function (pdf)). The function $f : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}^{n_x}$ is the twice continuously differentiable vector function of the dynamic equations of the system, $g : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}^{n_y}$ is the measurement equations function, and $h : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}^{c}$ is the vector of functions that describe all linear and nonlinear, time-varying or end-time, algebraic constraints for the system, where c denotes the number of these constraints. The objective function can have the general form

$$\mathcal{H}(x(t), u(t); \theta) = \mathcal{M}(x(t_k^F); \theta) + \int_{t_k}^{t_k^F} \mathcal{L}(x(t), u(t); \theta)\, \mathrm{dt}, \tag{15.6}$$

where $\mathcal{H} : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}$ is twice continuously differentiable, which enables the application of fast optimization algorithms based on first- and second-order derivatives in the solution of Equation 15.6. The objective function $\mathcal{H}$ consists of a terminal cost function, $\mathcal{M} : \mathbb{R}^{n_x} \times \Theta \to \mathbb{R}$, and a running cost function, $\mathcal{L} : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}$. The form of Equation 15.6 is general enough to express a wide range of objectives encountered in NMPC applications (such as use of the moving- or shrinking-horizon approach for regulation and setpoint tracking, direct minimization of the operation time, optimization of initial conditions as needed for optimal recipe design, multiple simultaneous objectives, treatment of soft constraints, and a terminal penalty term). For batch processes with end-point optimization, the objective usually reduces to the Mayer form ($\mathcal{L}(\cdot) = 0$), with the Lagrange term ($\mathcal{L}(\cdot)$) useful for implementation of soft constraints on the control rate or for setpoint tracking. In BNMPC, the optimization problem (Equations 15.1 through 15.5) is solved iteratively online, in a *moving horizon* with $t_k^F = t_k + T_p \le t_f$ or in a *shrinking horizon* ($t_k^F = t_f$), where T_p is the prediction horizon and t_f the batch time. In the case of BNMPC with an end-point objective, it is typical that the batch time t_f is also included as a decision variable in the optimization. In the case when $t_k^F = t_f$ and $k = 0$, the problem (Equations 15.1 through 15.5) is equivalent to a typical open-loop batch optimization problem as described in Section 15.2.

15.3.2 NMPC of Batch Reactor Operations

The control of batch processes presents several particularities. The major difference, compared to the operation of continuous processes, is that important properties such as Lyapunov stability do not have to hold because the batch operates only over a finite time. The controller performance is assessed based on the chosen objective function, the constraints, and robustness against model/plant mismatch and control implementation uncertainties. While the computational burden in continuous processes can be reduced by choosing a shorter control horizon than the prediction horizon, in the case of BNMPC the control and prediction horizons should be equal, to avoid large deviations of the predicted quantities from their (usually time-varying) setpoints due to the highly transient character of the process. This equality significantly increases the computational demand during the initial part of the batch. On the positive side, shrinking-horizon BNMPC leads to gradually decreasing computational requirements as the prediction horizon decreases during the batch.

Generally, in batch process operation, two different BNMPC problems arise, which are often equally important and often need to be applied in combination:

a. *Setpoint tracking*, whether determined offline or online, usually time-varying setpoint trajectories have to be followed in a moving or shrinking (or combination of the two) horizon approach. In this case, usually a quadratic (least-squares type) objective function is used, which allows for efficient Hessian approximation (e.g., based on the constrained Gauss–Newton method). A typical setpoint tracking problem is formulated as

$$u^{\star} = \underset{u_k,\ldots,u_{k+N_p}}{\arg\min} \left\{ \sum_{i=k+1}^{k+N_p} \left\| y_i - y_i^{ref} \right\|_{Q_y}^2 + \sum_{i=k}^{k+N_p} \left(\left\| u_i - u_i^{ref} \right\|_{Q_u}^2 + \| u_i - u_{i-1} \|_{Q_{\Delta u}}^2 \right) \right\}, \tag{15.7}$$

subject to the model equations, where y^{ref} and u^{ref} are the output and input references, respectively, and Q_y, Q_u, $Q_{\Delta u}$ are weighting matrices. This formulation is similar to NMPC for continuous processes, with the main difference being that the reference trajectory and the process dynamics are changing during the duration of the batch. This type of control problem is often encountered in microelectronics applications where suitable models are usually not available to relate the end-point property to the setpoint trajectory, or when the optimal trajectory is insensitive to disturbances. In the pharmaceutical industries, regulatory aspects can sometimes prevent the online adaptation of setpoint trajectories, so that the batch optimal control may be required to focus only on enhanced tracking performance. The control problem of type (a) is the only option for many batch processes, for which the loss of observability or controllability due to the lack of available measurements does not allow the use of complex models required for the prediction of end-point properties.

b. *End-point property control* (setpoint reoptimization, or online optimization approach). In batch processes, the real economic objective is usually related to the product quality at the end of the batch, leading to the formulation of a different control problem that is always implemented in a shrinking-horizon approach, formulated usually as

$$u^{\star} = \underset{u_k,\ldots,u_N}{\arg\min}\, \mathcal{M}(x_N), \tag{15.8}$$

where N is the number of discretization time periods for the entire batch period $[0, t_f]$.

Problem (b) is often solved offline and the resulting control trajectory is considered fixed (given by the recipe) and used as the setpoint in a type (a) problem. In this case, a setpoint tracking controller will follow the given setpoint as closely as possible, and in case of disturbances will try to minimize the deviations from the given trajectories. In case of certain disturbance scenarios, the initially optimal setpoint trajectory may not be optimal anymore, or may not be feasible for the controller, and a new trajectory might be necessary

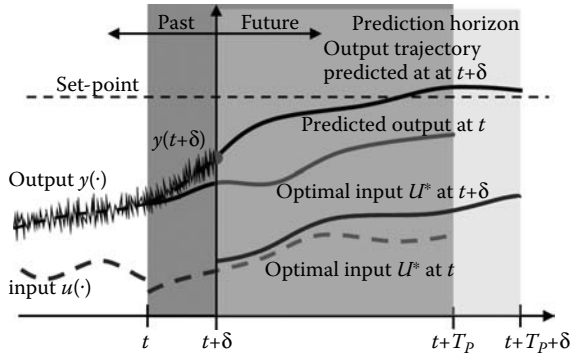

FIGURE 15.1 Main idea of the moving horizon approach in STNMPC.

to maintain product quality and/or process safety. The problem of simultaneous setpoint optimization (in terms of an economic/property objective function), and control input optimization (in terms of a standard quadratic performance criterion) represents an interesting batch process control problem, since the optimal setpoint or reference trajectory may correspond to loss of controllability or be near the limit of stability. Figures 15.1 and 15.2 illustrate the main algorithms of the moving- and shrinking-horizon approaches for the setpoint tracking nonlinear model predictive control (STNMPC) and the end-point criteria-based setpoint optimizing nonlinear model predictive control (SONMPC) algorithms. When both tight setpoint tracking and setpoint adaptation are important and possible, the approaches can be combined together in a hierarchical structure as illustrated in Figure 15.3. Although the two problems can be combined into one BNMPC formulation, the fundamentally different nature of the optimization and control problems allow a more efficient implementation in a hierarchical topology. The required complexity of model and prediction time horizon for the SONMPC is usually significantly higher than what is required for the STNMPC. Hence, models with different complexity and different sampling times can be used at the two levels, allowing a better control in the distributed structure. In both cases, generally full state feedback is not available and measurements are subject to noise and so the initial state vector is provided using a state estimation algorithm such as an *Extended Kalman Filter* (EKF, see Section 15.3.6), which can provide the required initial state vector for model predictions, filter measurement noise, and adapt the model if required via a parameter adaptive estimation scheme.

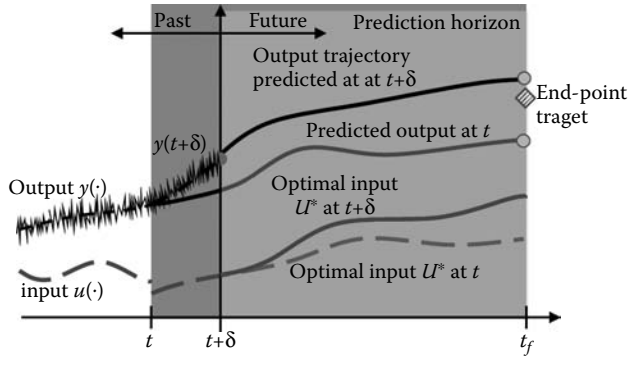

FIGURE 15.2 Main idea of the shrinking-horizon approach in SONMPC.

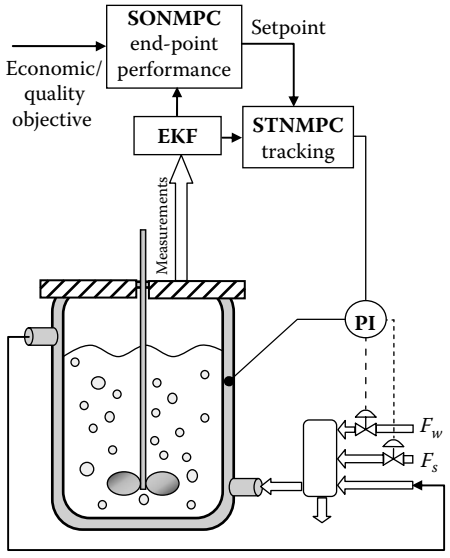

FIGURE 15.3 Schematic diagram of the hierarchical BNMPC of a batch chemical reactor.

15.3.3 Computational Aspects of the BNMPC Approach

15.3.3.1 Efficient Optimization via Direct Multiple Shooting

The optimal control problem $P_1(t_k)$ is rarely solvable analytically. The main idea behind direct methods is based on formulating a discrete approximation that transforms the optimization $P_1(t_k)$ into a nonlinear program (NLP) that can be handled by conventional NLP solvers. The time horizon $t \in \pi_k = [t_k, t_k^F]$ is divided into N_p subintervals (stages) $\pi_{k,i} = [t_{k,i}, t_{k,i+1}]$, $i = 0, 1, \ldots, N_p - 1$, with discrete time steps $t_k = t_{k,0} < t_{k,1} < \cdots < t_{k,i} < t_{k,i+1} < \cdots < t_{k,N_p-1} = t_k^F$. The continuous manipulated variable $u_k(t)$ is parameterized by a piecewise representation $\tilde{u}_{k,i}(t, \mathbf{p}_{k,i})$ for $t \in \pi_{k,i}$ with N_p local control parameter vectors $\mathbf{p}_{k,0}, \mathbf{p}_{k,1}, \ldots, \mathbf{p}_{k,N_p-1}$ with $\mathbf{p}_{k,i} \in \mathbb{R}^{n_p}$ and the optimization problem is solved with the local control parameter vectors being the decision variables. In most practical applications, a piecewise constant or piecewise linear control parametrization is used. For a piecewise constant parameterization, $\tilde{u}_k(t, \mathbf{p}_{k,0}, \ldots, \mathbf{p}_{k,N_p-1}) \triangleq [u_k, u_{k+1}, \ldots, u_{k+N_p-1}]$.

The discrete-time formulation of the optimal control problem $P_1(t_k)$ is

$$\textit{Problem } P_1(t_k): \qquad \min_{u_k, u_{k+1}, \ldots, u_{k+N_p}} \left\{ \mathcal{M}^{k+N_p}(x_{k+N_p}; \theta) + \sum_{j=k}^{k+N_p} \mathcal{L}_j(x_j, u_j; \theta) \right\}, \tag{15.9}$$

subject to

$$G_k(x_k, u_k; \theta) = 0, \tag{15.10}$$

$$H_k(x_k, u_k; \theta) \leq 0, \tag{15.11}$$

where N_p is the number of stages in the prediction horizon $[t_k, t_k^F]$, $G_k : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}^{n_x + n_y}$ corresponds to all equality constraints resulting from the algebraic Equations 15.3 of the model or from the discretized model equations, $H_k : \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \to \mathbb{R}^{c+2n_u}$ is the vector function of all inequality constraints (Equation 15.5), including constraints on the manipulated variables. The set of allowable manipulated variable moves can be written as hard bounds or can be handled by penalizing deviations from constraint satisfaction in the optimization objective (the so-called *soft constraints*). The vector functions G and H are assumed to be twice continuously differentiable. Successive quadratic programming

(SQP) is a common numerical method used to solve NLPs, both due to ready availability of software and reasonably computational efficiency. SQP is a quasi-Newton method that treats nonlinear optimization problems by solving a sequence of local linear-quadratic approximations. The Lagrangian of the optimization problem is approximated quadratically, typically by applying a numerical update formula. Constraints are approximated linearly. SQP methods generally apply the equivalent of Newton steps to the optimality conditions of the NLP problem to achieve a faster rate of convergence.

A very efficient solution technique for the problem (Equations 15.9 through 15.11) is based on the *multiple shooting approach*. The direct multiple shooting procedure consists of dividing up the time interval $[t_k, t_k^F]$ into M subintervals $[\tau_i, \tau_{i+1}]$ via a series of grid points $t_k = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_M = t_k^F$ that do not necessarily correspond to the discretization points (N_p) in the definition of problem $\tilde{P}_1(t_k)$. Using a local control parameterization, a shooting method is performed between successive grid points (see Figure 15.4). The solution of the differential equations for each of the M intervals is decoupled by introducing the initial values ω_i of the states at the multiple shooting nodes τ_i as additional optimization variables. The differential equations and cost on these intervals are numerically integrated independently during each optimization iteration, based on the current guess of the control, and initial conditions ω_i. The continuity/consistency of the final state trajectory at the end of the optimization is enforced by adding consistency constraints to the NLP. The additional interior boundary conditions are incorporated into one large NLP to be solved, which is given in a simplified form as

$$\textit{Problem } P_2: \qquad \min_{v} \mathcal{H}(v;\theta) \quad \text{subject to} \begin{cases} G(v;\theta) = 0 \\ H(v;\theta) \le 0 \end{cases} \tag{15.12}$$

where

$$\mathcal{H}(v;\theta) = \mathcal{M}(\omega_{M+1};\theta) + \sum_{i=0}^{M} \mathcal{L}_i(\omega_i, u_i;\theta), \tag{15.13}$$

and the optimization variable v contains all of the multiple shooting state variables and controls

$$v = [\omega_0, u_0, \omega_1, u_1, \ldots, \omega_{M-1}, u_{M-1}, \omega_M]. \tag{15.14}$$

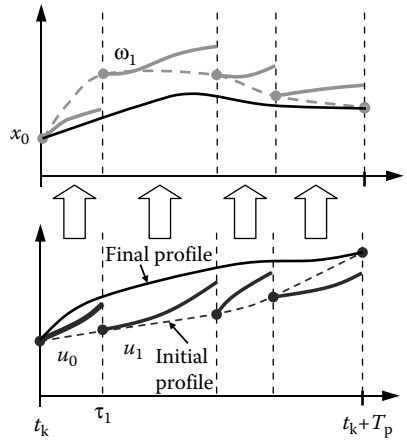

FIGURE 15.4 Illustration of the direct multiple shooting algorithm. (Reprinted from Nagy, Z.K., et al. *Control Eng. Pract.*, 15:839–859, 2007. With permission.)

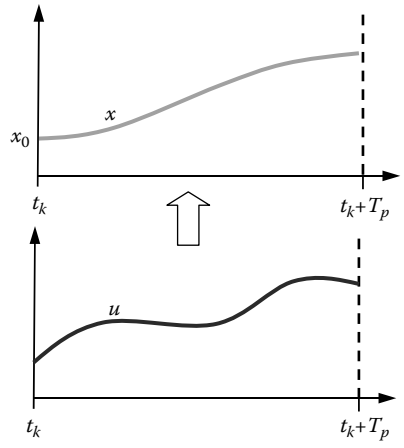

FIGURE 15.5 The sequential solution method for BNMPC problems.

The discretized initial value problem and continuity constraints are included in the equality constraints:

$$G(v;\theta) = \begin{bmatrix} \omega_0 - \hat{x}(t_k) \\ \omega_{i+1} - x_i(t_{i+1};\omega_i,u_i) \\ y - g(\omega_i,u_i;\theta) \end{bmatrix} = 0, \tag{15.15}$$

and the inequality constraints are given by $H(v;\theta) = H(\omega_i,u_i;\theta) \leq 0$, for $i = 0,1,\ldots,M$. The main idea of the direct multiple shooting algorithm is illustrated in Figure 15.4.

15.3.3.2 Sequential Optimization Approach

Other solution approaches for the BNMPC optimization are available. Most often a sequential approach is implemented in which the control vector is finitely parameterized in $\tilde{u}_{k,i}(t,\mathbf{p}_{k,i})$ and for each evaluation of the cost function the state trajectories are evaluated for the entire prediction horizon by numerically integrating the model equations using the current guess of the input parametrization vector from the optimizer (see Figure 15.5). This approach sequentially performs the numerical integration and optimization steps, which leads to feasible state trajectories in each optimization step.

15.3.3.3 Simultaneous Optimization Approach

An alternative optimization approach solves the optimization problem and model differential equations simultaneously. The differential equations are discretized, $x_{k+1} = f_k(x_k,u_k;\theta)$, and included in the optimization problem as additional constraints. Usually this approach uses collocation methods for the discretization of the differential equations. While the resulting NLP is very large, its significant sparsity can be exploited to increase numerical efficiency.

15.3.4 Real-Time NMPC Algorithm

The solution of the problem P_2 requires a certain, usually not negligible, amount of computation time δ_k, while the process evolves to a different state. In this case, the optimal feedback control $u^\star(t_k) = [u_{0|t_k}, u_{1|t_k}, \ldots, u_{N_p|t_k}]$ computed at moment t_k, corresponding to the information available up to this moment, will no longer be optimal. Computational delay δ_k must be taken into consideration in real-time applications of NMPC. An approach to reduce the effect of computational delay on the closed-loop dynamics is to, at moment t_k, first implement the control input $u_{1|t_{k-1}}$ from the second stage of the previous optimization problem into the process, then start the numerical solution of the current

optimization problem with fixed $u_{0|t_k} = u_{1|t_{k-1}}$. After completion, the numerical optimization algorithm idles for the remaining period of $t \in (t_k + \delta_k, t_{k+1})$, and then at the beginning of the next stage, at moment $t_{k+1} = t_k + \Delta t$, $u_{1|t_k}$ is introduced into the process, and the algorithm is repeated. This approach requires real-time feasibility for the solution of each open-loop optimization problems ($\delta_k \leq \Delta t$).

The initial values embedding strategy can significantly enhance computational performance. The approach is based on the fact that optimization problems at subsequent sampling times differ only by the initial values that are imposed through the initial value constraints. Accepting an initial violation of these constraints, the solution trajectory of the previous optimization problem can be used as an initial guess for the current problem. Since in the direct multiple-shooting approach, the decision variables include both the control input and the initial values of the states and the discretization points, for this approach to work efficiently, the entire decision vector (control plus states) has to be initialized with the solution of the previous optimization. All derivatives and an approximation of the Hessian matrix, which are already available for the solution trajectory from the previous step, can be used in the new optimization. This enables the solution of the first quadratic program in the SQP solution to be performed without any additional solution of the differential equations.

15.3.5 Robust End-Point BNMPC Formulations

While applying feedback via repeated optimization BNMPC is inherently more robust than the open-loop optimization-based control approaches, incorporating a robustness term in the formulation of the online optimization problem can significantly enhance the robust performance of the BNMPC scheme. Consider the case of parameter uncertainty, with $\delta\theta \in \mathbb{R}^{n_\theta}$ defined as the perturbation about the nominal parameter vector $\hat{\theta}$. The real uncertain parameter vector is then given by $\theta = \hat{\theta} + \delta\theta$. Assuming zero mean, normal measurement errors, and known covariance matrix, the set of possible parameter values is given by the hyperellipsoidal confidence region, defined as

$$\Theta(\alpha) \triangleq \{\theta : (\theta - \hat{\theta})^T \mathbf{V}_{\boldsymbol{\theta}}^{-1} (\theta - \hat{\theta}) \leq \chi^2_{n_\theta}(\alpha)\}, \tag{15.16}$$

where α is the confidence level, $\chi^2_{n_\theta}(\alpha)$ is a quantile of the chi-squared distribution with n_θ degrees of freedom, and $\mathbf{V}_{\boldsymbol{\theta}} \in \mathbb{R}^{n_\theta \times n_\theta}$ is the parameter covariance matrix. Uncertainty description (Equation 15.16) is the most commonly produced output by least-squares identification procedures from experimental data.

Several *robust optimization* approaches can be used to incorporate parametric uncertainties in the BNMPC optimization problem. One approach is to minimize a worst-case objective (this is often referred to as the *minmax approach*), such as

$$\mathcal{H} = \max_{\theta \in \Theta} \psi(x(t_f); \theta), \tag{15.17}$$

where $\psi(x(t_f); \theta)$ is the end-point property of interest. Although efficient techniques have been developed to solve the computationally demanding minmax optimization, the worst-case value usually has a very low probability of occurring so that poor performance is obtained for more representative parameter values (such as the nominal case). Different approaches have been proposed to avoid this drawback of the minmax technique, by instead formulating the optimal control problem in terms of providing a compromise between two conflicting objectives representing the performance and robustness terms. The resulted robust optimization can be implemented by formulating a weighted sum objective of the performance and robustness terms or by including the performance index only in the objective function and adding the robustness term as a constraint to the optimization.

The *mean-variance approach* uses the objective function

$$\mathcal{H} = (1 - w)\mathcal{E}[\psi(x(t_f); \theta)] + wV_\psi(t_f), \tag{15.18}$$

to account for parameter uncertainties, where $\mathcal{E}$ and $V_\psi \in \mathbb{R}$ are the expected value and variance, respectively, of the property at the end of the batch, and $w \in [0, 1]$ is a weighting coefficient that quantifies

the tradeoff between nominal and robust performance. The main advantage of this approach compared to the classical minmax optimization is that the tradeoff between nominal and robust performance can be directly specified by selecting the weight. The expected value and variance can be estimated efficiently using a second-order power-series expansion,

$$\delta\psi = L\delta\theta + \frac{1}{2}\delta\theta^T \mathbf{M}\delta\theta + \cdots \tag{15.19}$$

where $L = (\mathrm{d}\psi/\mathrm{d}\theta)_{\hat{\theta},u} \in \mathbb{R}^{n_\theta}$, and $\mathbf{M} = (\mathrm{d}^2\psi/\mathrm{d}\theta^2)_{\hat{\theta},u} \in \mathbb{R}^{n_\theta \times n_\theta}$ are the first- and second-order sensitivities, respectively. Assuming zero-mean normally distributed parameters $\delta\theta$, the expected value and variance of $\delta\psi$ based on Equation 15.19 are given by the analytical expressions

$$\mathcal{E}[\delta\psi] = \frac{1}{2} tr(\mathbf{M}\mathbf{V}_\theta), \tag{15.20}$$

$$V_\psi = L\mathbf{V}_\theta L^T + \frac{1}{2} tr[(\mathbf{M}\mathbf{V}_\theta)]^2, \tag{15.21}$$

where $tr(\mathbf{A})$ is the trace of the matrix $\mathbf{A}$. The feasibility of the optimization under parametric uncertainty can be assessed by reformulating the constraints in a probabilistic sense:

$$\mathbb{P}(h_i(x, u; \theta) \leq 0) \geq \alpha_i, \tag{15.22}$$

where $\mathbb{P}$ is the probability and α_i is the desired confidence level for the satisfaction of constraint i. The robust formulation of Equation 15.22 can be written using the t-test in the form

$$\mathcal{E}[h_i] + t_{\alpha/2,n_\theta}\sqrt{V_{h_i}} \leq 0, \quad i = 1, \ldots, c. \tag{15.23}$$

The expected value ($\mathcal{E}[h_i]$) and covariance (V_{h_i}) of the constraint h_i can be evaluated using first- or second-order approximations. For the first-order approximation $\mathcal{E}[h_i(x, u; \theta)] = h_i(x, u; \hat{\theta})$ and $V_{h_i} = L_{h_i}\mathbf{V}_\theta L_{h_i}^T$, whereas expressions similar to Equations 15.20 and 15.21 can be used for the second-order approximation, with $L_{h_i} = (dh_i/d\theta)_{\hat{\theta},u} \in \mathbb{R}^{n_\theta}$, and $\mathbf{M}_{h_i} = (d^2h_i/d\theta^2)_{\hat{\theta},u} \in \mathbb{R}^{n_\theta \times n_\theta}$. In this formulation, the algorithm shows robust performance in the sense of constraint satisfaction and decreased variance of the performance index.

15.3.6 State Estimation

Accurate state estimation is critical for the success of a BNMPC application. Among the various state estimation approaches, the EKF and moving horizon estimation (MHE) are the most widely used and are described below. See "For Further Information" for references on *particle filtering*, which is a class of state estimation methods that has been applied in recent years.

15.3.6.1 Parameter Adaptive EKF

The EKF has been widely used in control applications, but its performance strongly depends on the model accuracy. One approach to enhance the robust performance of a nonlinear model predictive controller is to use a robust formulation as discussed in the previous section. Alternatively (or in combination) the online adaptation of the model parameters can significantly enhance the robustness. To avoid highly biased model predictions, some of the model parameters are estimated together with the states, leading to a *parameter adaptive EKF* formulation. Define $\theta' \subseteq \theta$ as the subset of the estimated parameters from the parameter vector, and $\theta'' \triangleq \theta \backslash \theta'$ as the set of the remaining parameters. The augmented state vector

in this case is given by $\mathcal{X} = [x, \theta']^T$ and the augmented model used for estimation is given by

$$\dot{\mathcal{X}} = [f(\mathcal{X}, u;\, \theta''), 0]^T + [w, w_{\theta'}]^T, \tag{15.24}$$

where w, and $w_{\theta'}$ are zero-mean Gaussian white-noise variables. The time-varying state matrix of the locally linearized augmented model is defined by

$$\mathbf{A}(t_k) = \begin{bmatrix} \dfrac{\partial f(\mathcal{X}(t_k), u(t_k); \theta'')}{\partial x} & \dfrac{\partial f(\mathcal{X}(t_k), u(t_k); \theta'')}{\partial \theta'} \\ 0 & 0 \end{bmatrix}. \tag{15.25}$$

The time update of the states and state covariance are propagated by numerically integrating the model equations together with the covariance propagation equation for one sampling time ($t \in [t_{k-1}, t_k]$):

$$\dot{\mathcal{X}} = [f(\mathcal{X}, u; \theta''), 0]^T, \tag{15.26}$$

$$\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T(t) + \mathbf{Q}(t), \tag{15.27}$$

with $t \in [t_{k-1}, t_k]$ and initial conditions $\hat{\mathcal{X}}(t_{k-1})$ and $\mathbf{P}(t_{k-1})$ obtained from the last estimation, and the Jacobian $\mathbf{A}$ given by Equation 15.25. Define the solutions of Equations 15.26 and 15.27 as $\hat{\mathcal{X}}^-(t_k)$ and $\mathbf{P}^-(t_k)$, respectively. With these values the Kalman gain $\mathbf{K}$ is computed, and then the measurement update stage is performed according to

$$\mathbf{K}(t_k) = \mathbf{P}^-(t_k)\tilde{\mathbf{C}}^T(t_k)(\tilde{\mathbf{C}}(t_k)\mathbf{P}^-(t_k)\tilde{\mathbf{C}}^T(t_k) + \mathbf{R})^{-1}, \tag{15.28}$$

$$\mathbf{P}(t_k) = (\mathbf{I} - \mathbf{K}(t_k)\tilde{\mathbf{C}}(t_k))\mathbf{P}^-(t_k), \tag{15.29}$$

$$F_K(t_k) = \mathbf{K}(t_k)(y_m(t_k) - g(\hat{\mathcal{X}}^-(t_k), u(t_k); \hat{\theta}'')), \tag{15.30}$$

$$\hat{\mathcal{X}}(t_k) = \hat{\mathcal{X}}^-(t_k) + F_K(t_k), \tag{15.31}$$

where $y_m(t_k)$ corresponds to the measurements obtained from the real process at time t_k, $F_K(t_k)$ is the Kalman filter correction factor, and $\tilde{\mathbf{C}}$ is the Jacobian of the measurement equations with respect to the augmented states:

$$\tilde{\mathbf{C}}(t_k) = \left(\frac{\partial g(\mathcal{X}(t), u(t); \hat{\theta}'')}{\partial \mathcal{X}} \right)_{\hat{\mathcal{X}}(t_k), u(t_k)}. \tag{15.32}$$

The estimated states in Equation 15.31 are used as the initial value for the model prediction stage in the optimization algorithm.

The measurement covariance matrix is determined based on the accuracy of the measurements. The appropriate choice of the state covariance matrix, $\mathbf{Q}$, is however often challenging in practical applications. An estimate of $\mathbf{Q}$ can be obtained by assuming that the process noise vector mostly represents the effects of parametric uncertainty. Based on this assumption and performing a first-order power-series expansion of the model error equations using the nominal parameter vector and control trajectory, the process noise covariance matrix can be computed from

$$\mathbf{Q}(t) = \mathbf{S}_\theta(t)\mathbf{V}_\theta\mathbf{S}_\theta^T(t), \tag{15.33}$$

where $\mathbf{V}_\theta \in \mathbb{R}^{n_\theta \times n_\theta}$ is the parameter covariance matrix, and $\mathbf{S}_\theta(t)$ is the Jacobian computed using the nominal parameters and estimated states:

$$\mathbf{S}_\theta(t) = \left(\frac{\partial f}{\partial \theta} \right)_{\hat{x}(t), u(t), \hat{\theta}}. \tag{15.34}$$

Equation 15.33 provides an easily implementable way to estimate the process noise covariance matrix, since the parameter covariance matrix $\mathbf{V}_\theta$ is usually available from parameter estimation, and the sensitivity coefficients in $\mathbf{S}_\theta(t)$ can be computed by finite differences or via sensitivity equations. This approach leads to a time-varying full covariance matrix, which has been shown to provide better estimation performance for batch processes than the classically used constant diagonal $\mathbf{Q}$.

15.3.6.2 Moving Horizon Estimation

MHE uses a moving and usually fixed-size window of previous model predictions and process measurements. As an additional measurement arrives, the oldest measurement is discarded and the model is updated with the new information. The MHE formulation is based on the idea of penalizing the deviations between the measurement data and predicted outputs. In addition—for theoretical reasons—generally a regularization term on the initial state estimate is added to the objective function in the case of continuous processes. However, for batch systems where the states usually change significantly this term is not well defined. The advantage of MHE in the context of BNMPC is that the problem formulations are similar for both the estimation and control, requiring relatively small additional effort to implement the estimation. The estimation problem for the parameter adaptive MHE to be solved in every time step t_k is

$$\min_{\theta', \hat{x}(t_k)} \sum_{i=k-N_{est}}^{k} \left\| y_i - y_i^{meas} \right\|_{Q_{est}}^2, \tag{15.35}$$

subject to

$$\dot{x}(t) = f(x(t), u(t); \theta), \tag{15.36}$$

$$y(t) = g(x(t), u(t); \theta), \tag{15.37}$$

$$x_{\min} \leq x(t) \leq x_{\max}, \tag{15.38}$$

$$\theta_{\min} \leq \theta' \leq \theta_{\max}, \tag{15.39}$$

where y_i^{meas} are the process output measurements, Q_{est} is the weighting matrix that can incorporate a forgetting factor, N_{est} is the number of past samples used in the estimation window, $x_{\min}$ and $x_{\max}$ are the minimum and maximum bounds on the state variables, and $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum bounds on the estimated parameters θ'.

MHE is usually desirable when constraints are present, for strongly nonlinear models, or when measurements are available infrequently and at various sampling periods. All these features can be easily incorporated in the MHE formulation. However, the computational requirement of MHE is generally significantly higher than for EKF, and obtaining reliable real-time solutions may be the limiting factor in the practical applications of MHE.

15.4 Implementation Aspects of Batch NMPC in an Industrial Environment

The practical implementation of BNMPC is considerably more challenging than those associated with linear MPC applications. Model validation, reliability of state estimation, and effects of model/plant mismatch should be addressed before a BNMPC approach is implemented. Several key issues, which can lead to difficulties in practical implementation of NMPC in general and BNMPC techniques in particular are discussed below.

15.4.1 Efficient Development and Identification of Control-Relevant Model

A significant amount of time and expense is usually attributed to the modeling and system identification step in the design and installation of BNMPC. The manageable complexity of the developed model is very important for a BNMPC application. Although a detailed model may lead to better performance in principle, often practical situations require a compromise to prevent the effort for model building and computational cost involved in the resulting optimization from becoming prohibitively large. This is one of the reasons why most of the industrial BNMPC products have used empirical models identified

through plant tests, although first-principles models have better extrapolative capability and provide the most insight about the process. These benefits and continual increases in the computational power of control hardware suggest that first-principles models will be increasingly used over time in BNMPC applications.

15.4.2 Measurement-Based BNMPC

One of the most important questions that need to be discussed in the initial phase of the design of any practical BNMPC application is related to measurements which are available versus variables which are needed for the model. Often the answer to this question determines the type of BNMPC approach (whether STNMPC or SONMPC) to use. Besides the modeling difficulty and computational complexity attached to large-scale models, including many details into the model can lead to a large number of states that results in unobservable models based on the available measurements. The control-relevant model always has to be determined in conjunction with the observer design. One of the major bottlenecks in the development of more first principles-based BNMPC applications is the lack of proper sensor technology or the availability of state-of-the-art sensors in industrial plants. Due to observability problems with the available sensors, often empirical input–output models and simplified first-principles models have been used in industrial applications to reduce the computational requirements. Fortunately, many advances in sensor technology and computer hardware have occurred in recent years. New software and hardware sensors can provide comprehensive information about the processes in a fast and reliable way. These developments have made many more BNMPC applications feasible.

15.4.3 Model Identification

Many industrial BNMPC implementations use existing modeling software packages (e.g., HYSYS, gPROMS) rather than building and simulating the models from scratch. Although using existing models developed in such high-level programming environments initially may seem alluring, often these models are not appropriate for control purposes. Such models can serve as the starting point for the identification of a control-relevant model, but often have to be enhanced with additional data to improve the accuracy of the model in describing the process dynamics.

Optimal experimental design involves the determination of manipulated variables for experimental data collection to be used for model identification. Experimental design objectives can be formulated to minimize model uncertainties, increase plant friendliness, and/or maximize control relevance. Model identification refers not only to the derivation of model parameters but also to the identification of a proper model structure. In BNMPC applications, the model structures should be identified that capture the process behavior and are amenable to optimization-based control. To achieve this, model reduction techniques relevant to BNMPC can be useful in connection with low-order physical modeling approaches. Hybrid models that efficiently combine fundamental and empirical modeling techniques can be useful, as will be illustrated in Section 15.5.

15.4.4 Reliable and Fast Solution of the Online Optimization

The use of an efficient optimization approach is crucial in the real-time feasibility of BNMPC applications. End-point BNMPC is fundamentally different from classical NMPC problems, owing to its shrinking prediction horizon during the batch. On the one hand, this feature leads to progressively decreasing computational demand, but on the other hand, the prediction horizon must be until the end of the batch from the beginning, with fine time discretization of the control actions due to the transient nature of the process. The computational performance of BNMPC algorithms can be enhanced by applying algorithm engineering approaches to the tailored solution of the typical problems that arise in BNMPC. For example, hierarchical solution approaches that exploit problem structure can be applied. CPU time can be used more efficiently, for example, when convergence is achieved before the end of the sampling

time, the surplus time can be used to precompute some parts of the next step. Understanding how various model attributes contribute to particular features of the corresponding optimization (e.g., whether the error surface is nonsmooth) can help in the more efficient solution of the optimization.

Managing the online computational load is not the only concern related to practical BNMPC applications. Another important problem is related to the robustness of the optimization. A backup strategy in case of failures of the main BNMPC controller due to convergence problems in the optimization or dynamic simulation of the model should be incorporated in practical BNMPC implementations. The most straightforward but suboptimal approach is to use the last control input or the control input computed in previous sampling times corresponding to the current period, and then to let the lower-level controller act until the system is reset.

15.4.5 Long-Term Maintenance and Support of the BNMPC Algorithm in an Industrial Application

The implementation complexity of BNMPC approaches is high. Therefore it is important to assess how often long-term maintenance can be performed and assess the limits of the approaches in face of changing process and operating conditions. Suitable support tools for long-term on-site maintenance should be developed in parallel with the controller design and implementation. Such tools should be able to perform performance assessment, model identification, and controller tuning.

The complexity of the models also plays a role in the support and maintenance issues. A very complex model will probably require important effort in the development phase but can provide more flexibility during long-term maintenance. Reduced or empirical models can be more rigid in the face of changing conditions, hence requiring more significant maintenance effort.

15.5 Setpoint Tracking Batch NMPC of an Industrial Reactor

The first example applies BNMPC algorithm for the setpoint tracking control for an industrial pilot-scale polymerization reactor. This case study represents a very typical control problem for industrial batch reactors, considering the control objective, availability of sensors, and structure of the temperature control system. A schematic of the experimental pilot plant is in Figure 15.6. The reactor temperature is controlled using a heating–cooling system that is widely used for low-volume industrial batch reactors. The heating–cooling system is based on a closed oil circuit, which is recycled through the jacket with a constant flow rate F_j. The heating–cooling medium goes through a multitubular heat exchanger where the proportional-integral (PI) controller DTC adjusts the cooling water flow rate with the aim to keep the temperature difference between the input and the output of the heat exchanger constant. Heating is performed using an electric heater. The power of the heater is adjusted by a lower-level PI controller TC that controls the input temperature into the jacket. The setpoint of the PI controller TC is determined by the higher-level STNMPC that has the objective to track a predetermined temperature profile in the reactor.

A detailed first-principles model of the process containing material and energy balances as well as kinetic and thermodynamic models was developed and identified from offline experiments. Since only temperature measurement is available inside the chemical reactor, many states of the detailed process model are not estimable, or not even detectable. The detailed process model was used to determine the initial optimal temperature profile and for deriving a reduced-order control-relevant model. Available measurements associated with the reactor were its internal temperature (T_r) and the input and output temperatures into and from the jacket ($T_{j,in}$, T_j). With this set of measurements, the reduced-order model,

$$\dot{n}_M = -Q_r/\Delta H_r \tag{15.40}$$

$$\dot{T}_{r,k} = \frac{Q_r + U_w A_w (T_{w,k} - T_{r,k}) - (UA)_{loss,r}(T_{r,k} - T_{amb})}{m_M c_{p,M} + m_P c_{p,P} + m_{water} c_{p,water}}, \tag{15.41}$$

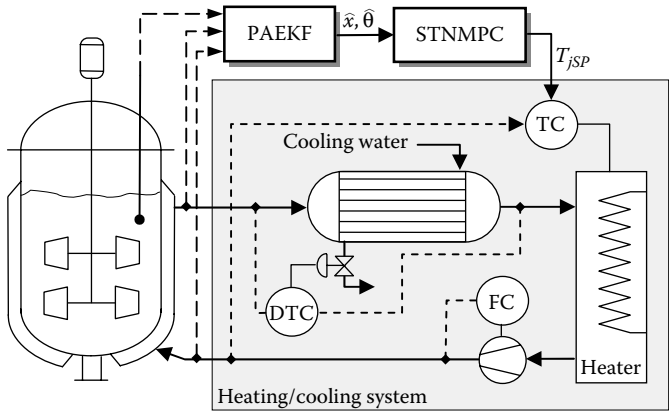

FIGURE 15.6 Schematic representation of the batch polymerization reactor with the heating/cooling system. (Reprinted from Nagy, Z.K., et al. *Control Eng. Pract.*, 15:839–859, 2007. With permission.) PAEKF: parameter adaptive Extended Kalman filter; DTC: differential temperature controller, FC: flow controller; and TC: temperature controller.

$$\dot{T}_{w,k} = \frac{U_j A_j (T_{j,k} - T_{w,k}) - U_w A_w (T_{w,k} - T_{r,k})}{m_w c_{pw}}, \tag{15.42}$$

$$\dot{T}_{j,k} = \frac{\mathcal{N} F_j \rho_j c_{pj} (T_{j,k-1} - T_{j,k}) - U_j A_j (T_{j,k} - T_{w,k}) - (UA)_{loss\,j} (T_{j,k} - T_{amb})}{m_j c_{p,j}}, \tag{15.43}$$

was used in the STNMPC, where $k = 1, \ldots, \mathcal{N}$, $T_r = T_{r,\mathcal{N}}$, $T_j = T_{j,\mathcal{N}}$, $T_{j,0} = T_{j,in}$, n_M is the number of moles of monomer, ΔH_r is the enthalpy of reaction, T_w is the wall temperature, U and A are heat transfer coefficients and areas from reactor-to-wall $(\cdot)_w$ or wall-to-jacket $(\cdot)_j$, $c_{p,M/P/water/w/j}$ and $m_{M/P/water/w/j}$ are the heat capacities and masses of monomer, polymer, water, wall, and oil, T_{amb} is the ambient temperature, ρ_j is the oil density, and $(UA)_{loss,r/j}$ are heat loss coefficients in the reactor and the jacket, respectively. To estimate the transport delay, the reactor, the wall, and the jacket were divided into $\mathcal{N} = 4$ elements, leading to a system of 13 differential equations. For high-performance temperature control, the estimation of the generated heat Q_r is important, and an empirical nonlinear relation $Q_r = f_Q(n_M, T_r)$ was determined from the detailed first-principles model, simulating the process for different temperature profiles. Maximum-likelihood estimation was used to fit the parameters ($\theta = [(UA)_{loss,r}, (UA)_{loss,j}, U_j A_j, m_w, U_w A_w, m_j]$) of the model (Equations 15.40 through 15.43) to the experimental data collected from the industrial pilot plant for several batches of water (for which $Q_r = 0$). This procedure gave the optimal nominal parameter estimates, $\hat{\theta}^\star$, and the corresponding uncertainty description given by the covariance matrix, estimated from the Hessian of the maximum-likelihood objective function at the optimal parameter estimate, $\mathbf{V}_\theta \approx H^{\star^{-1}} = (\partial^2 \psi / \partial \theta^2)^{-1}_{\theta = \hat{\theta}^\star}$, which was then used to initialize the state covariance matrix in the EKF according to Equation 15.33. In the implementations, the parameters $\theta' = [Q_r, U_w A_w]$ were estimated together with the model states in the parameter adaptive EKF to provide an adaptation of the model due to the changing conditions during polymerization. The adaptation of the model parameters was not only important to capture their variations in the real process, but also reduced the effects of model plant/mismatch for offset-free state estimation. Model (Equations 15.40 through 15.43) was used in the STNMPC algorithm, where the objective was to provide a tight setpoint tracking by solving online, at each sampling time k, the optimization problem:

$$\min_{u(t)} \int_{t_k}^{t_k^F} \{(T_r(t) - T_{r,SP}(t))^2 + Q_{\dot{u}} (\dot{u}(t))^2\} \, \mathrm{d}t, \tag{15.44}$$

subject to

$$\begin{aligned} u_{\min} &\le u(t) \le u_{\max}, \\ \dot{u}_{\min} &\le \dot{u}(t) \le \dot{u}_{\max}, \\ 0\% &\le u_{PI}(t) \le 100\%, \end{aligned} \tag{15.45}$$

where the setpoint profile $T_{r,SP}$ used in the controller was given by a previously used benchmark recipe. The second term in Equation 15.44 is a regularization term, which gives a smoother input by minimizing time variation in the control input ($\dot{u}$). The tradeoff between smooth control input and quick control response is tailored by choosing the weighting coefficient $Q_{\dot{u}}$. The manipulated input of the BNMPC, $u(t) = T_{j,in}^{SP}$, was the setpoint temperature to the lower-level PI controller TC in Figure 15.6, which controlled the input temperature into the jacket. The communication between the real industrial plant and the BNMPC algorithm was performed via the OPC automation standard. The lower-level PI controller was included in the model used by the BNMPC algorithm, with the bounds on the output signal of the PI controller (u_{PI}) incorporated as additional constraints in the online optimization problem (last term of (45)) to be able to predict its saturation, which is an important source of nonlinearity in the system. More generally, the dynamics of and constraints imposed by lower-level control systems should be investigated during the design of upper-level control systems, to assess whether this information must be incorporated.

A weighting coefficient of $Q_{\dot{u}} = 0.4$ and prediction and control horizons of 8000 s were used in the optimization with a sampling interval of 20 s. The manipulated variable was discretized in 400 piecewise-constant inputs, leading to a reasonably high-dimensional optimization problem. The multiple shooting approach with real-time iteration and initial value embedding guarantees the real-time feasibility of the STNMPC implementation. Even with the large control discretization of 400, the computation time was about 5 s, which was below the sampling time of 20 s. The experimental results shown in Figure 15.7 show the excellent control performance achieved with the STNMPC. The achieved tracking performance was significantly better compared to cascade PI control (Figure 15.8), which is the standard control architecture for these types of control problems. The maximum control error decreased from ~4.5°C in the case of cascade PI control to below 0.5°C for the STNMPC approach.

15.6 Hierarchical BNMPC for Simultaneous Setpoint Tracking and Optimization

Simulation studies have also been performed in which STNMPC was combined with a higher-level SONMPC as shown in the schematic in Figure 15.3. In STNMPC, a similar objective was used as in Equation 15.44. The temperature control for the large-scale plant was implemented through a split valve, which regulated the ratio of cooling water F_w and heating medium (steam, F_s), respectively, which is another heating–cooling system configuration widely used for larger-scale industrial batch reactors. The optimized manipulated variable was the input to the split valve represented by $u(t) \in [0, 1]$, and the flow rates were given based on the minimum and maximum values as

$$F_w = (F_{w,\max} - F_{w,\min})u + F_{w,\min}, \tag{15.46}$$

$$F_s = (F_{s,\min} - F_{s,\max})u + F_{s,\max}. \tag{15.47}$$

For proprietary reasons, fictitious kinetic parameters are used in the simulations presented next, and the product property indicators are not revealed. However, the qualitative conclusions are consistent with the results obtained with the real parameters.

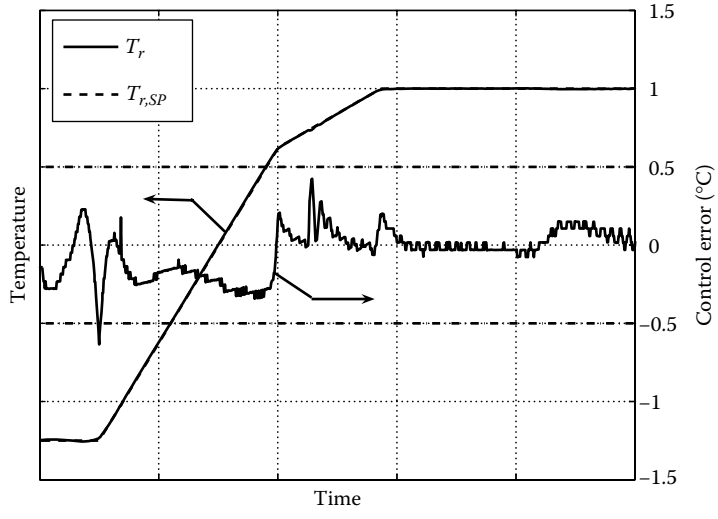

FIGURE 15.7 Experimental data showing the setpoint tracking performance of BNMPC in an industrial polymerization reactor. The reactor temperature follows very tightly the setpoint (within ±0.5°C). (Reprinted from Nagy, Z.K., et al. *Control Eng. Pract.*, 15:839–859, 2007. With permission.)

The setpoint optimizing controller (SONMPC) computed the optimum reactor temperature $T_r^{\star}(t)$ used as the reference temperature $T_{r,SP}(t)$ in the lower-level setpoint tracking controller (STNMPC), by repeatedly solving the end-point optimization

$$T_{r,SP}(t) = T_r^{\star}(t) = \underset{T_r(t)}{\arg\min} \left(\mathcal{Q}(t_f) - \mathcal{Q}_{recipe}\right)^2 \tag{15.48}$$

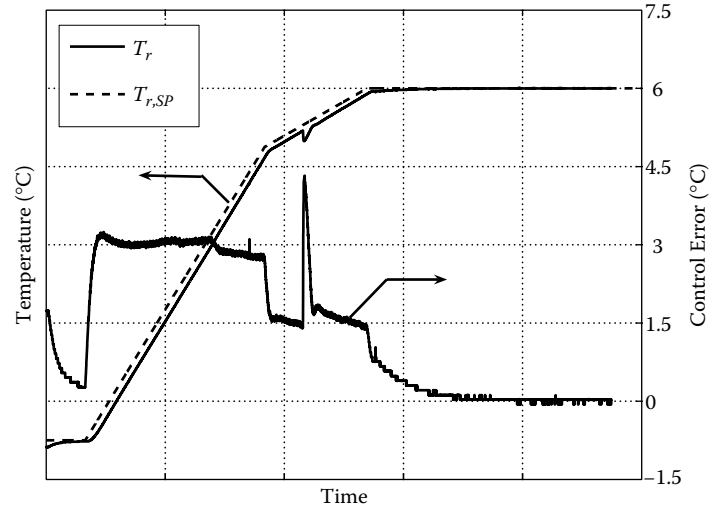

FIGURE 15.8 Experimental data showing the setpoint tracking performance of cascade PI control in an industrial polymerization reactor (sampling interval of 3 s). (Reprinted from Nagy, Z.K., et al. *Control Eng. Pract.*, 15:839–859, 2007. With permission.)

with the objective to minimize the deviation of the final product quality $\mathcal{Q}(t_f)$ from the desired value $\mathcal{Q}_{recipe}$ in a fixed batch time t_f. This optimization was solved subject to the constraints

$$T_{r,\min} \leq T_r \leq T_{r,\max}, \tag{15.49}$$

$$\dot{T}_{r,\min} \leq \dot{T}_r \leq \dot{T}_{r,\max}, \tag{15.50}$$

$$\mathcal{P}_{\min} \leq \mathcal{P}(t_f), \tag{15.51}$$

where $T_{r,\min}$, $T_{r,\max}$, $\dot{T}_{r,\min}$, and $\dot{T}_{r,\min}$ are the minimum and maximum temperatures and temperature changes, respectively, and the constraint (Equation 15.51) ensures that a minimum productivity will be met at the end of the batch, with $\mathcal{P}$ being the productivity measure (such as concentration or conversion). The STNMPC and SONMPC optimizations are also subject to the model equations. Different models are used in the two BNMPC formulations. The model used in the upper-level SONMPC consists of the detailed chemical kinetics, material balances, and differential equations required for computing the product properties. Energy balances were not included since the reactor temperature is directly optimized at this level. The model used in the lower-level STNMPC contained detailed energy balances, but no equations for the product properties. The solution of both STNMPC and SONMPC was performed using the direct multiple shooting algorithm.

The hierarchical BNMPC setup was tested in a disturbance scenario with 10% error in all kinetic parameters. A parameter adaptive EKF was used to estimate the kinetic parameters and the unmeasured states. The system states were augmented with the adapted parameters and differential equations of the form $\dot{\theta} = 0$. In the EKF, a third more complicated model was used that contained all equations required to estimate the states needed for both BNMPCs. Reactor temperature, jacket inlet and outlet temperatures, concentrations of raw material and product, and two of the states required to compute the property indicators were considered as measured variables. With this set of measurements, the augmented system was observable and the EKF quickly converged. A sampling interval of 30 s and control and prediction horizons of $N_p = 10$ were used in the STNMPC. In the SONMPC, a discretization of $N = 100$ intervals was used, which results in a sampling interval of 5 min. A sampling time of 3 s was used in the EKF.

Figure 15.9 shows the initial optimal temperature profile (corresponding to the model parameters with 10% error), and the final temperature setpoint obtained in the final iteration of the SONMPC,

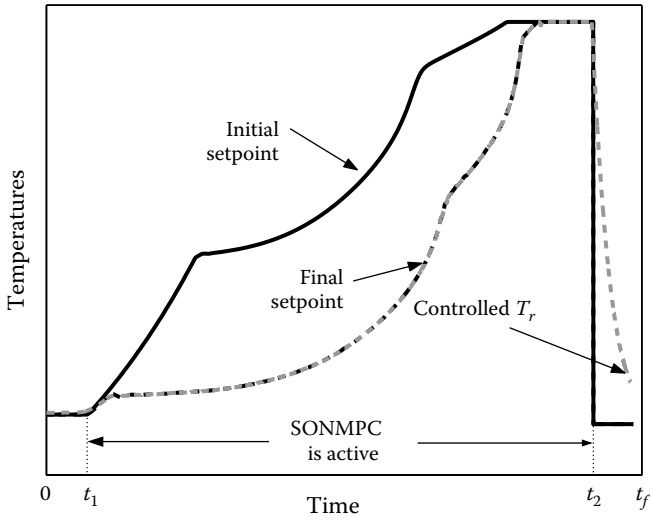

FIGURE 15.9 Initial and final setpoint temperature trajectories obtained by SONMPC, and the controlled reactor temperature obtained with the hierarchical BNMPC algorithm.

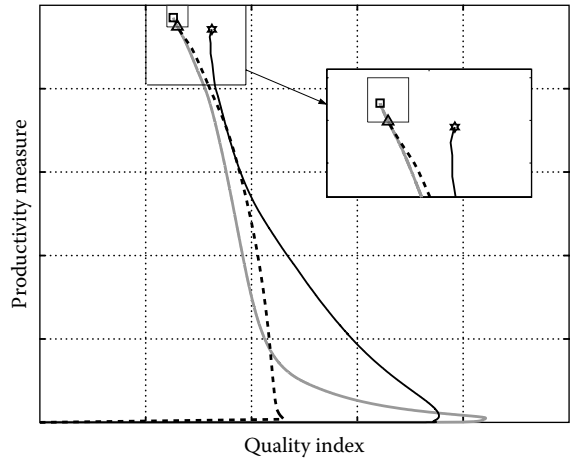

FIGURE 15.10 State-space plots of the variation in the quality index ($\mathcal{Q}$) and productivity measure ($\mathcal{P}$) during the batch. The thick solid line shows the evolution of the variables for the initial setpoint trajectory, if the initial model parameters were exactly known. The dashed black line shows the evolution using the hierarchical BNMPC algorithm applied to the model that initially has 10% error in the parameters. The thin black line shows the evolution when the SONMPC is turned off and the STNMPC follows the initial setpoint trajectory obtained with 10% error in the model parameters. The inset shows in more detail where the end-point properties in the three cases (represented by the square, triangle, and star, respectively) are compared to the desired range (given by the rectangle).

indicating that the optimal temperature profile is very sensitive to parameter variations for this process. The controlled reactor temperature obtained from the STNMPC using the most recently computed setpoint trajectory from the SONMPC for $t \in [t_1, t_2]$ indicates that the hierarchical BNMPC algorithm very closely tracks the temperature trajectory that is optimal for the perturbed model parameters. Figure 15.10 shows the variation along the batch of two key variables, related to productivity and quality, for different scenarios. For the hypothetical case where the initial model parameters were exactly known, the initial setpoint trajectory in Figure 15.9 would be followed, resulting in $\mathcal{P}(t_f)$ and $\mathcal{Q}(t_f)$ in the feasible region. With 10% initial error in the model parameters and SONMPC turned off, STNMPC tracks the initial setpoint trajectory, and the final product violates both the productivity and quality requirements. For the hierarchical BNMPC algorithm, the setpoint trajectory is adapted, which results in a feasible batch. Figure 15.11 shows the manipulated variables (cooling and heating flow rates) for the controlled temperature trajectory obtained by applying the two-level hierarchical BNMPC approach. The CPU times required to solve the optimization problems in the two BNMPC algorithms are also shown in Figure 15.11. STNMPC required usually 2–4 SQP iterations and less than 3 s to solve one open-loop optimization problem. SONMPC required about 5 min to solve the end-point optimization problem during the initial part of the batch, and the CPU time decreased as the control horizon continuously decreased from the initial $N = 100$ stages and the estimated model parameters reached their converged values.

15.7 Robust End-Point Batch-NMPC for the Crystal Size Distribution Control in Cooling Crystallisation

This section applies end-point batch NMPC to a simulated batch crystallization process. Crystallization from solution is an industrially important unit operation due to its ability to provide high-purity separation. The control of the crystal size distribution (CSD) can be critically important for efficient downstream

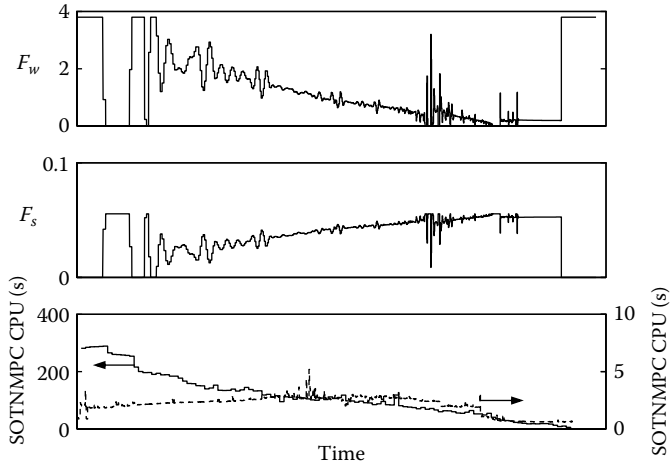

FIGURE 15.11 Manipulated variables and CPU times for the SONMPC and STNMPC controllers (on a DELL Latitude D600 with Intel Pentium M 1400 MHz processor).

operations (such as filtration or drying) and product quality (e.g., bioavailability, tablet stability, dissolution rate). The process is usually subject to significant uncertainties, providing strong incentives for the application of robust control schemes. The model system simulated in this section is the batch cooling crystallization of KNO_3 in water. The process model is

$$x^T = [\mu_0, \ldots, \mu_4, C, \mu_{seed,1}, \ldots, \mu_{seed,3}, T], \tag{15.52}$$

$$f(x,u;\theta) = \begin{bmatrix} B \\ G\mu_0 + Br_0 \\ 2G\mu_1 + Br_0^2 \\ 3G\mu_2 + Br_0^3 \\ 4G\mu_3 + Br_0^4 \\ -\rho_c k_v (3G\mu_2 + Br_0^3) \\ G\mu_{seed,0} \\ 2G\mu_{seed,1} \\ 3G\mu_{seed,2} \\ \dfrac{-UA(T-T_j) - 3\Delta H_c(C)\rho_c k_v G\mu_2 m_s}{(\rho_c k_v \mu_3 + C + 1)m_s c_p(C)} \end{bmatrix}, \tag{15.53}$$

where μ_i is the ith moment ($i = 1, \ldots, 4$) of the total crystal phase and $\mu_{seed,i}$ is the ith moment ($i = 0, \ldots, 3$) corresponding to the crystals grown from seed, r_0 is the size of nucleated crystals, k_v the volumetric shape factor, and ρ_c the density of the crystal, U the heat transfer coefficient, A the heat transfer area, T_j the jacket temperature, m_s the mass of the solvent, $\Delta H_c(C)$ the heat of crystallization which is an empirical function of the solute concentration, and $c_p(C)$ the heat capacity of the slurry. The crystal growth rate G and the nucleation rate B are

$$G = k_g S^g, \tag{15.54}$$

$$B = k_b S^b \mu_3, \tag{15.55}$$

where $S = (C - C_{sat})/C_{sat}$ is the relative supersaturation and $C_{sat} = C_{sat}(T)$ is the saturation concentration, C is the solute concentration, and T is the temperature. The model parameter vector consists of the

kinetic parameters for growth and nucleation:

$$\theta^T = [g, \ln k_g, b, \ln k_b], \tag{15.56}$$

with nominal values

$$\hat{\theta}^T = [1.31, 8.79, 1.84, 17.38]. \tag{15.57}$$

A hyperellipsoidal parameter uncertainty description characterized by a covariance matrix with

$$\mathbf{V}_\theta^{-1} = \begin{bmatrix} 102,873 & -21,960 & -7,509 & 1,445 \\ -21,960 & 4,714 & 1,809 & -354 \\ -7,509 & 1,809 & 24,225 & -5,198 \\ 1,445 & -354 & -5,198 & 1,116 \end{bmatrix} \tag{15.58}$$

was obtained using standard parameter estimation approaches. The manipulated variable can be treated as being the crystallizer temperature ($u(t) = T(t)$), in which case a lower-level feedback controller is used to follow the desired temperature trajectory, or the jacket temperature $u(t) = T_j(t)$. In both cases the inequality constraints used for this system have a similar form. For $u(t) = T(t)$, these constraints are

$$T(t) \in \mathcal{U} = [T_{\min}, T_{\max}], \tag{15.59}$$

$$h(x, u; \theta) = \begin{bmatrix} \dfrac{dT(t)}{dt} - R_{T,\max} \\ -\dfrac{dT(t)}{dt} + R_{T,\min} \\ C(t_f) - C_{f,\max} \end{bmatrix} \leq 0, \tag{15.60}$$

where $R_{T,\min}$ and $R_{T,\max}$ are the minimum and maximum temperature ramp rates, and $C_{f,\max}$ is the maximum solute concentration at the end of the batch that specifies the minimum yield required by economic considerations.

A BNMPC implementation for the batch cooling crystallizer is shown in Figure 15.12. The state vector was estimated using a similar EKF technique as in the previous examples and as described in Section 15.3.6.1, with a time-varying state covariance matrix obtained using a first-order power-series expansion applied to the parameter uncertainty description (Equation 15.58) resulted from the model identification. The measurement vector was $y = [\mu_1, \mu_2, \mu_3, C, T]^T$. The first three variables (moments μ_1, μ_2, and μ_3) can be measured using video microscopy or laser backscattering. Several online techniques are available for the solution concentration measurements (such as conductivity or attenuated total reflection Fourier transform infrared spectroscopy). Temperature measurements are readily available

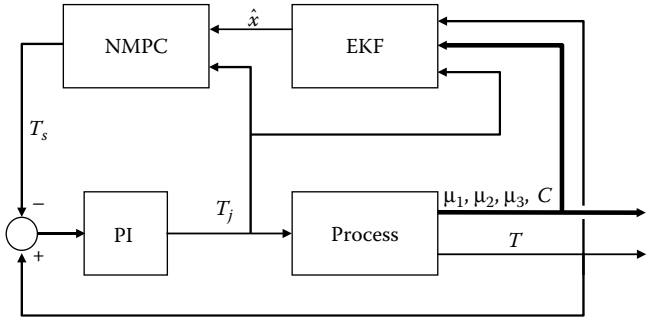

FIGURE 15.12 The BNMPC structure coupling NMPC with the EKF. (Reprinted from Nagy, Z.K. and R.D. Braatz, *AIChE J.*, 49:1776–1786, 2003. With permission.)

using thermocouples. The estimated states were used in the BNMPC optimization, which computed the setpoint on the crystallizer temperature that was sent to a lower-level PI controller that manipulated the jacket temperature to achieve the desired temperature. The temperature profile was described as piecewise-linear trajectories by discretizing the batch time in N equal intervals and considering the temperatures at every time as the optimization variables. The BNMPC with the EKF was implemented in MATLAB® using the sequential optimization approach, in which a stiff differential-equation solver was used in combination with an optimization subroutine (fmincon). The objective function used in the robust BNMPC algorithm was

$$\mathcal{H} = (1-w)\psi(x(t_f);\hat{\theta}) + wV_\psi(t_f) + \lambda \int_{t_k}^{t_f} \|T(t) - T_{\text{nom}}(t)\|_2 \, dt, \tag{15.61}$$

where the CSD property used as the performance index in the objective function was the ratio of the nucleated crystal mass to the seed crystal mass:

$$\psi(x(t_f)) = \frac{\mu_3 - \mu_{\text{seed},3}}{\mu_{\text{seed},3}} \tag{15.62}$$

and $T_{nom}(t)$ was the optimal temperature profile obtained with the nominal parameters. The robustness term V_ψ was obtained using a first-order power series. Simulation results showed that the first-order power-series overestimated the variance of the performance index compared to the more accurate second-order approximation given by Equation 15.19. To provide enhanced robust performance, the more accurate value of V_ψ was not needed and the first-order approximation of V_ψ provided acceptable robust performance at lower computational effort. The inclusion of the last term in the objective function (Equation 15.61) improved the nominal performance. In this particular application, setting the appropriate weight λ resulted in similar nominal performance for larger values of the weighting coefficient on the variance (w) as for the case of BNMPC without the uncertainty term.

The nominal open-loop optimal temperature profile obtained by solving the optimization with the objective function (Equation 15.61) with $w = \lambda = 0$ is shown in Figure 15.13. The nominal open-loop temperature profile practically coincides with the nominal BNMPC results since the BNMPC optimization

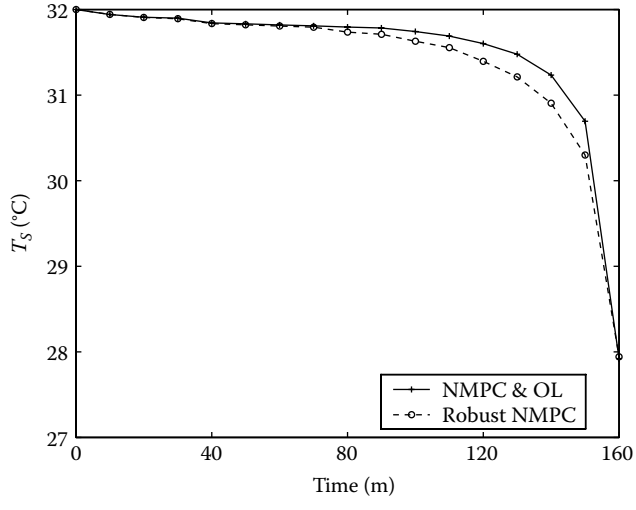

FIGURE 15.13 Optimal temperature profiles for nominal BNMPC ($w = 0$), robust BNMPC ($w = 0.2$), and for open loop (OL) optimal control. (Reprinted from Nagy, Z.K. and R.D. Braatz, *AIChE J.*, 49:1776–1786, 2003. With permission.)

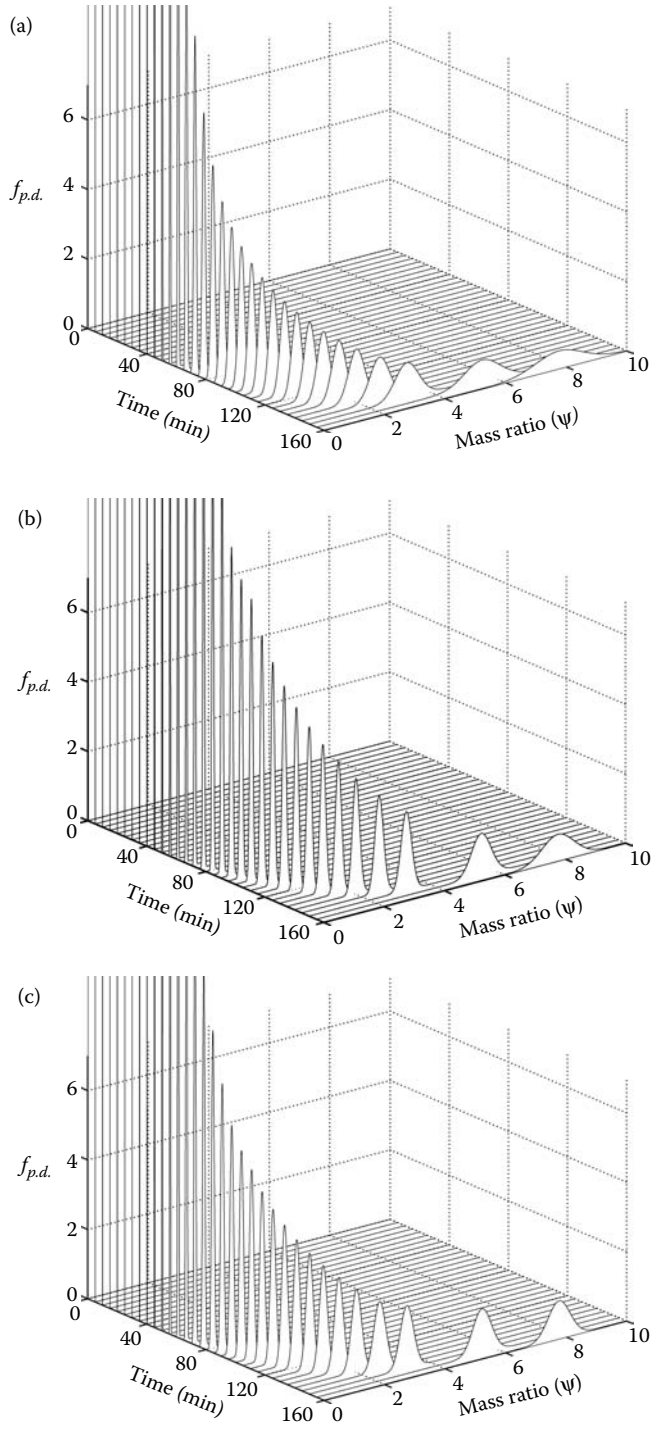

FIGURE 15.14 Variation of the probability distribution function of the performance index (the nucleation to seed mass ratio) during the batch for (a) open-loop optimal control; (b) nominal BNMPC; and (c) robust BNMPC. The probability distribution function was determined using first-order distributional analysis. (Reprinted from Nagy, Z.K. and R.D. Braatz, *AIChE J.*, 49:1776–1786, 2003. With permission.)

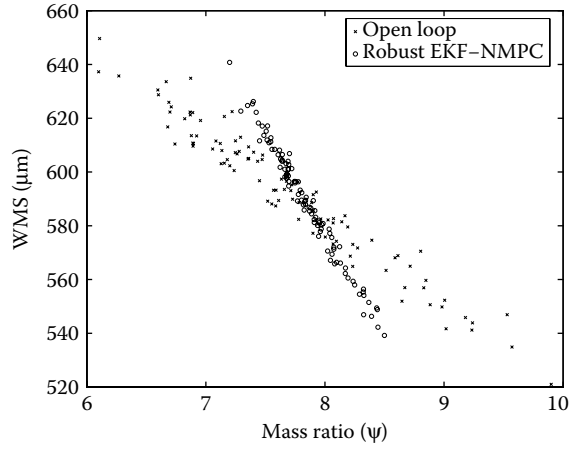

FIGURE 15.15 The weight-mean size and the ratio of nucleated mass to seed mass at the end of the batch for 100 Monte Carlo simulations using the dynamic model, when open-loop optimal control and BNMPC is implemented. (Reprinted from Nagy, Z.K. and R.D. Braatz, *AIChE J.*, 49:1776–1786, 2003. With permission.)

at $t = 0$ was practically the same as the open-loop optimal control problem. If the BNMPC optimizations were solved each step until full convergence for the nominal model, then the subsequent profiles would also be the same as the open-loop temperature profile. In practice, disturbances, model/plant mismatch, and the application of fixed-iteration BNMPC schemes (i.e., schemes that allow convergence of the optimization over several sampling periods rather than solving the optimization in each sampling period until full convergence) for better computational performance would lead to differences between the open-loop and nominal BNMPC profiles. Figure 15.13 also shows the temperature profile resulting from the robust BNMPC approach (Equation 15.61) for $\lambda = 0.01$ and $w = 0.2$, for which similar nominal performance was achieved with significantly enhanced robust performance. Distributional analysis provides a more comprehensive assessment of the robust performance. Figures 15.14a–c show the variation of the pdf of the performance index along the batch run for open-loop optimal control, nominal BNMPC, and robust BNMPC. The probability distribution functions were computed using the first-order sensitivities along the optimal temperature trajectories. The mean of the distributions of ψ increased monotonically during the batch run for all three cases. The BNMPC approaches produced a considerably narrower distribution for the entire batch compared to open-loop optimal control. Compared to the nominal BNMPC, the robust BNMPC approach yielded a narrower end-point distribution, which is expected since the end-point variance was included in the optimization objective (Equation 15.61), and provided a slightly wider distribution in the third quarter of the batch run.

To evaluate the importance of including the variance of a particular product property in the objective function, the variation of another product property (weight-mean size $WMS = \mu_4/\mu_3$), which was not included in the objective, was also computed using Monte Carlo simulation, generating 100 random parameter vectors from the uncertainty description (Equation 15.58). Figure 15.15 shows the effect of parameter uncertainty on the two CSD properties at the end of the batch. The WMS at the end of the batch is not significantly more robust for the BNMPC algorithms than for the open-loop implementation. Hence a batch control strategy can be highly robust for some product quality variables while being less robust for others. This illustrates the importance of considering in the robustness term of the objective function all product quality variables for which increased robustness is desired.

The robust BNMPC approach enhances robust performance for two reasons. First, the closed-loop structure is inherently more robust than the open-loop approaches as illustrated by the better performance of the nominal BNMPC compared to open-loop control. Second, the robust formulation of the online optimization problem further augments robust performance.

15.8 Conclusions

Batch processes, which are widely used for the manufacture of high-value-added products, have particular features that can make their control challenging. A distinguishing characteristic of most batch processes is the presence of at least one end-point property objective that cannot be measured until the end of the batch, hence need to be predicted via a model. This chapter reviews the field of batch process control with an emphasis on BNMPC. Typical problem formulations for industrial batch process control are described, which include both the setpoint tracking and setpoint optimization control based on end-point performance criteria. The importance of robust and real-time feasible solution of the online optimization problem as well as the benefits of taking uncertainties into account in controller design are emphasized, and several practical approaches to tackle these issues are presented. Case studies that apply BNMPC to an industrial pilot-scale reactor and a batch crystallization process illustrate some of the issues that a practitioner must address to provide quantifiable plant performance improvements that can be supported in an industrial environment.

15.9 Defining Terms

Batch-to-batch control: A control algorithm with the objective of achieving convergence of the product quality over several batch runs, by using the information from previous batches in order to design the control law for the next batch.

Batch nonlinear model predictive control (BNMPC): Nonlinear model predictive control algorithms formulated to address the needs of batch process control problems.

Control horizon: The length of time in which future control inputs are computed at each sampling time in an MPC algorithm.

Extended Kalman filter (EKF): A state estimator that applies a Kalman filter to an updated linearized process model to estimate the state covariance matrix.

Iterative learning control: The use of successive experiments to determine the control input trajectory that is closest to producing a desired output response. This control approach is applicable to batch processes, in which the batch can be repeated many times. Typical applications areas have been in the control of robots in manufacturing lines and for high-value chemical processes.

Mean-variance optimization: A robust optimization formulation that optimizes a weighted sum of the expected value of the objective function and the stochastic deviation of the expected value due to model uncertainties. To reduce computational cost, it is common in practice to replace the expected value with the value of the optimization function for nominal values of the model parameters.

Minmax optimization: A robust optimization formulation that optimizes the worst-case value for the objective function. This is also known as *minimax optimization* in the literature.

Model predictive control (MPC): A control algorithm that incorporates a process model to repeatedly solve an online optimization to determine the control inputs (i.e., values of the manipulated variables).

Moving-horizon control: An MPC implementation in which the prediction horizon is constant and shifts forward by one sampling interval at each control iteration. When applied to batch processes, the MPC implementation is usually switched to a shrinking-horizon once the prediction horizon extends beyond the final batch time.

Moving-horizon estimation (MHE): An algorithm that incorporates a process model to repeatedly solve an online optimization to determine state estimates.

Multiple shooting: An efficient technique for the numerical solution of optimal control problems that runs a separate model simulation over each subinterval of the control horizon.

Nonlinear model predictive control (NMPC): A control algorithm that incorporates a nonlinear process model to repeatedly solve an online optimization to determine the control inputs (i.e., values of the manipulated variables).

Nonlinear program (NLP): Optimization problem in which the objective or constraints are nonlinear algebraic functions of the optimization variables.

Parameter adaptive Extended Kalman filter: An Extended Kalman filter that is applied to some or all of the model parameters in addition to the states.

Particle filtering: A class of nonlinear state estimation algorithms that employs repeated direct numerical solution of the nonlinear dynamic process model.

Polynomial chaos expansion: An approach for quantifying the effects of model uncertainties on state and output trajectories in which the process is approximated by a series expansion with basis functions optimized based on the probability distribution functions for the model parameters.

Prediction horizon: The length of time for which future model predictions are made at each sampling time in an MPC algorithm.

Quadratic program: Nonlinear program in which the objective function is a quadratic and the constraints are a linear function of the optimization variables.

Receding-horizon control: This is the same as moving-horizon control.

Robust optimization: An optimization that has been formulated to take uncertainties into account.

Run-to-run control: This is the same as batch-to-batch control.

Sequential optimization approach: An approach for the numerical solution of optimal control problems by sequentially performing dynamic model simulation and the optimization steps.

Setpoint optimizing nonlinear model predictive control (SONMPC): An NMPC algorithm with the objective of optimizing control inputs and/or setpoint trajectories to reach an end-point target, typically defined to optimize a product quality variable.

Setpoint tracking nonlinear model predictive control (STNMPC): An NMPC algorithm that determines the control inputs to optimally track a setpoint trajectory.

Shrinking-horizon control: An MPC implementation applied to batch processes in which the prediction horizon shrinks by one sampling interval at each control iteration to have a fixed final batch time. When the final batch time is allowed to vary, the prediction horizon runs from the current time to the time in which an end-point condition is satisfied (the end-point condition is often defined in terms the satisfaction of an inequality that defines yield or purity).

Simultaneous optimization approach: An approach for the numerical solution of optimal control problems by incorporation of a discretization of the differential equations for the dynamic model as constraints in the optimization.

Soft constraints: A commonly used method in MPC algorithms in which deviations from constraint satisfaction are penalized in the optimization objective. This approach is often used to ensure or to increase the likelihood of feasibility of the optimization problem at each sampling time.

Successive quadratic programming (SQP): A class of numerical algorithms for determining the local optimum of an NLP by repeated solution of quadratic programs determined by approximated the nonlinear optimization objective by a quadratic function and the nonlinear constraints by a polytope. These optimization algorithms are also called sequential quadratic programming.

References

Allgöwer, F., T.A. Badgwell, S.J. Qin, J.B. Rawlings, and S.J. Wright, Nonlinear predictive control and moving horizon estimation—An introductory overview, in *Advances in Control, Highlights of ECC'99*, P.M. Frank (Ed.), Springer, Berlin, 391–449, 1999.

Atkinson, A.C. and A.N. Donev, *Optimum Experimental Designs*, Clarendon Press, Oxford, 1992.

Beck, J.V. and K.J. Arnold, *Parameter Estimation in Engineering and Science*, Wiley & Sons, New York, 1977.

Bequette, B.W., Nonlinear control of chemical processes—A review, *Ind. Eng. Chem. Res.*, 30:1391–1413, 1991.

Biegler, L.T., Efficient solution of dynamic optimization and NMPC problems, in *Nonlinear Predictive Control*, F. Allgöwer and A. Zheng (Eds.), Birkhaüser-Verlag, Basel, 219–244, 2000.
Bien, Z. and J. Xu (Eds.), *Iterative Learning Control: Analysis, Design, Integration and Applications*, Kluwer Academic Publishers, Boston, MA, 1998.
Bodizs, L., M. Titica, N. Faria, B. Srinivasan, D. Dochain, and D. Bonvin, Oxygen control for an industrial pilot-scale fed-batch filamentous fungal fermentation, *J. Process Control*, 17:595–606, 2007.
Cuthrell, J.E. and L.T. Biegler, Simultaneous optimization and solution methods for batch reactor profiles, *Comp. Chem. Eng.*, 13:49–62, 1989.
Darlington, J., C.C. Pantelides, B. Rustem, and B.A. Tanyi, Decreasing the sensitivity of open-loop optimal solutions in decision making under uncertainty, *Eur. J. Oper. Res.*, 121:343–362, 2000.
de Oliveira, N.M.C. and L.T. Biegler, Constraint handling and stability properties of model-predictive control, *AIChE J.* 40(2):1138–1155, 1994.
Diehl, M., *Real-Time Optimization for Large Scale Nonlinear Processes*, PhD Thesis, University of Heidelberg, 2001.
Diehl, M., H.G. Bock, and E. Kostina, An approximation technique for robust nonlinear optimization, *Math. Program.*, 107:213–230, 2006.
Diehl, M., H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, Real-time optimization and nonlinear model predictive control of processes governed by differential algebraic equations, *J. Process Control*, 12:577–585, 2002.
Diehl, M., J. Gerhard, W. Marquardt, and M. Moenigmann, Numerical solution approaches for robust nonlinear optimal control problems, *Comp. Chem. Eng.*, 32:1279–1292, 2008.
Eaton, J.W. and J.B. Rawlings, Feedback-control of chemical processes using online optimization techniques, *Comp. Chem. Eng.*, 14:469–479, 1990.
Feehery, W.F., J.E. Tolsma, and P.I. Barton, Efficient sensitivity analysis of large-scale differential-algebraic systems, *Appl. Numer. Math.*, 25:41–54, 1997.
Findeisen, R., L.T. Biegler, and F. Allgöwer (Eds.), *Assessment and Future Directions of Nonlinear Model Predictive Control*. Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 2007.
Franke, R., E. Arnold, and H. Linke, HQP: A solver for nonlinearly constrained large-scale optimization, http://hqp.sourceforge.net, 2009.
Gelfand, A.E. and A.F.M. Smith, Sampling-based approaches to calculating marginal densities, *J. Am. Statist. Assoc.*, 85(410):398–409, 1990.
Goodwin, G.C., M.M. Seron, and J.A. De Dona, *Constrained Control and Estimation—An Optimisation Approach*, Springer-Verlag, Berlin, 2005.
Gupta, M. and J.H. Lee, Robust repetitive model predictive control, *J. Process Control*, 16:545–555, 2006.
Henson, M.A., Nonlinear model predictive control: Current status and future directions, *Comp. Chem. Eng.*, 23:187–201, 1998.
Hermanto, M.W., R.D. Braatz, and M.-S. Chiu, Nonlinear model predictive control for the polymorphic transformation of L-glutamic acid crystals, *AIChE J.*, 55:2631–2645, 2009.
Hermanto, M.W., R.D. Braatz, and M.-S. Chiu, Integrated batch-to-batch and nonlinear model predictive control for polymorphic transformation in pharmaceutical crystallization, *AIChE J.*, 56, 2010, DOI 10.1002/aic.12331.
Hermanto, M.W., X.Y. Woo, R.D. Braatz, and M.-S. Chiu, Robust optimal control of polymorphic transformation in batch crystallization, *AIChE J.*, 53:2643–2650, 2007.
Julier, S.J. and J.K. Uhlmann, Unscented Kalman filtering and nonlinear estimation, *Proc. IEEE*, 92:401–422, 2004.
Larson, P.A., D.B. Patience, and J.B. Rawlings, Industrial crystallization process control, *IEEE Control Systems Mag.*, 26:70–80, 2006.
Li, S. and L.R. Petzold, *Design of New DASPK for Sensitivity Analysis*, Technical Report, University of California, Santa Barbara, 1999.
Ljung, L., *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
Moore, K., *Iterative Learning Control for Deterministic Systems*, Springer-Verlag, London, 1993.
Myerson, A., *Handbook of Industrial Crystallization*, Butterworth Heinemann, London, 2001.
Nagy, Z.K. and F. Allgöwer, A nonlinear model predictive control approach for robust end-point property control of a thin film deposition process, *Int. J. Robust Nonlinear Control*, 17:1600–1613, 2007.
Nagy, Z.K. and R.D. Braatz, Robust nonlinear model predictive control of batch processes, *AIChE J.*, 49:1776–1786, 2003.
Nagy, Z.K. and R.D. Braatz, Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis, *J. Process Control*, 14:411–422, 2004.
Nagy, Z.K. and R.D. Braatz, Distributional uncertainty analysis using power series and polynomial chaos expansions, *J. Process Control*, 17:229–240, 2007.

Nagy, Z.K., B. Mahn, R. Franke, and F. Allgöwer, Efficient output feedback nonlinear model predictive control for temperature control of industrial batch reactors, *Control Eng. Pract.*, 15:839–859, 2007.
Qin, S.J. and T. Badgwell, A survey of industrial model predictive control technology, *Control Eng. Pract.*, 11:733–764, 2003.
Rawlings, J.B., S.M. Miller, and W.R. Witkowski, Model identification and control of solution crystallization processes: A review, *Ind. Eng. Chem. Res.* 32:1275–1296, 1993.
Robertson, D.G., J.H. Lee, and J.B. Rawlings, A moving horizon-based approach for least-squares estimation, *AIChE J.*, 42:2209–2224, 1996.
Rustem, B., Stochastic and robust control of nonlinear economic systems, *Eur. J. Oper. Res.*, 73:304–318, 1994.
Srinivasan, B., S. Palanki, and D. Bonvin, Dynamic optimization of batch processes I: Characterization of the nominal solution, *Comp. Chem. Eng.*, 27:1–26, 2003a.
Srinivasan, B., S. Palanki, and D. Bonvin, Dynamic optimization of batch processes II: Role of measurements in handling uncertainty, *Comp. Chem. Eng.*, 27:27–44, 2003b.
Terwiesch, P., M. Agarwal, and D.W.T. Rippin, Batch unit optimization with imperfect modeling: A survey, *J. Process Control*, 4:238–258, 1994.
Valappil, J. and C. Georgakis, Systematic estimation of state noise statistics for extended Kalman filters, *AIChE J.*, 46:292–308, 2000.

For Further Information

Many textbooks are available on least-squares and maximum-likelihood model identification and the estimation of the hyperellipsoidal uncertainty descriptions from experimental data (e.g., Beck and Arnold, 1977; Ljung, 1987). Many detailed descriptions of optimal experimental design for nonlinear processes have been published, many of which describe industrial applications (Atkinson and Donev, 1992; Beck and Arnold, 1977; and citations therein).

MHE explicitly incorporates constraints during the state estimation (e.g., Goodwin et al., 2005; Robertson et al., 1996). Unscented Kalman filtering and particle filtering methods (Gelfand and Smith, 1990; Julier and Uhlmann, 2004; and citations therein), which employ repeated direct numerical solution of the dynamic nonlinear process model, have become regularly applied in recent years (e.g., Hermanto et al., 2009). Particle filtering methods are easy to implement, even for systems with large numbers of states, and have demonstrated good performance in applications. The parameter adaptive EKF formulation in Section 15.3.6.1 was developed by Valappil and Georgakis (2000). The sensitivities in this method and those used in many of the robust control methods can be computed by finite differences or by more sophisticated methods using automatic differentiation that are incorporated into some differential-algebraic equation solvers (Feehery et al., 1997; Li and Petzold, 1999).

Many reviews of NMPC have been published over the past two decades (Allgöwer et al., 1999; Bequette, 1991; Findeisen et al., 2007; Henson, 1998; Qin and Badgwell, 2003). Shrinking-horizon BNMPC algorithms were formulated in the late 1980s (Eaton and Rawlings, 1990). Several detailed descriptions of the multiple shooting algorithm are available (Diehl, 2001; Diehl et al., 2002; and citations therein). The HQP solver is available from Franke et al. (2009). Various methods for the numerical solution of NMPC algorithms such the sequential and simultaneous approaches are described in numerous papers (e.g., Biegler, 2000; Cuthrell and Biegler, 1989; de Oliveira and Biegler, 1994; and citations therein).

Dozens of papers have been published on measurement-based optimization approaches in recent years, for a range of applications (e.g., Bodizs et al., 2007; Srinivasan et al., 2003a,b; and citations therein). There have been many developments in batch-to-batch and iterative learning control in the last 15 years (e.g., Moore, 1993; Bien and Xu, 1998; Gupta and Lee, 2006). More information on exploiting the periodic nature of batch processes during control is discussed in this chapter of this Handbook on iterative learning control. An overview is available on methods to integrate batch-to-batch and within-batch approaches (Hermanto et al., 2010).

Many methods for analyzing the effects of uncertainties in batch control systems and solving minmax batch optimal control problems have been developed (Darlington et al., 2000; Nagy and Braatz, 2004, 2007;

Hermanto et al., 2007; Terwiesch et al., 1994; and citations therein). Many of these methods are based on power series or polynomial chaos expansions and are general enough to apply to nonlinear distributed-parameter integro-differential-algebraic equations under feedback. Several detailed descriptions of robust open-loop batch optimization approaches are available (e.g., Diehl et al., 2006, 2008; Nagy and Braatz, 2004). Alternatives to the minmax approach include the sensitivity robustness and the mean-variance approaches (Nagy and Braatz, 2004; Rustem, 1994; and citations therein).

Robust BNMPC can be formulated to use closed-loop control laws, in which a robust local feedback law is identified and applied during each BNMPC sampling period (Nagy and Allgöwer, 2007). This approach can provide significantly better robust performance than the typical BNMPC implementation, which has no feedback between the BNMPC sampling times.

Sections 15.5 and 15.6 are adapted from Nagy et al. (2007) with permission from Elsevier. The NMPC formulations were numerically solved using OptCon, which is based on the SQP optimizer HQP (Franke et al., 2009) used in conjunction with the implicit differential-algebraic-equation solver DASPK (Li and Petzold, 1999). The NMPC implementation is based on the direct multiple shooting method, which exploits the special structure of NMPC optimization problems. OptCon uses low-rank updates of the approximation of the Lagrangian Hessian of the nonlinear subproblems combined with a sparse interior point algorithm for the efficient treatment of the linear-quadratic subproblems in the nonlinear SQP iterations. Bounds and inequality constraints are handled using a barrier method and line search is used for global convergence of the SQP iterations. The software is available from the first author.

Models and measurement techniques for crystallization processes are reviewed in many books and papers (e.g., Larson et al., 2006; Myerson, 2001; Rawlings et al., 1993; and citations therein). The particular application in Section 15.7 was adapted from Nagy and Braatz (2003) with permission from the American Institute of Chemical Engineers (AIChE).

16

The Use of Multivariate Statistics in Process Control

Michael J. Piovoso
Pennsylvania State University

Karlene A. Hoo
Texas Tech University

16.1 Introduction

Advancements in automation and distributed control systems make possible the collection of large quantities of data. But without the corresponding adequate tools, it is not possible to interpret the data. Every modern industrial site believes that this data bank is a gold mine of information if only the *important* and relevant information could be extracted painlessly and quickly. Timely interpretation of data would improve quality and safety, reduce waste, and improve business profits. This interpretation is possible except for the following dilemmas: undetected sensor failures, uncalibrated and misplaced sensors, lack of integrity of the data historian and of data compression techniques used to store the data, and transcription errors. It is no wonder that data analysis methods in the face of these serious problems may appear to be inadequate. Meanwhile, the data bank continues to grow without appearing to garner any useful information. Without accurate and timely measurements, feedback control of the process to some specified objective is very difficult, if not impossible. For example, in the chemical industry, composition measurements are usually not made online; rather they are sampled and analyzed off-line. The delay between the samples and the results are usually on the order of hours. Thus, timely information about the purity of the composition is unavailable to take remedial control action.

In a typical chemical process, it is not unusual to sample and store hundreds of process variable measurements. These data can be characterized as being, noisy and collinear. In addition, there are instances when measurements are not present in the data set, and also when the value of variables are grossly erroneous. To handle these data requires tools that are capable of handling the redundancy,* noise, and missing information.

* Although there are hundreds of measurements, there are not hundreds of different events occurring.

A good design of experiments and *a priori* knowledge of the process would allow the use of standard techniques such as multiple linear regression (MLR) to develop predictive models. In practice, the luxury of carrying out a design of experiments on an operating process is unlikely because of production requirements and financial loss. In majority of the situations, only historical data are available. MLR works best when the independent variables are noise free and uncorrelated, which is unrealistic for real process data.

This article discusses the use of more appropriate multivariate statistical techniques to analyze process data (historical), and to develop predictive models in support of process monitoring and control. In particular, the multivariate methods of partial least squares or projection to latent structures (PLS), principal component analysis/principal component regression (PCA/PCR) will be presented starting with the theoretical development, followed by examples to illustrate concepts and implementation on real industrial processes.

PCA is a method for modeling a set of data assembled in a matrix **X**, where the rows are the sampled process variables at a fixed sampling time, and a column is an uniformly sampled variable. PCA produces a mapping of the data set **X** onto a reduced subspace defined by the span of a chosen subset of eigenvectors of the variance–covariance matrix of the **X** data. This set of eigenvectors or directions in the **X** space are referred to as the PCA loadings. This technique has the advantage in that it allows the development of a linear model, which produces an orthogonal set of pseudo-measurements that contain the significant variations of the **X** data. The first pseudo-measurement or principal component explains the greatest amount of variation and the second the next largest amount after removal of the first effect, and so on. These pseudo-measurements are the inner products of the true measurements with the loadings and are called the scores. The entire sets of scores and loadings define the process data, and the loadings are the statistical process model. A subset of the first few scores provide information in a lower dimensional space of the behavior of the process during the period in which the measurements were made. This set of scores and the PCA loadings can be used to determine if the present process operation has changed its behavior relative to the data that was used to define the scores and loadings (Piovoso et al., 1992a). In addition this score space has properties, which make it attractive for doing multivariate statistical process control (SPC) (Kresta and MacGregor, 1991).

While PCA is suitable for process monitoring, when there is a specific control objective, PCA is not the appropriate tool. For example, if a critical measurement is not readily available, such as when a laboratory analysis is required, then PLS or PCR are tools to consider. PLS is like PCA in that it provides a model for the **X** space of data. However, it is not the same model as the PCA model. The PLS **X**-space model is a rotated version of the PCA model. The rotation is defined in a way such that the scores of the measurements provide the maximum information about the quantity to predict called the **Y**-data. An example in which PLS might be an appropriate tool is the control of the nonmeasured composition of the distillate output of a distillation column (Piovoso and Kosanovich, 1994). Traditionally, this is accomplished by selecting a tray temperature, which best correlates with the actual composition and holding that temperature, at a prescribe value. Generally, there are many tray temperatures available; PLS might be used to model composition using the redundant information contained in multiple tray temperatures. Now, the control objective is not to hold one tray temperature constant, but rather to allow all the tray temperatures to move as needed to maintain the desired composition setpoint.

PCR is an extension of PCA to the modeling of some **Y** data from the **X**. The approach to defining this relationship is accomplished in two steps. The first is to perform PCA on the **X** data and then to regress the scores onto the **Y** data. Unlike PLS, PCR establishes its loadings independent of the **Y** data set.

The statistics used in processing monitoring are based on the assumption that the PCA components are assumed to be mutually statistically independent instead of just uncorrelated. With most process, this assumption is not valid; independent component analysis (ICA) overcomes this limitation and allows modeling based on non-Gaussian behavior. ICA provides a meaningful way of decomposing a set of data into a linear combination of independent sources. All but one of the sources must be non-Gaussian to obtain a valid decomposition.

16.2 Multivariate Statistics

In this section, we review the theoretical foundations of the multivariate statistical methods providing only necessary information that will allow the reader to grasp the important concepts. A thorough review of PCA can be found in the article by Wold et al. (1987), and that of PLS and PCR by Martens and Næs (1989).

16.2.1 Principal Component Analysis

PCA involves several steps. First, the data are mean centered, and often normalized by the standard deviation. Mean centering implies that the average value for each variable is subtracted from the corresponding measurement. Scaling is necessary to avoid problems associated with having some measurements with large values and others with small ones. For example, pressure may be measured in the thousands of Pascals while temperature may be in units of hundreds of degrees Celcius. Scaling puts all the numbers on the same magnitude. It implies multiplying the mean centered data by an appropriate constant, usually the inverse of the standard deviation.

From this normalized data, the variance–covariance matrix is generated by the relationship $\mathbf{X}^T\mathbf{X}$, where $\mathbf{X}$ is the normalized data matrix. The $\mathbf{X}^T\mathbf{X}$ matrix is positive semidefinite* and it defines the directions in the $\mathbf{X}$-space where most of the variability occurs. These directions constitute the eigenvectors of $\mathbf{X}^T\mathbf{X}$. The eigenvalues are related to the amount of variability explained by each eigenvector.

PCA compresses the information in the $\mathbf{X}$ matrix, $\mathbf{X} = (x_1^T, x_2^T, \ldots, x_K^T)^T$, into a set of pseudo-variables $\mathbf{T} = (t_1, t_2, \ldots, t_A)$. The row vector x_k represents process measurement at time k, and the column vector t_a represents the time history of the projection of all the measurements onto the ath eigenvector.

PCA is illustrated in Figure 16.1. In this example, two measurements are being made on a given process. Observe that the data are not linearly independent because as measurement 1 increases, measurement 2 also increases. In general, the measurements define a K-dimensional hyperspace, where K is the rank of $\mathbf{X}^T\mathbf{X}$. The first eigenvector is the direction where the data exhibit the greatest variability. This is illustrated by the vector, p_1. The second eigenvector will be orthogonal to the first (in the direction of greatest variability of the residual $(\mathbf{X} - t_1 p_1^T)$), and is denoted by p_2. Some eigenvectors may define directions of extraneous information, as for example, p_2 which appears to be needed only to explain the noise that is in the data. If only one eigenvector, p_1, is used to represent $\mathbf{X}$, then a smoothed reconstruction of the $\mathbf{X}$ data is possible.

Consider measurement vector at the kth time interval, x_k. This data point has a projection onto the first eigenvector. The distance from the origin along the vector p_1 to the projection point is the associated score. The coordinates of this projection point on the line spanned by the first eigenvector represent a reconstruction of the data from a single eigenvector. The projection error, e_k, is a point in the subspace orthogonal to that containing the reconstructed data point.

Generalizing, PCA can be decomposed as follows. Let $\mathbf{P} = (p_1, p_2, \ldots, p_A)$, where p_j is the jth eigenvector of the covariance matrix $\mathbf{X}^T\mathbf{X}$; and

$$\tau_a = t_a^T t_a \tag{16.1}$$

The scalar, τ_a, defines the amount of variability in the normalized measurement data that is explained by the ath eigenvector. Furthermore, the matrix $\mathbf{P}$ has the property† that

$$\mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{16.2}$$

where $\mathbf{I}$ is the identity matrix. The relationship between $\mathbf{P}$ and τ_a is

$$\mathbf{X}^T\mathbf{X}p_a = p_a\tau_a \tag{16.3}$$

* Eigenvalues are all nonnegative.

† Orthonormal.

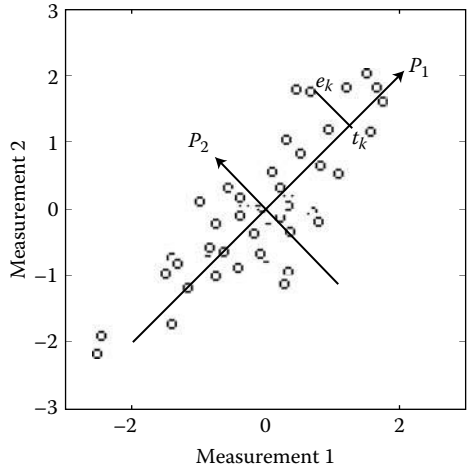

FIGURE 16.1 PCA illustration.

If all the eigenvectors have been extracted, then **X** can be reconstructed perfectly from

$$\mathbf{X} = \mathbf{TP}^T \tag{16.4}$$

If only $A < K$ eigenvectors are used, then **X** is only approximately recovered. In this case,

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E}_x \tag{16.5}$$

and $\mathbf{E}_x$ are the reconstruction errors. Figure 16.1 shows not only the projection of a measurement onto an eigenvector producing a score, but also the component of the orthogonal error.

16.2.2 Principal Component Regression

PCR is an extension of PCA applied to the modeling of **Y** data from the **X** or measurement data. For example, if **X** is composed of temperatures and pressures, **Y** may be the set of compositions that result from thermodynamic considerations. The approach to defining this relationship is accomplished in two steps. The first is to perform a PCA on the **X** data which yields a set of scores for each measurement vector. That is, if x_j is the jth vector of the K measurements at a time j, t_j is the corresponding jth vector of A scores. Given the matrix of scores **T**, the **Y** data are regressed on the matrix of scores by

$$\mathbf{Y} = \mathbf{Tq} + \mathbf{E}_y \tag{16.6}$$

Using the orthogonality of the matrix of eigenvectors, **P** and Equation 16.4, **T** is related to data matrix **X** by

$$\mathbf{T} = \mathbf{XP} \tag{16.7}$$

Substituting this in Equation 16.6 gives,

$$\mathbf{Y} = \mathbf{XPQ} + \mathbf{E}_y \tag{16.8}$$

or

$$\mathbf{B} = \mathbf{PQ} \tag{16.9}$$

where **B** are the PCR coefficients of **X** onto **Y**.

16.2.3 Independent Component Analysis

ICA is similar to PCA. In PCA, we seek directions in the X-space that provides the maximum explanation of the variability. In ICA, we seek a set of statistically independent sources that are consistent with the data. Let $\mathbf{x}$ be an m-dimensional vector of data. The n-dimensional source vector $\mathbf{s}$, is determined from $\mathbf{x}$ by

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{16.10}$$

where $\mathbf{W}$ is a matrix that must be determined from the data. $\mathbf{W}$ is chosen so that the elements of $\mathbf{s}$ are as independent as possible. Let G be a function of the components of $\mathbf{s}$. As described by Hyvarinen (1999), $\mathbf{W}$ is that matrix which maximizes some measure of the linear independence of $\mathbf{s}$.

An important distinction here is the difference between statistical independence and noncorrelation. For a set of data to be statistically independent, the joint probability density is a product of the marginal probability density,

$$f(y_1, y_2, \ldots, y_m) = f_1(y_1) f_2(y_2) \ldots f_m(y_m) \tag{16.11}$$

where f represents the joint probability density and $f_j(y_j)$ is the marginal probability of the random variable y_j. If the random variables $\{y_1, y_2, \ldots, y_m\}$ are statistically independent, then they are also uncorrelated. If they are uncorrelated, they need not be statistically independent. Being uncorrelated merely implies that the mean of the product of two random variables is the product of their individual means. Thus,

$$\mathcal{E}(y_1 y_2) = \mathcal{E}(y_1)\mathcal{E}(y_2) \tag{16.12}$$

If the random variables are statistically independent, then

$$\mathcal{E}(g_1(y_1) g_2(y_2)) = \mathcal{E}(g_1(y_1))\mathcal{E}(g_2(y_2)) \tag{16.13}$$

In the case of Gaussian random variables, if the random variables are uncorrelated, they also are statistically independent. If this is the case, ICA is not particularly interesting or even possible. If the data are Gaussian, then PCA is an appropriate analysis.

Having found the sources, $\mathbf{s}$, the original data, $\mathbf{x}$, can be reconstructed through

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{16.14}$$

where $\mathbf{A}$ is an $m \times n$ matrix that defines the necessary combinations of the independent sources needed to generate $\mathbf{x}$. The vector $\mathbf{n}$ is the additive noise. To determine the independent sources and to reconstruct the original data $\mathbf{x}$, both $\mathbf{A}$ and $\mathbf{W}$ must be defined. The assumptions needed to generate those matrices are,

1. $n - 1$ of the n sources must be non-Gaussian. One source may be Gaussian.
2. The number of variables in $\mathbf{x}$ must be greater than the number of sources ($m > n$).
3. The columns of the matrix $\mathbf{A}$ must be linearly independent.

One of the original applications of ICA was to blind source separation. In this problem, there are m sensors recording m discrete time signals. These signals are a linear combination of n different sources. As an example, suppose there are a number of people speaking at a cocktail party and a number of recorders (more than the speakers) recording these conversations. The problem is to separate each of the speakers based on the data found on each of the recorders. The ICA method has been applied broadly to electroencephalographic (EEG) data (Vigario, 1997) and economic data (Kiviluoto and Oja, 1998).

ICA has found application in process monitoring and fault detection and identification as an alternative to PCA. ICA is a particularly attractive approach because it depends on non-Gaussian data that is most commonly found in processes. Lee et al. (2004) use ICA for statistical process monitoring and fault detection of continuous processes. Since the T^2 statistic is no longer meaningful, they propose the I^2 statistic, which is a sum of the square of the sources. By breaking the sources into systematic and

nonsystematic parts, a second statistic, I_e^2, based on the nonsystematic part of the source was developed (Lee et al., 2004). This statistic is useful in fault detection and identification problems. In addition, relationships for contribution plots also were developed so that variables affected by disturbance or faults readily can be identified. Lee et al. applied the new statistics and the usual Q statistic to data from a mathematical simulation and data from a waste water treatment plant.

Yoo et al. (2004) have extended ICA to a multiway version (MICA) to study batch process monitoring. In this work Yoo and coworkers unfold the three-dimensional batch data into a two-dimensional matrix in a manner similar to what is usually applied to Multiway PCA (MPCA). Each row corresponds to one batch with all the variables at the first time instance followed by all the variables at the second, and so forth. Using these data, a MICA model is developed. Yoo et al. (2004) also use the I_e^2 and Q statistics to determine if the batch has performed properly. They illustrate their work on a fed-batch penicillin fermentation process.

16.2.4 Partial Least Squares

PLS is a term for a family of philosophically and technically related mathematical techniques that were originally proposed by Herman Wold's fundamental concepts of iterative fitting of bilinear models in several blocks of variables (Wold, 1982; Wold et al., 1984). The original applications were to concrete data-analytical problems in economics and social sciences. This new technique was developed to address the shortcomings of standard methods when dealing with a modest number of observations, highly collinear variables, and data with noise in both the **X** and **Y** data sets. Standard techniques such as MLR had severe parameter identification and convergence problems.

PLS, sometimes called projections to latent structures, is similar to PCR. Both decompose the **X** data into a smaller score space **T**. They differ in how they relate the scores to the **Y** data. In PCR, the scores from the PCA decomposition of the **X** data are regressed onto the **Y** data. By contrast, in PLS, both the **Y** data and the **X** data are decomposed into scores and loadings. The orthogonal sets (**T, U**) in which results are generated in a way that maximizes the covariance between the scores for the **X** data and those for the **Y** data. This is attractive, particularly in situations where not all the major sources of variability in **X** are correlated to the variability in the **Y** data. PLS attempts to find a different set of orthogonal representation for the **X** data to give better predictions of the **Y** data. Thus, a given number A of orthogonal vectors will yield a *poorer* representation of the **X** data, while the scores for this same set of vectors will yield a *better* prediction of **Y** than would be possible with PCR.

Mathematically, **X** is decomposed into a model for the first principal component as

$$\mathbf{X} = t_1 p_1^T + \mathbf{E}_{x,1} \tag{16.15}$$

The **Y** data are similarly decomposed as

$$\mathbf{Y} = u_1 q_1^T + \mathbf{F}_{y,1} \tag{16.16}$$

Having found t_1, p_1, u_1, and q_1, the procedure is repeated for the residual matrices $\mathbf{E}_{x,1}$ and $\mathbf{F}_{y,1}$ to find t_2, p_2, u_2 and q_2. This continues until the residuals contain no useful information. The t's and u's are the scores, and the p's and q's are the loadings of the **X** and **Y** data respectively. Each t_i and u_i are related by linear regression to form

$$u_i = b_i t_i + r_i \tag{16.17}$$

Excellent treatments with more details are found in Geladi and Kowalski (1986) and Martens and Næs (1989).

The implementation of the PLS algorithm is slightly different for the case of single y-variable as opposed to the multiple y-variables. For the former, PLS is a single pass solution; whereas, for the latter, it is iterative. The approach for the single y-variable is to introduce an additional set of loading *weights* **W**, which permits an easier interpretation of the results than would be otherwise possible. The first step

to this modeling problem, like that of PCR, is to mean center and scale the **X** data so that one variable does not overwhelm another by its size. Also, the y-variable needs to be mean centered. The maximum number of loadings, Amax, to investigate must be specified. The following steps need to be performed for each factor a from $1 \leq a \leq$ Amax. Here, $\mathbf{X}(0)$ is the mean corrected and scaled versions of **X** and $y(0)$ is mean corrected version of the single y-variable.

1. Find $\hat{w}_a$ that maximizes $\hat{w}_a^T \mathbf{X}_{a-1}^T y_{a-1}$ subject to $\hat{w}_a^T \hat{w}_a = 1$.
2. Find the scores t_a as the projection of $\mathbf{X}_{a-1}$ on $\hat{w}_a$. Thus

$$t_a = \mathbf{X}_{a-1} \hat{w}_a. \tag{16.18}$$

3. Regress $\mathbf{X}_{a-1}$ on t_a to find the loadings p_a via

$$p_a = \mathbf{X}_{a-1}^T t_a / (t_a^T t_a) \tag{16.19}$$

4. Regress t_a on y_{a-1} to find

$$q_a = y_{a-1}^T t_a / (t_a^T t_a) \tag{16.20}$$

5. Subtract $t_a\, p_a^T$ from $\mathbf{X}_{a-1}$ and call this $\mathbf{X}_a$.
6. Subtract $t_a\, q_a^T$ from y_{a-1} and call this y_a.
7. Increment a by 1 and if $a \neq$ Amax + 1 go to step 1.

No iteration is required for finding loadings and scores when there is but a single y-variable.

The above algorithm needs to be slightly modified and iterated for multiple y's. The term y_{a-1} in step 1 above is replaced by a temporary u_a. The quantity u_a is iterated until convergence. Initially, u_a is chosen to be one of columns of the **Y**-data. The algorithm for this would be:

1. $\hat{w}_a = u_a^T \mathbf{X_{a-1}} / (u_a^T u_a)$ (regression of **X** onto u).
2. Normalize $\hat{w}_a$ to unit length.
3. $t_a = \mathbf{X}_{a-1} \hat{w}_a / (\hat{w}_a^T \hat{w}_a)$ (compute scores).
4. $q_a = t_a^T \mathbf{Y}_{a-1} / (t_a^T t_a)$ (regression of columns of **Y** onto t).
5. $u_a = \mathbf{Y}_{a-1}^T q_a / (q_a^T q_a)$ (upgrade estimate of u).
6. If u_a has not converged go to 1, otherwise.
7. $p_a = \mathbf{X}_{a-1}^T t_a / (t_a^T t_a)$ (compute the x-loadings).
8. Calculate residuals $\mathbf{X}_a = \mathbf{X}_{a-1} - t_a p_a^T$ and $\mathbf{Y}_a = \mathbf{Y}_{a-1} - t_a q_a^T$.
9. Increment a by 1 and if $a \neq$ Amax + 1 set u_a to a column of $\mathbf{Y}_{a-1}$ and go to 1.

The use of simultaneous information in the **X** data and **Y** data does have some disadvantages for the case of multiple y's. In order to have orthogonal sets of score vectors (**Q** and **T**), two sets of *loadings* or basis vectors (generally termed **W** and **P**) are needed for the **X** data. **W** is referred to as the *weights* and **P** the *loadings*. The **W** is orthonormal, but the **P** is not. Without the two, the score vectors are not orthogonal. One then loses computational advantages of having orthogonal scores (note item 8 above). If the scores are correlated, then the simultaneous regression of the **Y** and **X** on all the scores **T** is necessary. Also, a lack of orthogonality leads to greater variance in the regression results. The need to remove one vector at a time is of overwhelming importance; hence, the two sets of loading vectors is the preferable alternative to the simultaneous regression.

PLS, like any data modeling paradigm, may underfit or overfit the data. By underfitting, not enough loadings are used, and the model fails to capture some of the information. By overfitting, too many loading are used, and the model tends to fit some of the noise. Both cases produce suboptimal models. Thus, it is of paramount importance to validate the model to avoid these problems. Although there are many ways of doing this, we will discuss only the cross-validation method. The reader is referred to Martens and Næs (1989) for more details of this method and other validation methods. In cross validation, the data, both **X** and corresponding **Y**, are segregated into groups, typically 4–10. Using all but one of the groups, a new

PLS model is generated as the number of loadings varying from 1 to Amax. Each of these models is used to predict the **Y** data in the group withheld. The prediction error sum of squares (PRESS) is computed for each model. This procedure is repeated until each group is withheld once and only once. Then the overall PRESS is generating for a given number of loadings, *a*, by summing over the prediction errors for all withheld data. A plot of the PRESS versus loading number will typically reach a minimum and then start increasing again. The value corresponding to the minimum PRESS is taken as the number of loadings needed. Fewer than this number tend to underfit the data and more begin to overfit.

Example

An example demonstrates how PCR and PLS compare with MLR. Consider the following **X** data:

$$\mathbf{X} = \begin{bmatrix} 1.0 & 0.0 & -1.9985 \\ -1.0 & 1.0 & 3.4944 \\ 0.0 & -1.0 & -1.5034 \\ 1.0 & 1.0 & -0.4958 \end{bmatrix} \tag{16.21}$$

This **X** data can be viewed as three process measurements that are to be used to predict a single *y*-variable. There are four observations of $y = [-3\ 2475\ \ 5.2389 - 2.0067 - 1.2417]^T$. The correct model that relates the **X** data to the observation y is $y = -0.58x_1 + 1.333x_3$ where x_1 and x_3 are the first and third columns of the **X** data, respectively.

For these data, the model identified by MLR is $y = 0.75x_1 - x_2 + 1\ 5x_3$. The discrepancy between the true result and this estimate is due to the poor conditioning of the **X** data. Column 3 is highly correlated to columns 1 and 2. In fact, $x_3 \approx -2x_1 + 1.5x_2$. This collinearity makes the MLR solution highly sensitive to outliers and noise in the data. To illustrate this, consider the solution when 0.001 is added to each element of *y*. The new solution is given by $y = 0.0703x_1 - 0\ 3827x_2 + 1.5889x_3$. Subtracting 0.001 from each element of y produces a solution $y = 1.5793x_1 - 1.6173x_2 + 2.4111x_3$.

The problem is not only related to noise on the **Y** data. Suppose that we add zero-mean, normally-distributed, noise with variance 10^{-4} to each element of the **X** matrix. The MLR solution now becomes $y = -5.5716x_1 + 3.7403x_2 - 1.1518x_3$.

Notice the wide changes in the model with small changes in the data. Models that exhibit high sensitivity to small errors are not very robust. They will usually yield bogus results when used in a predictive mode. This is a serious drawback of MLR when applied to data that is highly collinear. On the other hand, the method works well when the **X** data are orthogonal as is the case in experimental designs.

Now, compare the results of MLR with those obtained using PCR. The PCA model requires two eigenvectors or loadings, and the resulting PCR model for the same set of data is $y = -0.6288x_1 + 0.0364x_2 + 1.3093x_3$. This is much closer to the true model than was found with MLR. Furthermore, the PCR solution is less sensitive to small deviations in either the **Y** data or **X** data. The model is essentially unchanged if 0.001 is added or subtracted from the original **Y** data. The same is true when a the small noise signal (10^{-4} variance) is added to the **X** data.

PLS works equally well. A two loading model provides a very similar solution to PCR, $y = -0.62831x_1 + 0.03697x_2 + 1.3093x_3$. Like PCR, the PLS solution is far less sensitive to small errors in the **Y** and **X** data than MLR method. Clearly, PLS and PCR have considerable advantage over MLR when there are few observations of correlated **X** data.

16.3 Areas of Applications

Multivariate statistical methods have many applications in the process industry. This section will deal with two application areas: data analysis/estimation and inferential measurement for control. Data analysis can be used to develop a model of the expected behavior of the process using PCA/PLS techniques. In theory, deviations from the model are an indication that something is amiss. Inferential measurements are used in cases where the quantity to be controlled is measured by infrequent laboratory analysis. A statistical

model can be used to estimate its value for control purposes. A method for controlling in the score space is developed. Examples are presented.

16.3.1 Data Analysis

Chemical processes typically have automated methods for collecting large amounts of data. These data may be a potential gold mine of information, but because tools are not in place that would separate pertinent information quickly from irrelevant noise, this potential is not realized. Multivariate statistical methods may possibly aid in separating the significant information from the noise.

Although many measurements are made, there are only a few physical phenomena occurring. Thus, many of the variables are highly correlated. Trying to learn about the process by looking at the data in a univariate fashion, only confuses the issues. Multivariate methods, such as PLS and PCA, often capture the essence of the information in a lower-dimensional space defined by two or three primary loadings or latent vectors. Observing these data and understanding the significance of any clusters in this lower dimensional space generally lead to insights about the process.

A process can be viewed as an instrument which can be calibrated with multivariate statistical tools (Piovoso et al., 1992a). Process data provide information about the process. If this instrument is calibrated correctly, that is, it provides the same measures for all periods in which the outputs or the quality variables are on-aim, then it can provide the kind of information that process engineers and operators need. If the process is drifting away from the quality targets, then this should be detected easily on-line and adjusted appropriately to prevent a poorer quality product and to reduce losses and downtime. To realize this idea, reference data during which the process is producing top-grade product are gathered. Process variability may be captured in a PCA model. This model provides a *fingerprint* of the process and sets the standard by which process operation is judged. New data are compared to this multivariate model to determine if they are consistent with the normal operation.

16.3.1.1 Process Monitoring and Detection

Real-time monitoring addresses the classification of new process data relative to the reference data. In this regard, it cannot be overemphasized that careful selection of the reference process data (normal operating conditions) must be used to develop the calibration model. Detection on the other hand, judges the appropriateness of the model. Both real-time monitoring and detection will be discussed in the context of the score space.

The Mahalanobis distance, h_i, is a measure of the extent to which the scores for new data, x_i can be classified as belonging to the set of scores which were used to generate the calibration model (Shah and Gemperline 1990). Thus, h_i measures the goodness of fit *within the model space*. For a new data vector x_i of size $(1 \times m)$, h (scalar) is calculated as

$$h_i = \left(x_i - \mu\right)^T \mathbf{S}^{-1} \left(x_i - \mu\right) \tag{16.22}$$

where μ is the centroid of the calibration model, and $\mathbf{S}$ is the variance–covariance matrix of the calibration set $\mathbf{X}$. Since t is the projection of data, x, onto a reduced space defined by the first A eigenvectors, and because the mean of the score vectors is zero due to mean centering of the data, h_i can be computed in the score space defined by the principal components to be

$$h_i = \frac{1}{m} + \sum_{a=1}^{A} \frac{t_{ia}^2}{t_a' t_a} \tag{16.23}$$

where m is the number of process measurements used in the model generation, t_{ia} is the ath score for the data vector, x_i, and t_a is the vector of scores corresponding to the ath principal component for the data used to generate the model. Following Shah and Gemperline (1990), if h_i is beyond the 90th percentile of the calibration set for A principal components, then the scores for x_i are considered outside the model space; otherwise the new data vector is similar to the reference data.

16.3.1.2 Process Description

An example is presented to illustrate the application of process monitoring and detection to a continuous chemical process (Piovoso et al., 1992a). In the first stage of that process, several chemical reactions occur that produce a viscous polymer product; in the second stage, the polymer is treated mechanically to prepare it for the third and final stage. Critical properties such as viscosity and density if altered significantly, will affect the final product resulting in a loss of revenue and operability problems for the customer. Moreover, it is difficult to determine which stage is responsible for the quality degradation. The situation is further confounded by a lack of on-line sensors to measure continuously the critical properties. This makes it impossible to relate any specific changes to a particular stage. Indeed, property changes are detected by laboratory measurements that may have delays of 8 h or more. The results of the analysis represent past information, consequently the current state of operations may not reflect the process state.

Such infrequent measurements make the control of the product quality difficult. At best, the operators have learned a set of heuristics that, if adhered to, usually produces a good product. However, unforeseen disturbances and undetected equipment degradations may occur, which will also affect the product. There are periods of operations when the final process step produces a degraded product in spite of near-perfect upstream operations.

Since the second stage operates in support of the third stage and to compensate for property errors in the first stage, this work will focus on process monitoring and detection of the second stage. A simplified description of the second stage is as follows. The feed from the reaction stage is combined with a solvent in a series of mixers operated at carefully controlled speeds and temperatures. A sample of the mixture is taken at the exit of the final mixer for lab analysis. From there, the process fluid is sent to a blender whose level and speed are controlled to impart certain properties. The fluid is then filtered to remove undissolved particulates before it is sent to the final stage of the process. A significant amount of mechanical energy is necessary to move the viscous fluid through a complex transport network of pipes, pumps, and filters. Consequently, the life span of the equipment is unpredictable. Even the same type of equipment that is placed in service at the same time may need replacing at widely differing times. Problems of incipient failure, pluggage, and unscheduled downtimes are an accepted, albeit undesirable, part of operations. To prolong continuous operations, pumps and filters are installed in pairs so that the load on one can be temporarily increased while the other is being serviced. Tight control of the process fluid is desirable because the equipment settings in the final stage of the process are preset to receive a uniform process fluid. As such, small deviations in the fluid properties may result in machine failure and nonsaleable product.

The frequent maintenance on the equipment is not the only the source of control problems. Abrupt changes also occur due to the throughput demands in the final stage of the process. For example, if there is a decrease in demand due to downstream equipment failure, or a sudden increase due to the addition of new or reserviced equipment, the second stage must reduce or increase production as quickly as possible. It is more dramatic when throughput has to be turned down because the process fluid properties will change if not treated immediately. These situations are frequent and unscheduled, thus the second stage of this process moves around significantly and never quite reaches an equilibrium. Clearly, throughput is a dominant effect on the variability in the sensor values and process performance.

16.3.1.3 Model Development

A calibration model to monitor and improve process operations of the second stage is developed since it is here that critical properties are imparted to the process fluid. Intuitively, if the second stage can be monitored to anticipate shifts in normal process operation or to detect equipment failure, then corrective action can be taken to minimize these effects on the final product. One of the limitations of this concept is that there will be disturbances that may affect the final product that will not manifest themselves in the variables from which the model is created. The converse is also true, that disturbances in the monitored

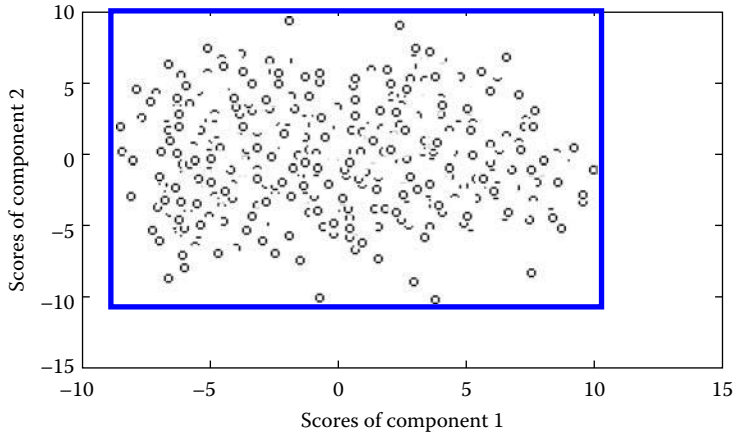

FIGURE 16.2 Scores of component 1 versus scores of component 2, PCA model.

variables may not affect the final product. However, faced with few choices, the use of a calibration model is a rational approach to monitor and to detect unusual process behavior to improve process understanding.

Since throughput has an effect on all the measurements, a PCA analysis would be overwhelmed by this effect, and it would obscure other information about the state of the process. By first eliminating the throughput effect using PLS analysis, an examination of the residuals using PCA reveals other sources of variations critical to process operations.

Routine process data are collected over a period of several months. Cross validation is used to detect and remove outliers, and only data corresponding to *normal* process operations, that is, when top-grade product is made, are used in the model development. Two calibration models are developed, both are reduced order model that capture dominant directions of variability in the data. The PLS model shows that two loadings are needed to explain approximately 60% of the variance in the measurements. A third loading does not change significantly the total explained variance, and the explained variances in the throughput variables are 100%. PCA analysis on the residuals shows that five principal components explain 90% of the residual variability. Additional components provide no added statistical significance.

Figure 16.2 shows the observations of the residual data plotted in the score space of principal components 1 and 2. The observations are spread out over a wide region and include a wide range of rate settings. In a related work, Piovoso et al. (1992b) show how filtering the data with a nonlinear, finite impulse, median hybrid filter can magnify differences in the operations and reveal structure in the data. By examining the loadings of the PCA, one may be able to relate the principal components to some physical phenomena in the process. This is particularly true for the early components since they explain most of the variability.

16.3.1.4 Online Monitoring and Detection

The PLS and PCA models are used online to monitor and to detect statistically significant deviations in process operations. The system configuration is given in Piovoso et al. (1992a). For each new data vector, the Mahalanobis distance is calculated. The Mahalanobis criterion gives a measure of location of new data within model subspace. If h_i is within the model space, then no alert is sent to the operators. Figure 16.3 illustrates a plot of the normalized Mahalanobis distance for a period of 30 h with the most recent time at the right side of the plot. The normalization is done with respect to the largest Mahalanobis distance in the calibration model. The operations from the fifth to the eleventh hour in the past indicated that the process variables fell outside the region of normal plant behavior. The reason for the unusual behavior

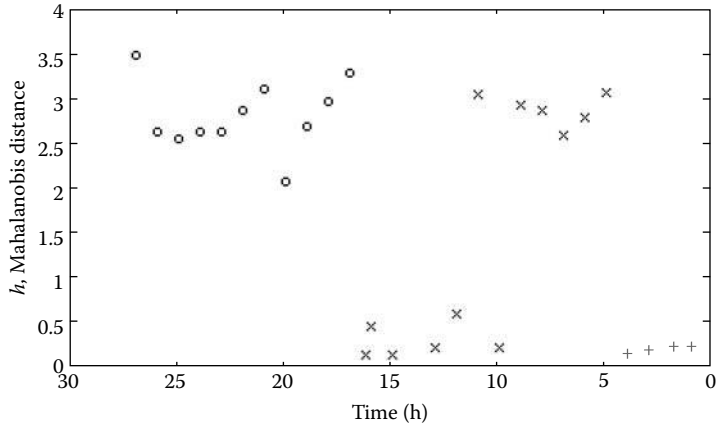

FIGURE 16.3 Mahalanobis distance for 30 h of online operation. +: most recent 4 h; x: most distant history.

for that period was an unexpected pump failure. There is indication of this failure at the eleventh hour and actual failure occurs at the ninth hour. During the most distant period, the process variables are again outside the desired region because there was a filter pluggage due to large particulates in the process fluid.

Along with the Mahalanobis distance plots, diagrams such as that shown in Figure 16.4 are used by operating personnel to monitor the process operations easily. The ○ denotes the scores for the calibration model, and the + denote the most recent period of operation. The arrows are used to indicate the process history. Clearly, there is a time period in which the process operation can be judged as dissimilar to the calibration set. Either an unusual control action or a disturbance shifted the process state from the desired region as evidence by the two operating points in the upper left hand side. Eventually, appropriate control actions are applied and the process returns to the region within defined by the calibration set.

One current limitation is the lack of good representative data. A design of experiments is necessary to map the entire range of operations as is discussed for the binary distillation column example presented in Section 16.3.4. If only a portion of the operating space is known, it is possible that the correlations between the inputs/outputs under closed-loop operations, may be incorrect if the process moves to a regime that is not a part of the reference set used to develop the model.

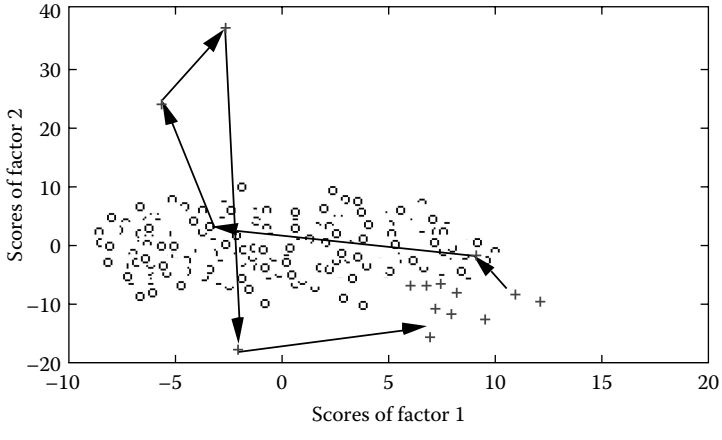

FIGURE 16.4 Score plot. +: most recent operation, o: desired operating region.

16.3.2 Batch Processes

Batch and semibatch processes play an important role in the chemical industry due to their low volume–high-value products. Examples include reactors, crystallizers, injection molding processes, and the manufacture of polymers. Batch processes are characterized by a prescribed processing of materials for a finite duration. Successful operation means tracking a prescribed recipe with a high degree of reproducibility from batch to batch. Temperature and pressure profiles are implemented with servo-controllers, and precise sequencing operations are produced with tools such as programmable logic controllers.

The main characteristics of batch processes—flexibility, finite duration, and nonlinear behavior—are associated with both their success and their incompatibility with the usual techniques for monitoring and control. However, disturbances and the absence of online quality measurements often affect the reproducibility of batch processes. Nomikos and MacGregor (1994) propose SPC schemes for batch processes, based on MPCA, that use the information of the on–line measurements directly to systematically and scientifically recognize significant deviations from normal operating behavior of a process. Analogous to the prior example, an empirical model, based on the MPCA analysis of data obtained when the process is operating well, is used to characterize the normal behavior of the process. The evolution of the future batches is then monitored by comparing them against this MPCA model using the statistical control limits developed from the reference database. Kosanovich et al. (1994) discuss the application to an industrial batch process.

16.3.2.1 MPCA Method

MPCA is an extension of PCA to handle data in three-dimensional arrays. The three dimensions arise from batch trajectories that consist of batch runs, variables, and sample times. These data are organized into an array $\mathbf{X}$ of dimension $(I \times J \times K)$ where I is the number of batches, J is the number of variables, and K is the number of time samples over the duration of the batch. MPCA is equivalent to performing ordinary PCA on a two-dimensional matrix formed by unfolding $\mathbf{X}$ so that each of its vertical slices contain the observed variables for all batches at a given time. In this approach, MPCA explains the variation of variables about their mean trajectories. MPCA decomposes $\mathbf{X}$ into a summation of the product of t-score vectors and p-loading matrices, plus a residual matrix ($\mathbf{E}_x$) that is minimized in a least-squares sense,

$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{t}_r \bigotimes \mathbf{P}_r + \mathbf{E}_x$$

and R is the number of principal components used in the analysis*.

This decomposition summarizes and compresses the data, with respect to both variables and time, into low dimensional score spaces. These spaces represent the major variability over the batches at all points in time. Each p-loading matrix summarizes the major time variations of the variables about their average trajectories over all the batches. By doing this, MPCA utilizes not just the magnitude of the deviation of each variable from its mean trajectory but also the correlations among them. The appropriate number of principal components may be found by cross validation.

To analyze the performance of a set of batch runs, an MPCA analysis can be performed on all the batches and the scores for each batch can be plotted in the space of the principal components. All batches exhibiting similar time histories will have scores which cluster in the same region of the principal component space. Batches that exhibit deviations from normal behavior will have scores falling outside the main cluster, but batches with similar behavior will cluster in the same region.

* $\bigotimes$ is the tensor product.

16.3.2.2 Process Description

The chemical process from which data are taken is a single batch polymer reactor (Kosanovich et al., 1994). The critical properties that must be controlled for the final product are related to the extent of reaction (e.g., molecular weight distribution). The product's critical properties are determined by off-line chemical measurements and not every batch has its critical properties measured. The property measurement results are available 12 h or more after each the completion of the batch. These results cannot be used in a timely fashion to compensate for poor product quality. Furthermore, it is often difficult to establish the root cause of property deviation when a bad batch is manufactured.

The total time for the batch cycle is less than 2 h. For this analysis, we can consider the two-dominant phenomena, vaporization and polymerization, that produce the polymer. During the first part of the batch cycle, the solvent is vaporized and removed from the reactor; this takes approximately 1 h. In the latter part, polymerization occurs to attain a desired molecular weight distribution. The finished product is then expelled under pressure from the vessel to complete the cycle. The total batch time is monitored carefully and under the continuously supplied external heat source is the main control knob. There is also statistical control of the vessel temperature. Known sources of variability from batch to batch within a product type are variations in the heat content of the heat source, various levels of impurities in the ingredients, and residual polymer buildup over the operating life of a reactor.

16.3.2.3 Analysis and Results

Data for 50 batches made in the same reactor, for the same batch recipe, are collected at 1-min intervals. The database variables contain information about the state of the reactor (temperatures, pressures) and the state of the external heat source. This is not unusual as temperatures and pressures reflect the progress of the reaction in the vessel. Initial analysis indicates that changes in the level and quality of the external heat source result in clustering within the score space. A score plot of principal components 1 and 2 is shown in Figure 16.5. To eliminate that effect, only those batches with the same setting are studied. The data are further segregated into two groups reflecting vaporization and polymerization. In the interest of space, only analysis of the vaporization stage is discussed.

Applying MPCA to the vaporization stage data reveals that the first direction of variability is related to the reactor temperature rise and the second, to the quality of the heat supplied by the heat source. These two principal components explain approximately 55% of the total variance. It is not surprising that heat effects dominate since boiling is the primary event, and the boiling rate and subsequent temperature rise are dependent on the heat transfer rate from the heat source to the reactor content. What is learned

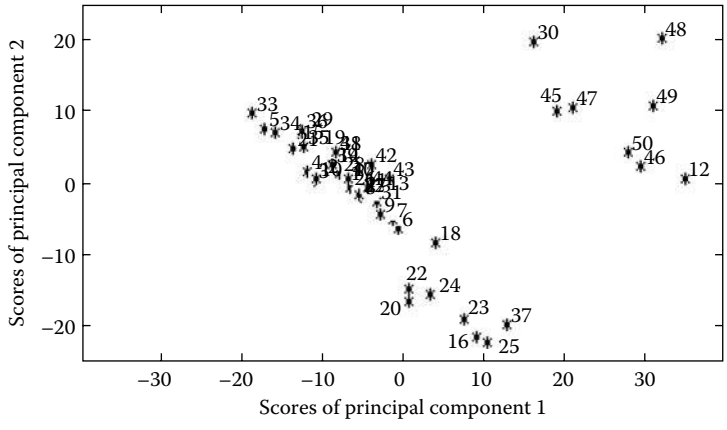

FIGURE 16.5 Score plot of the first two principal components for 50 batches using first stage data.

however, is that for a constant heat input and similar reactor initial conditions, the rate at which the reactor temperature rises, differs from batch to batch. This implies that some batches may boil more quickly than others and that maintaining the prescribed boiling time may vaporize more than just the solvent, thereby affecting the final polymer composition. A control strategy that forces the batch to follow a prescribed reactor center temperature profile rather than a timed sequence should reduce the variability in the temperature rise from batch to batch. Additional work by Kosanovich and Schnelle (1995) on the same process supports this recommendation.

16.3.3 Inferential Control

Multivariate statistical methods can be used to develop nonparametric models for use in feedback control. This is useful particularly in cases where instruments are not available to provide on-line measurements of the quantity to be controlled. One example is composition measurement. Models can be used to predict the information needed, and first-principles models are best suited for this. However, such models are time consuming to develop and are generally not available. Multivariate statistical models are an alternative to this (Ljung, 1989). One can develop a nonparametric model to predict the quantity to be controlled and use this as the measurement in a closed-loop configuration (Kresta et al., 1991). Alternatively, closed-loop control can be formulated and implemented within the reduced space defined by a PCA model (Kasper and Ray, 1992; Piovoso and Kosanovich, 1994).

16.3.3.1 PCA/PCR Controller Design

A controller design is proposed that is based on a PCA model. Within the chemical industry, processes involve many stages, all of which influence the final product properties. Maintaining these properties at their desired specifications requires good control of all the stages. Processes of this type are characterized as having a large number of exogenous variables and a few manipulated ones per process stage. The exogenous variables are used to indicate the process state, and the manipulated variables are used to control indirectly unmeasured quantities. This approach provides no automatic mechanism to adjust controller setpoints when disturbances occur. Careful monitoring of these variables will allow one to correlate to some relative degree good and bad product properties with the exogenous variables' variations. To the extent that quality data can be obtained, a relationship between the exogenous and the quality variables can be used to define an acceptable region of operation. When the operation is inside this region, the process is functioning as it did when good product properties were observed. If the process is deviating from the desired process region, suitable control action based on the relationship between the exogenous and the manipulated variables may be calculated to return the process to within the desired process region.

More precisely, we can develop a PCA model to represent the desired process region in the score space, and then design a controller in the score space that maintains operation within this region. The control moves in the score space are then mapped to the real variable space and implemented on the process. In this fashion, the process is kept within the desired region provided that the PCA model has established correctly the relationships between the exogenous variables and the manipulated ones. This proposed control formulation is analogous to modal control. A high-purity binary distillation column operated at atmospheric pressure is selected as an example to explain the development of the controller. This specific control objective is to maintain the distillate product purity, x_d at 99.5%; the tray temperatures are the exogenous variables, $\mathbf{X}_{ex}$, and reflux rate is the manipulated variable, x_{mp}.

Let $\mathbf{X}$ be composed of two types of variables $\mathbf{X}_{ex}$ and $\mathbf{X}_{mp}$. For various operating conditions, we have

$$\mathbf{X} = [\mathbf{X}_{ex} | \mathbf{X}_{mp}] \tag{16.24}$$

We begin with a development of a PCR model, which in this case yields,

$$[\mathbf{X}_{ex}|\mathbf{X}_{mp}] = \mathbf{T}\mathbf{P}^T + \mathbf{E}_x \tag{16.25}$$

$$x_d = \mathbf{T}q^T + f_y \tag{16.26}$$

The equivalent controller setpoint in the score space is determined from $x_{d,sp}$ by

$$t_{sp} = x_{d,sp}(q^T)^\dagger \tag{16.27}$$

where $(q^T)^\dagger$ is the pseudoinverse of q^T, a (A × 1) vector. The score vector, t can be computed from the projection of x onto the matrix of eigenvectors **P**

$$t = x\mathbf{P} \tag{16.28}$$

Define $\Delta t = t_{sp} - t$ in the score space as the error between the desired score setpoint and the scores associated with the vector, x at a sample time. Conversely, the error in the score space can be reconstructed as an error in the **X** space by

$$\Delta x = \Delta t \mathbf{P}^T \tag{16.29}$$

The temperature variables cannot be manipulated arbitrarily; only the reflux rate can be changed to drive the process in a direction that produces a new x vector so that $t \rightarrow t_{sp}$. In the **X** space, this implies that the required changes in the reflux rate ought to drive the temperatures toward the values that produce $x_{d,sp}$ (t_{sp}). From the score space perspective, the manipulated variable changes must be determined so as to generate changes in the exogenous variables consistent with remaining in the desirable part of the score space. To achieve this, the relationship between the exogenous and manipulated variables in the score space must be defined.

Consider the partition of the matrix of eigenvectors **P** as

$$\mathbf{P}^T = [\mathbf{P}_{ex}|\mathbf{P}_{mp}] \tag{16.30}$$

$\mathbf{P}_{ex}$ is an $(A \times r)$ matrix, where r is the number of exogenous variables, and $\mathbf{P}_{mp}$ is a $(A \times (m-r))$ matrix, where $(m-r)$ is the number of manipulated variables. The relationship between the exogenous and the manipulated variables can be found from the relationship between $\mathbf{P}_{ex}$ and $\mathbf{P}_{mp}$ as follows:

$$\mathbf{P}_{mp} = \mathbf{P}_{ex}\Lambda \tag{16.31}$$

where Λ is a $(r \times (m-r))$ matrix of coefficients that defines the relationship between the manipulated and the exogenous variables in the score space.

The justification for the linear relationship, Equation 16.31, lies in the fact that the eigenvectors define a hyperplane within the **X** space. Because the data are mean centered, the hyperplane goes through the origin and the eigenvectors, p_j^T are unit vectors that lie on the hyperplane. Hence, p_j^T defines a point on this hyperplane. Solving the above equation for Λ yields,

$$\Lambda = \mathbf{P}_{ex}^\dagger \mathbf{P}_{mp} \tag{16.32}$$

where $\mathbf{P}_{ex}^\dagger$ is the pseudoinverse of $\mathbf{P}_{ex}$.

16.3.3.2 Implementation

The PCA/PCR controller would function in the following way. Given $x_{d,sp}$ the corresponding t_{sp} can be determined using Equation 16.27. Similarly, given x, t is found from Equation 16.28. The difference between t and t_{sp} represents the desired change in the scores, Δt which can be used to generate a corresponding Δx (Equation 16.29). Using the Λ matrix from Equation 16.32, the change in the reflux rate that would drive future changes in tray temperatures closer to zero so that $x_d \rightarrow x_{d,sp}$ is found.

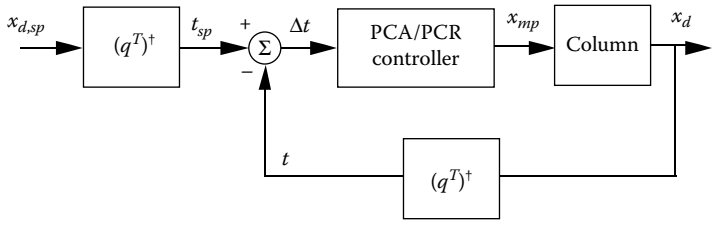

FIGURE 16.6 Block diagram of PCA/PCR controller.

Because this is a steady-state model, the resulting controller moves may be exceptionally large, violating constraints in the **X** space. Thus, only a fraction of the change can be implemented. With no knowledge of the process dynamics built into the model, this fraction becomes a tuning parameter. If the control interval is long compared to the plant dynamics, all of the computed changes could be implemented. On the other hand, if the control interval is short compared to the time constants of the plant, the fully implemented calculated control move might be too large because the plant will not be able to respond fast enough. Figure 16.6 illustrates this scheme.

16.3.4 Binary Distillation Column

The distillation column is by far the most commonly studied process in the Chemical Engineering literature. In this work, we attempt to estimate and control the distillate composition of a binary column to a 99.5% purity. For our purposes, the following assumptions are made: constant molal overflow (CMO), 100% stage efficiency, and constant column pressure. A simple, linear liquid hydraulic relationship between the liquid leaving the *n*th tray and the holdup on the *n*th tray is assumed (Luyben, 1990). There are a total of 20 trays, the condenser is a total condenser, and the reboiler is modeled as another tray in the column. The vapor mole fraction is obtained by a bubble point calculation, and we assume ideal vapor phase and Raoult's law. Two proportional-plus-integral (PI) controllers are used at both ends of the column to maintain the inventory in the reflux drum and the level in the bottom of the column. Two additional PI controllers are used to control the distillate and bottoms compositions in an L/V configuration. Nominal conditions for the column are: reflux ratio of 2, and the distillate/feed ratio of 0.5.

16.3.4.1 Composition Estimation

It is typical to control the endpoint compositions using the temperature on a selected tray, one in the rectifying section, the other in the stripping section. In most situations, this is effective; however, there are cases which are not pathological, that show that the relationship between a single temperature and product composition cannot account for changes in feed composition and the product changes at the other end of the column. In most industrial columns, temperature measurements are available at more than one location; it would seem prudent to make use of all observed system data to infer the endpoint compositions. In this way the dependence of a single tray temperature is removed and the use of process information maximized. Mejdell and Skogestad (1991) propose the use of such a composition estimator using PCR or PLS to estimate endpoint compositions in a binary and a multicomponent distillation column. Their results indicate improved estimates of the product compositions and robustness to measurement noise. Their work demonstrates the use of static estimators for dynamic control when no composition measurements are available.

The data sets are the steady-state temperature and distillate profiles obtained from a four factor, five-level designed experiment where feed rate, feed composition, vapor and reflux rates are varied. Variations on the order of ±20% changes in feed composition and flow rates are contained in the data. In addition, the data are collected with an uncorrelated, uniformly distributed noise of ±0.2°C on the temperature measurements. A PLS model with three loading vectors using only eight tray temperatures is sufficient

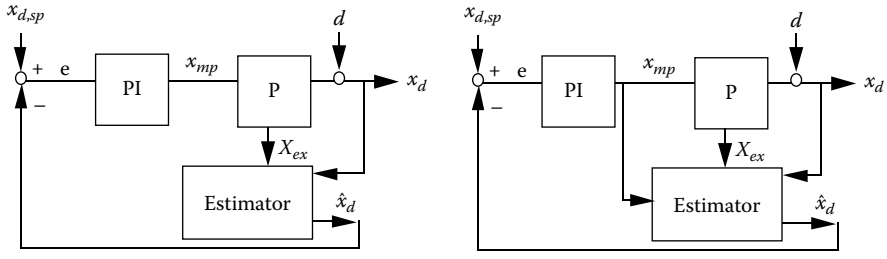

FIGURE 16.7 Feedback control structure with temperature only (left) and temperature-reflux (right).

based upon cross-validation statistics, to explain 98% of the total variance in the distillate composition. This PLS model based on tray temperatures is used only to provide an estimate of the distillate composition for feedback control using a PI controller (see Figure 16.7 (left)).

It is conceivable to develop a composition estimator based on temperatures and reflux (see Figure 16.7 (right)). In the PLS model, reflux is not scaled as the temperatures since it is not the same type of measurement (Kresta, et al., 1991). Only mean centering is applied to the reflux data, and the data are obtained as described previously. As before, a PLS model with three loadings explains 97% of the total variance in the composition.

Both models predict the distillate composition adequately. The temperature-reflux prediction model is not as accurate as the temperature only estimator, which is consistent with the findings of Mejdell and Skogestad (1991). For this reason, they did not consider the contribution of manipulated variable measurements to the development of the composition estimator. Feedback control using either of these two estimation schemes for distillate composition setpoint and input load changes results in satisfactory prediction of composition and good closed-loop performance.

16.3.4.2 PCA/PCR Estimation and Control

To apply the PCA/PCR controller formulation to control the column requires that a PCA model be developed from data composed of temperatures and reflux information. The scores of that model are then regressed on the distillate composition information (PCR). By cross validation, it is determined that three principle components are needed to explain 96% of the total variability in the distillate composition. The controller is implemented as discussed in Section 16.3.3.2. This provides integral-only controller. The response of the system to both a setpoint change and a load disturbance is more aggressive as compared

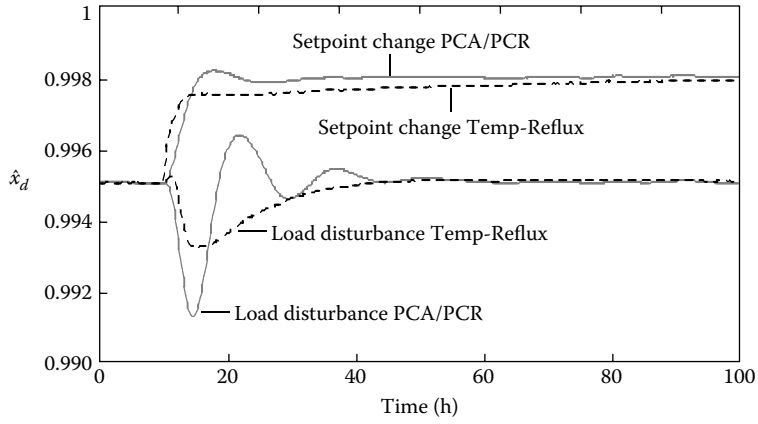

FIGURE 16.8 Estimated distillate composition, PCA/PCR controller scheme.

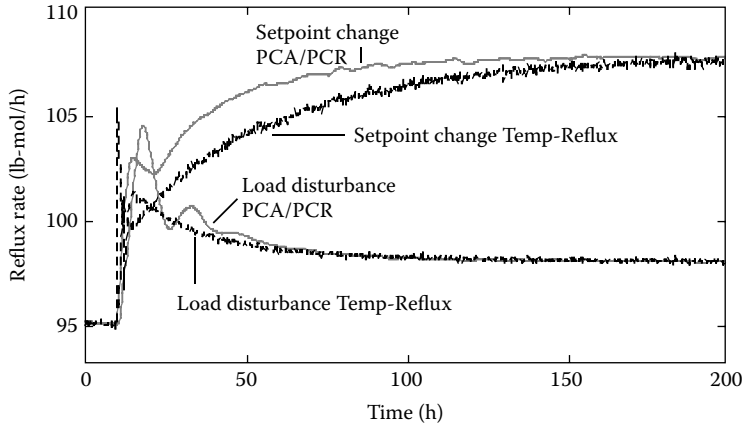

FIGURE 16.9 Reflux rate, PCA/PCR controller scheme.

to the temperature only/PI and temperature-reflux/PI cases. The Ziegler–Nichols tuning methodology is used to set the PI controller parameter values. Figures 16.8 and 16.9 summarize the system's response in both the PCA/PCR controller and the temperature-reflux/PI controller studies. The PCA/PCR controller case produce less noise in the manipulated variable than does the temperature-reflux/PI controller case. An appropriate fraction of the control move can be implemented in situations where the controller action is too aggressive. For the distillation column we would trade-off speed of response with noise in the control moves and in the estimate of the distillate composition. In some situations, if the portion of the control move implemented is too large, the closed-loop system may become unstable. Conversely, if it is too small, the response might be too sluggish. Constraints on the size of the manipulated variable, or on the rate of change of the manipulated variable can be incorporated in a straightforward fashion for any of the models discussed here.

16.4 Summary

Data are gathered in many chemical processes at a very high rate. Unfortunately, much of that data is analyzed infrequently unless a major operating problem occurs. This chapter presents multivariate statistical techniques that can be used to reduce the vast array of numbers into a smaller, more meaningful set, deal with noise, collinearity, and provide a basis for improved control.

Three such techniques are presented: PCA, PCR, and PLS. A generic example is provided to highlight the utility of these techniques and to compare it to the more traditional MLR. A real industrial example illustrates the combined use of PLS and PCA for monitoring and detection of abnormal events. Data are first analyzed to determine what is normal variability in the process. Subsequently, models are developed which define the variability in a compact way. The information in the model is then used to classify the current process state as to whether it is a member of the class of normal variations. If it is not, information (not root cause) about why the data are not of the expected form is made available to the operator to allow for possible corrective action. Additional discussion on the use of the Mahalanobis distance is provided as a discriminant statistic.

An extension to PCA, MPCA is developed for batch processes. This is necessary to handle the three-dimensional nature of batch data. Its usage is demonstrated on an industrial batch polymer reactor to provide new insights, to corroborate existing process knowledge, and to propose meaningful controls improvement. The analysis indicates clearly that a division of the data into sets that correspond to the two major chemical phenomena will provide clarity of information within the data and allow for

interpretation based on process understanding. Doing so leads to the identification of the principal directions of variability associated with each phenomenon and suggests where to improve the existing control strategy.

A novel controller design using PCA and PCR is presented and demonstrated on a high purity distillation column. Its development is based on producing manipulated variable moves that are a function of the exogenous variables that produce a set of scores consistent with being in the score space of the PCA model. Such a controller belongs to the class of modal controllers. The results on the distillation column, when the entire control action is implemented, show that the controller action is aggressive with less noisy estimates and faster settling times. Implementing a fraction of the controller action gives comparable results when compared to the temperature-reflux/PI controller case.

References

Geladi, P. and Kowalski, B. 1986. Partial least-squares regression: A tutorial, *Analytica Chimica Acta* 185:1–17.

Hyvarinen, A. 1999. Survey on independent component analysis, *Neural Computing Surveys* 2:94–128.

Kasper, M. H. and Ray, W. H. 1992. Chemometric methods for process monitoring and high performance controller design, *AIChE Journal* 38:1593–1608.

Kiviluoto, K. and Oja, E. 1998. Independent component analysis for parallel financial time-series, *ICONIP'98* 2:895–898.

Kosanovich, K. A., Piovoso, M. J., Dahl, K. S., MacGregor, J. F., and Nomikos, P. 1994. Multi-way PCA applied to an industrial batch process, *Proceedings of the American Control Conference*, Baltimore, MD, 2:1294–1298.

Kosanovich, K. A. and Schnelle, P. D. 1995. Improved regulation of an industrial batch reactor, *Spring AIChE Conference*, Houston, TX.

Kresta, J. V. and Macgregor, J. F. 1991. Multivariate statistical monitoring of process operating performance, *Canadian Journal of Chemical Engineering* 69:35–47.

Lee, J., Yoo, C., and Lee. I. 2004. Statistical process monitoring using independent component analysis, *Journal Process Control* 14:467–485.

Ljung, L. 1989. *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ.

Luyben, W. L. 1990. *Process Modeling, Simulation and Control for Chemical Engineers*, McGraw-Hill, New York, NY.

Martens, H. and Næs, T. 1989. *Multivariate Calibration*, John Wiley & Sons, New York, NY.

Mejdell, T. and Skogestad S. 1991. Estimation of distillation composition from multiple temperature measurements using partial least squares regression, *Industrial Engineering and Chemical Research* 30:2543–2555.

Nomikos, P. and MacGregor, J. F., 1994. Monitoring of batch processes using multi-way PCA, *AIChE Journal* 40:1361–1375.

Piovoso, M. J. and Kosanovich, K. A. 1994. Applications of multivariate statistical methods to process monitoring and controller design, *International Journal of Control* 59(3):743–765.

Piovoso, M. J., Kosanovich, K. A., and Yuk, J. P. 1992a. Process data chemometrics, *IEEE Transactions on Instrumentation and Measurements* 41(2):262–268.

Piovoso, M. J., Kosanovich, K. A., and Pearson, R. K. 1992b. Monitoring process performance in real-time, *Proceedings of the American Control Conference*, Chicago, IL, 3:2359–2363.

Shah, N. K. and Gemperline, P. J. 1990. Combination of the Mahalanobis distance and residual variance pattern recognition techniques for classification of near-infrared reflection spectra, *American Chemical Society* 62:465–470.

Vigario, R. 1997. Extraction of ocular artifacts from EEG using independent component analysis, *Electroencephalography and Clinical Neurophysiology* 103(3):395–404.

Wold, H. 1982. Soft modelling. The basic design and some extensions, in *Systems under Indirect Observations*, editors K. Jőreskog and H. Wold, Elsevier Science, North-Holland, Amsterdam.

Wold, S., Esbensen, K., and Geladi, P. 1987. Principal component analysis, *Chemometrics and Intelligent Laboratory Systems* 2:37–52.

Wold, S., Ruhe, A., Wold, H., and Dunn, W. 1984. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses, *SIAM Journal on Scientific and Statistical Computing* 5:735–743.

Yoo, C. K., Lee, J., Vanrolleghem, P. A., and Lee, I. 2004. On-line monitoring of batch processes using multiway independent component analysis, *Chemometrics and Intelligent Laboratory Systems* 71:151–163.

17

Plantwide Control

Karlene A. Hoo
Texas Tech University

17.1 Introduction

An industrial plant, continuous or batch, is a network of connected processing units that is used to manufacture multiple commodity products. The terms "plant" and "process" are almost synonymous in the process control community. And a system usually means the plant and the peripheral supporting infrastructure such as the control hardware, steam generators (utility), water treatment facility and so forth. Traditionally, process design and controller synthesis are carried out sequentially. That is, the process consisting of connected unit operations, is designed to satisfy the design parameters (product rates, purity specifications, feed conditions) then the control strategy is determined to regulate the process at the operating conditions associated with the design parameters. Historically, this serial approach to process design and plantwide control has been successful, however, with the ever-increasing demand for pollution preventing plants, sustainable processes, and low environmental impact, this approach can be improved in such a way that the issues of dynamic operability, stability, and controllability can be factored into the design and the control strategy from the outset.

Process design is concerned with the design of the process while plantwide control focuses on the control system. The complementary effects between the design and its control system have been pointed out by [1]. For the control system to work for a specific design, not only should the steady-state economic performance of the design be optimal, but also the open-loop dynamic performance of the design should be operable, stable, and controllable. In other words, process design (a steady-state concept) and operability (dynamic and steady state features) should be considered simultaneously to set limits on the achievable closed-loop performance. This remark is not surprising since a process by its very nature is nonlinear, of high dimension, with complex interactions among the connected unit operations, and constrained by safety and environment regulations.

While the design task is to construct a flowsheet of connected unit operations that achieve the designed objectives including being economically viable, the plantwide control task is to design controllers for a deconstructed flowsheet that when reconstructed satisfactorily regulates the controlled

variables while minimizing interactions. How to best deconstruct the flowsheet becomes an additional task when considering plantwide control design.

Many different plantwide control methods have been proposed in the open literature but no single approach is practiced by the process industry. A possible reason is the variety of chemical processes; thus the expectation that a single foundational plantwide control method should fit all processes may be too optimistic. Perhaps more than one plantwide control methodology is more realistic with the approach subdivided according to the dominant phenomena associated with the chemical process (inorganic synthesis, cryogenic, etc.). This chapter is organized as follows. Section 17.2 provides a general review of the plantwide control literature. Section 17.3 expands on the better known methods and provides an illustrative example, the synthesis of dimethyl ether (DME). Finally, Section 17.4 summarizes the chapter and provides some recommendations.

17.2 Overview of Plantwide Control Research

Traditional approaches to process design and controller synthesis do not guarantee that the process design is controllable even at the designed conditions. The basic objective of plantwide design and control is to consider the steady-state design of the process and its dynamic performance simultaneously. In other words an integration of the process design and the controller synthesis. However, even though the advantages of investigating the integrated design problems from a plantwide perspective can be appreciated, the formulation and solution of such a large-scale problem have a high degree of difficulty.

In this work, the approaches that appear in the open literature can be divided loosely into two categories: heuristics (other acronyms include experiential, rules-of-thumb, and engineering judgement) and nonheuristics (mostly a mathematical framework). In the first category, these approaches for the most part are based on rich and varied industrial experiences and professional judgments of many practitioners. In the second category, the conclusions rely on linear and nonlinear theories, optimal theory, statistics and probability. Methods in this category investigate two dynamic subproblems, the *open-loop system* and the *closed-loop system*. The former is concerned with the dynamics of the uncontrolled process design while the latter considers the process and its designed control strategy. Of great importance in either analysis are the number of control degrees of freedom and closed-loop stability.

An overarching format that is practiced by these two categories is the implicit or explicit deconstruction of the large dimension of the chemical flowsheet into smaller groups of connected unit operations to reduce the dimension of the plantwide controller design. The decomposition can be functional, structural or a combination of both. Further the subgroups do not have to follow the process stream directions.

17.2.1 Heuristic Methods

Industrial experiences and professional judgments have accumulated since manufacturing at the large-scale signified an economically successful nation and also to address global competition. Essential features of a large-scale batch or continuous process include the variety and sizes of the unit operations, the interconnections among them and the sheer volume of material and utilities consumed and/or produced. Since the common practice is to design the chemical process first (the result is a flowsheet that satisfies the steady-state designed conditions) and then implement a control strategy, it is not surprising that some regulation issues may not be solvable by the control structure; sometimes the issues are a function of the constraints imposed by the process design itself. A closer inspection of heuristic methods reveals that the conservation of mass, energy and momentum are being applied under the disguise of product quality and production rate specifications, energy management, safety, and economics (continuous operations).

17.2.1.1 Procedural Rules

Procedural rules are steps to formulate the process design and control objectives. For example, since economics is a high priority for the viability of a process, there is an explicit emphasis placed on product quality. Product quality usually is a result of the reactor and separator unit operations.

Significant contributions to this category have been made by Luyben and coworkers [2–7]. The genesis of their rules is the conservation of material and energy. Their nine-step procedure to the synthesis of a plantwide control strategy is as follows [2,4,8–10]:

1. Establish the control objectives.
2. Determine the control degree of freedoms.
3. Establish an energy management system.
4. Regulate the production rate.
5. Regulate the product quality.
6. Directly regulate the flow rate in every recycle loop and control inventory.
7. Perform a component balance for each chemical component.
8. Regulate individual unit operations.
9. Optimize the economics and/or improve dynamic controllability.

These nine steps reveal that both material and energy are conserved and effects such as the "snowball effect" will not occur in the recycle loops [10]. The snowball effect is avoided because the recycle flow rate is explicitly regulated for each material recycle loop; thus the influence of the recycle on the system will not be cumulative.

A similar five step procedure is proposed by Price and coworkers [11,12].

1. Regulate the production rate.
2. Regulate inventories.
3. Regulate product quality.
4. Regulate to remain within equipment and operating constraints.
5. Regulate to improve economic performance.

The first observation between these two different procedural approaches is the order of what is regulated. Clearly, this sets the usage of the control degrees of freedom and once they are assigned they are no longer available. A second observation is that economics is vital and appears in the form of regulation of the production rate and product quality. A final observation is that for continuous operability, liquid inventories are well regulated.

Another procedural approach is that published by Shinnar and coworkers [13,14]. The kernel of this approach is the identification of a set of variables such that when regulated in the presence of an expected set of disturbances, stable closed-loop performance is attained. A closer investigation of this set reveals a tight link to the economics of the process. The identification of this specific set of controlled variables relies on engineering experience and is not readily generalizable. The example system they studied over many times is the complex and nonlinear fluid catalytic cracker (FCC). The FCC is a mainstay in petrochemical plants. Other than the major disadvantage of requiring engineering experience to design the plantwide controller strategy, other disadvantages include no implicit consideration of multivariable interactions, location of the sensors and actuators, complex and nonlinear nature of the unit operations, stability, and dynamic response times.

17.2.1.2 Hierarchical Deconstruction Methods

Since most of the heuristic approaches are based on considerations of economics, safety, and continuous operations; it is not surprising that the mathematical formulations' objective function(s) include these considerations in some form. Since convergence of multiobjective functions is a major obstacle, a research challenge is to deconstruct the process design into manageable pieces. The basis for the decomposition may

be functional (reaction system, separation system), structural (natural stream flow of feed to products), or a combination of both.

Buckley [15] suggested the deconstruction of the plantwide control problem based on material balance and product quality. Both criteria point to an economic basis. Douglas and coworkers propose a five tier functional design hierarchy [16–20]. Since this hierarchy is taught in almost all chemical engineering design courses it is worth restating.

1. Determine if the process to be designed should operate as a batch or as a continuous process. Continuous processes are more common in chemical and petrochemical plants. Batch processes usually are suited for small production of specialty chemicals.
2. Identify all major input and output streams. From a material balance viewpoint, every component entering the system has a path to exit the system.
3. Since no reaction is 100% complete and to promote favorable economics, determine the material recycle streams. It is worth mentioning that recycle streams also create more operability and controllability challenges.
4. The reaction system produces unwanted by-products; a separation system is necessary to obtain the desired product purity.
5. Analogous to material recycle, energy management is driven by economics; and similarly there are regulation challenges not only for operational constraints but also for safety.

An obvious limitation of this approach is that the design decisions are based only on steady-state design objectives. Ponton and Liang [21] propose a modification to the Douglas design hierarchy to include control issues at each design level. They suggest the identification of the control degrees of freedom and the design of a control structure at each level. A limitation of this approach is that the interactions among the levels are not accounted for.

Morari and coworkers apply optimization theory with a particular three-tier deconstruction to the plantwide control problem [22–28].

1. Steady-state optimization is used to optimize the operating conditions that have a large economic impact from a plantwide perspective.
2. The advanced process control level is to determine the set-points of the controlled variables in response to the optimal operating conditions. Often a mathematical model of the process is used in a model-based control strategy to account for interactions, constraints, and nonlinear behavior.
3. The regulation level has the fastest dynamics. Here, the focus is regulating the individual unit operation while minimizing interactions.

Stephanopoulos and coworkers have conceptualized the connection among levels as a nested framework [29–32]. The higher levels form the outer loop of the optimization problem and serve to provide set-points and active constraints for the inner loops. The inner loops feedback the current process state to the outer loop to improve the optimization continuously. In this way, the objective of the whole plantwide optimization problem will be guaranteed by the outer loop, with the objectives of the inner loops adjusted to achieve the overall objective. This framework is based on the concept of a cascade control structure.

In general, heuristic methods have been used successfully to design and control large scale processes. However, heuristics that are associated with one class of processes, for example, hydrocarbon systems, or cryogenic systems, may not be appropriate for another class of processes. And even for the same type of process, domain experts do not always agree on the same heuristics. It is not surprising that the solutions (design or the controller) are not identical but close examination will reveal similarities at some level. By the very nature of heuristics, it is clear that heuristic-based plantwide controller designs may not always be comparable on the same basis.

17.2.2 Mathematical Methods

The rapid development and use of mathematical methods is primarily due to the availability of automated tool sets such as AspenTech's (Houston, TX) AspenOne® product suite, SimSci-Esscor's (Houston, TX) PRO/II™, Process Systems Enterprise's (London, UK) gPROMS®, Mathworks' (Natick, MA) MATLAB® and associated toolbox, Honeywell's (Minneapolis, MN) Profit Suite™, and Control Station's (Tolland, CT) LOOP-PRO™.

However, it should be pointed out that since the majority of these methods are not intuitive and require a thorough understanding of control and process engineering theories, only a small subset of these tools have been used effectively by process engineers. In fact, consultants familiar with these automated tools are usually brought in to design, test, and validate the process design and plantwide control structures.

There are many ways to classify the large number of methods in this category, for example, steady-state and dynamic, open- and closed-loop, time and frequency, model-based and nonmodel-based, and so forth. In this work, we exclude frequency-based methods as most chemical process industries employ time-based methods.

17.2.2.1 Open-Loop Methods

The assumption is that if a controllability analysis can determine the dynamic performance of the process at the designed conditions, the screening of alternate designs will be more efficient and only those designs that are fully controllable will survive. In this work, the working definition of controllability is the ability of the process to achieve acceptable control performance by regulating the controlled variables within specified bounds using the available actuators [33].

This definition of controllability assumes the existence of a control law with which the process can be regulated in the face of known and bounded disturbances. Thus, controllability is a property of the process itself with no particular requirements of what type of control law is applied. Classical control concepts assume full state feedback conditions. Open-loop analysis refers to the dynamic process design behavior when perturbed. Often open-loop methods rely on linear stability theory to establish that the design is a stable design about the designed conditions. Closed-loop analysis involves the process and the plantwide control strategy. Since closed-loop, nonlinear stability is not readily established theoretically, numerical simulations usually are employed to show that the selected control strategy regulates the system to within some prespecified performance measures. Stability does not establish controllability but the converse is true.

Most of the controllability analysis tools are based on linear theory presuming that the process only experiences small perturbations about the stable designed operating conditions. There are several model forms that can be used to represent a linear system. The general nonlinear system can be represented by

$$\begin{aligned} &\dot{x} = f(x,u); \quad y = h(x) \\ \text{s.t.} \quad &\phi(x,u) = 0; \quad \psi(x,u) \le 0 \end{aligned} \tag{17.1}$$

where f and h are vectors of nonlinear, real functions, x represents the states, y are the outputs, u are the inputs, $\phi(\cdot,\cdot) = 0$ are the equality constraints and $\psi(\cdot,\cdot) \le 0$ are the inequality constraints. A Taylor series expansion permits the development of an approximate linear, state-space model from Equation 17.1,

$$\dot{x} = Ax + Bu \tag{17.2}$$

$$y = Cx \tag{17.3}$$

where A, B, and C are time-invariant matrices of the right sizes. The transfer function model can be derived from this linear model and can be used with frequency-design methods. When a model of the system is not available, system identification methods are used to identify input–output models (step-response models, time-series models). A caveat, the system identification approach only can identify the controllable modes.

A quantitative tool that is used often to establish controllability is the *relative gain array* (RGA) [34]. The RGA was introduced as a steady-state measure of the interactions of the manipulated variables onto the controlled variables. The RGA is useful as a first pass tool to validate heuristic selected control and manipulated variable pairings [11]. A number of extensions to the original RGA derivation have appeared in the research literature. For example, the RGA has been extended to nonsquare systems. Stanley et al. [35] developed the steady state relative disturbance gain (RDG) in a similar manner to quantify the impact of disturbances on controllability. To address the dynamic impact of disturbances, Huang et al. and Hovd et al. [36,37] incorporated dynamic information by evaluating the corresponding RGA at all frequencies.

The Niederlinski Index (NI) is another useful measure to analyze the stability of the control loop pairings using only the steady state results of the RGA. A negative NI value indicates instability in the chosen control loop pairing. Guidelines to use the steady-state RGA and NI can be found in [33, 38]. Singular value analysis also has been applied to evaluate the controllability of a process [39–42]. The singular value represents the smallest gain of the process among the available inputs. The greater the singular value the more resilient the process is to disturbances.

A major advantage of examining the open-loop system is based on the assumption that the dynamic and steady-state operability of a plant is designed into the process. That is, a system's ability to react to changes only can be guaranteed if the control requirements are considered at the design stage. It is possible that design decisions made solely on steady-state considerations may impose severe limitations on the operability and controllability of the process. By analyzing the dynamic open-loop design using operability and controllability measures, the limitations imposed by the steady-state design can be identified and if possible altered. From the discussion above, it is not difficult to conclude that although the analytical measures can guide the control design they do not necessarily give the same solution. Some researchers have shown that because the analytical methods are based on different foundations, the resulting control structures from a performance perspective may appear to be in conflict [42,43]. Often, heuristics, costs, and other intangibles are used to select one control solution over another. If an approximate mathematical model of the closed-loop system is available, one can simulate the controller performance under well-controlled conditions.

17.2.2.2 Closed-Loop Methods

Closed-loop methods are those that address the dynamic process and controller designs simultaneously. Narraway and Perkins [44] have developed a systematic design procedure that uses a hybrid mixed integer linear programming (MILP) method to determine both the process design and the regulatory feedback control structure based on a linear economic analysis. By solving the MILP problem, optimistic bounds on the dynamic economic performance of partially or fully specified control structures can be generated. This method was extended to consider parameter uncertainty [45,46] and multivariable controllers [47].

Georgakis et al. [48] applied an optimization-based formulation to determine the input bounds that would guarantee the required output specifications throughout the dynamic response. Swartz [49] used a Q-parametrization of linear feedback controllers to define the upper bound of the performance of any linear stabilizing controller. Bahri [50] applied a back-off approach where the system is moved off from the original "optimal" design by considering the influence of the expected disturbances. The amount reduced is related to both the economic profit as well as the dynamic operability of the design.

17.2.3 Combined Methods

From the above discussion, both engineering knowledge and mathematical analysis are useful in addressing a plantwide design and control problem. Although these methods were introduced from a perspective of heuristics versus quantitative methods, many approaches combine both. Usually, the combined approach employs heuristic knowledge to deconstruct the process flowsheet, followed by some combination of both approaches to design the control strategy.

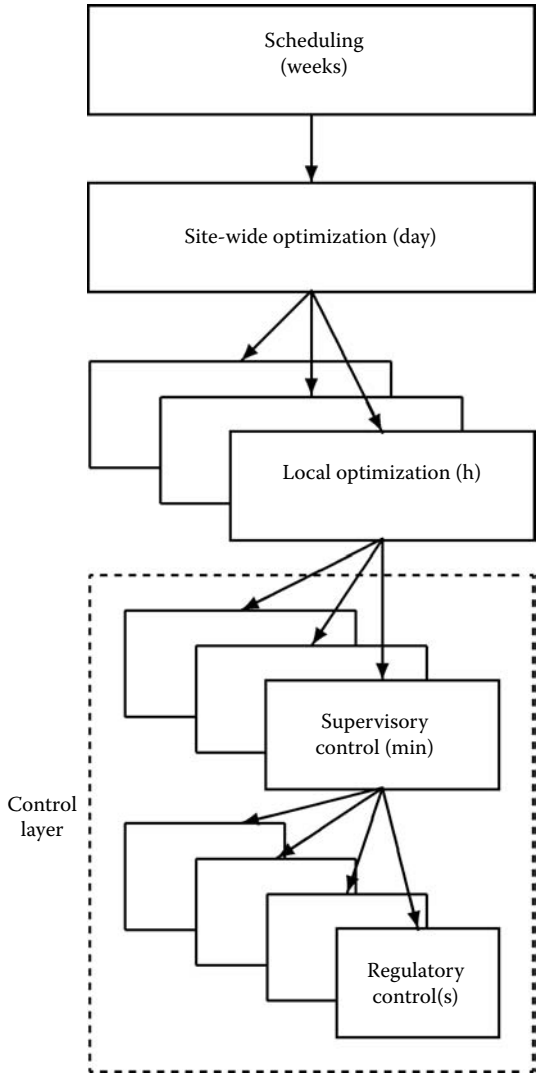

FIGURE 17.1 The control hierarchy of Larsson and Skogestad. (Adapted from T. Larsson and S. Skogestad, *Int. J. Control*, 21(4):209–240, 2000.)

Skogestad and coworkers [33,51–57], contribute the concept of self-optimizing control variables (SOCVs) and the control design hierarchy shown in Figure 17.1.

A variable is said to be self-optimizing if with constant inputs the plant can regulate the controlled variables to within acceptable bounds. The following guidelines are provided to select the SOCVs:

- Robust to known disturbances.
- Easy to measure and control.
- Sensitive to changes in the manipulated variables.
- Independent of each other.

A loss function is formulated and optimized to assess the choice of the SOCVs in the presence of a set of expected disturbances. This method has some similarities with the partial control concept proposed by Shinnar and coworkers [13,14]. A common issue is the selection of the expected set of measured disturbances.

Vasbinder and Hoo [58] proposed and developed a decision-based methodology called *modified analytic hierarchical process* (mAHP). This method relies on a particular process flowsheet decomposition to group the unit operations according to steady-state and dynamic operability requirements. Once the subgroups are identified any controller design method can be used to develop the control structure for each subgroup.

17.3 Simulation Example: Dimethyl-Ether Process

This section introduces an instructional process example to demonstrate some of the issues discussed above but also to demonstrate the development of a plantwide control strategy.

DME is widely used as a propellant and also can be used as a refrigerant, solvent, and fuel in various industries. There are many ways to produce DME; here we demonstrate its production starting with a pure methanol (MeOH) feedstock [58]. The process flowsheet as provided by Turton et al. [59] show typical chemical process unit operations. The material recycle and heat integration are for purely economic reasons. Here, we employ a somewhat modified process flowsheet shown in Figure 17.2 and we design a decentralized control structure that applies experiential knowledge to design the basic control structure and to select the set of expected disturbances. Then we apply the SOCV method to select the control variables (CVs). Other approaches can be used but the emphasis here will be on analysis of pre- and post-control performance. Note, that the flowsheet and subsequent results are found using AspenPlus® engineering suite 2006.

17.3.1 Steady-State Analysis

The description of this process was adapted from [59]. The production from MeOH to DME is typically by catalytic dehydration of MeOH. The main reaction is

$$2CH_3OH \rightarrow (CH_3)_2O + H_2O \tag{17.4}$$

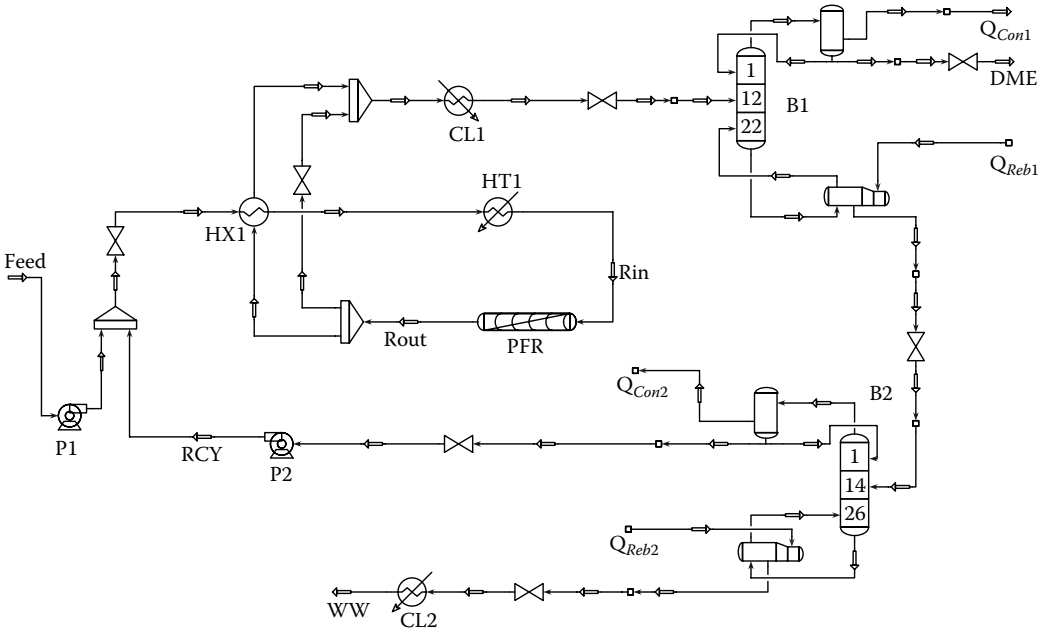

FIGURE 17.2 Process flowsheet of the DME steady-state design.

TABLE 17.1 DME Operating Conditions

Parameter	Value		
k	1.21×10^6 kmol/m^3-h		
E_a	8.048×10^4 kJ/kmol		
R	8.314 J/mol-k		
	Temperature	25°C	
	Molar flowrate	260.0 kmol/h	
Feed	Mass flowrate	219.44 tons/day	
	Pressure	101 kPa	
		MeOH	99.4
	Compositions wt%	H_2O	0.6

This reaction is exothermic. The operating conditions are: inlet temperature of 250°C and a pressure of about 1.5 MPa. The kinetic rate expression for this reaction is characterized by an Arrhenius expression,

$$r_{MeOH} = k \exp\left(-\frac{E_a}{RT}\right) P_{MeOH} \tag{17.5}$$

where r_{MeOH} is the MeOH conversion rate, k and E_a are the kinetic rate parameters (see Table 17.1), and P_{MeOH} is the partial pressure of MeOH. It is assumed that no significant side reactions will occur at these operating conditions. The desired product rate is 50,000 tons/year (~124.0 kmol/h) of DME at a product quality of least 99.5 wt% DME. Because the acid zeolite catalyst will be deactivated the reactor temperature should not exceed 400°C. Another restriction is that the purity of the waste water should be at least 99.8 wt% to satisfy environmental regulations.

Fresh MeOH is pressurized to 1.56 MPa from ambient conditions before combining with the recycle stream. The mixture is preheated to 250°C by a feed-heat exchanger and a heater. The designed single pass MeOH conversion is ~80.0%. Two distillation columns are used to obtain a purified DME product and the recycle of unreacted MeOH. The distillate stream which is predominantly unreacted MeOH is recycled to the reactor. Designed operating conditions are listed in Table 17.1.

The design parameters are: pump rates, utility loads, total number of column trays, column feed tray location, length, and diameter of the reactor. The conversion is sensitive to the inlet reactor temperature. If the reactor inlet temperature decreases, the reaction extent and exit temperature will decrease. The feed-heat exchanger may not always provide the required heat duty; hence an additional heater is added and becomes a control degree of freedom to respond to reactor inlet temperature changes (see Figure 17.3).

The steady-state design is solved using AspenPlus 2006 engineering suite; the results are listed in Table 17.2. The steady-state product requirements are achieved by this design. That is, the DME concentration in the product stream and its flowrate are 99.56 wt% and 129.0 kmol/h, respectively; and the water concentration in the water column bottom stream is 99.89 wt%.

We now analyze the dynamic operability of the process from an open-loop perspective employing the controllability measures and methods previously introduced.

17.3.2 Plantwide Control Structure Synthesis

Following the plantwide control design procedure found in [2,8] we arrive at the following analysis.

1. *Step 1:* Qualitative analysis of the control requirements to satisfy DME production.
   At the designed operating conditions, the effect of conversion on the reactor temperatures is shown in Figure 17.4. The MEOH conversion is measured at the exit of the reactor. The exit temperature and the conversion have a nonlinear dependence on the inlet temperature in the range of 240–260°C. To avoid violation of the maximum operating temperature limit (400°C), the outlet temperature should be regulated. The dependence of the MeOH conversion on the outlet

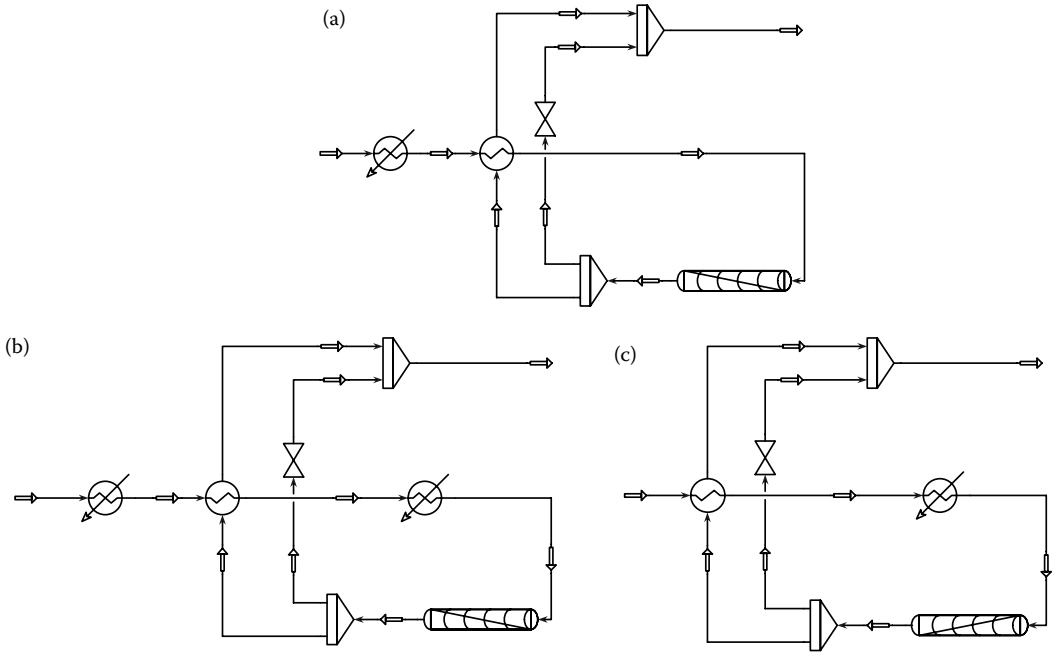

FIGURE 17.3 Configuration comparison of preheating the reactor inlet material in: (a) Turton (Adapted from R. Turton, et al., *Analysis, Synthesis, and Design of Chemical Processes*. Prentice-Hall International, Upper Saddle River, NJ, 1998), (b) Vasbinder (Adapted from E. Vasbinder and K. Hoo, *Ind. Eng. Chem. Res.*, 42:4586–4598, 2003), (c) Current design.

temperature is linear in the range of 300–390°C. It follows that meeting the designed conversion can be accomplished by regulation of the outlet temperature.

The open-loop time constant of the process without material recycle is ∼2.0 h while with material recycle it is 3.5 times greater (∼7.0 h). The dynamics of the process also will be significantly altered

TABLE 17.2 DME Process: Major Equipment and Design Variables

Unit Operation	Variable	Value
Reactor (PFR)	Length, Diameter	10.0 m, 0.72 m
	T_{Rin}, P_{Rin}, T_{Rout}, P_{Rout}	250.0°C, 1.5 MPa, 368.0°C, 13.9 atm
Pump (P1)	Power, Outlet pressure	9.78 kW, 1.56 MPa
Pump (P2)	Power, Outlet pressure	2.45 kW, 1.56 MPa
Heater (HT1)	Q, Outlet temperature	3.59×10^3 kW, 250.0°C
Cooler (CL1)	Q, Outlet temperature	-3.84×10^3 kW, 81.5°C
Cooler (CL2)	Q, Outlet temperature	-0.34×10^3 kW, 50.0°C
HeatEx (HX1)	Heat transfer coefficient, area	50.0 W/m^2K, 40.2m^2
	N_T, N_f[a], Tray diameter, spacing	22, 12, 2.0 m, 0.61 m
Product column (B1)	F_{DME}, Top pressure, Rr	129.0 kmol/h, 1.05 MPa, 0.79
	Q_{Reb1}, Q_{Con1}	1.38×10^3 kW, -1.15×10^3 kW
Water column (B2)	N_T, N_f, Tray diameter, spacing	26, 14, 2.0 m, 0.58 m
	F_{WW}, Top pressure, Rr	131.0 kmol/h, 0.7 MPa, 1.86
	Q_{Reb2}, Q_{Con2}	1.55×10^3 kW, -1.58×10^3 kW

[a]Numbered top to bottom.

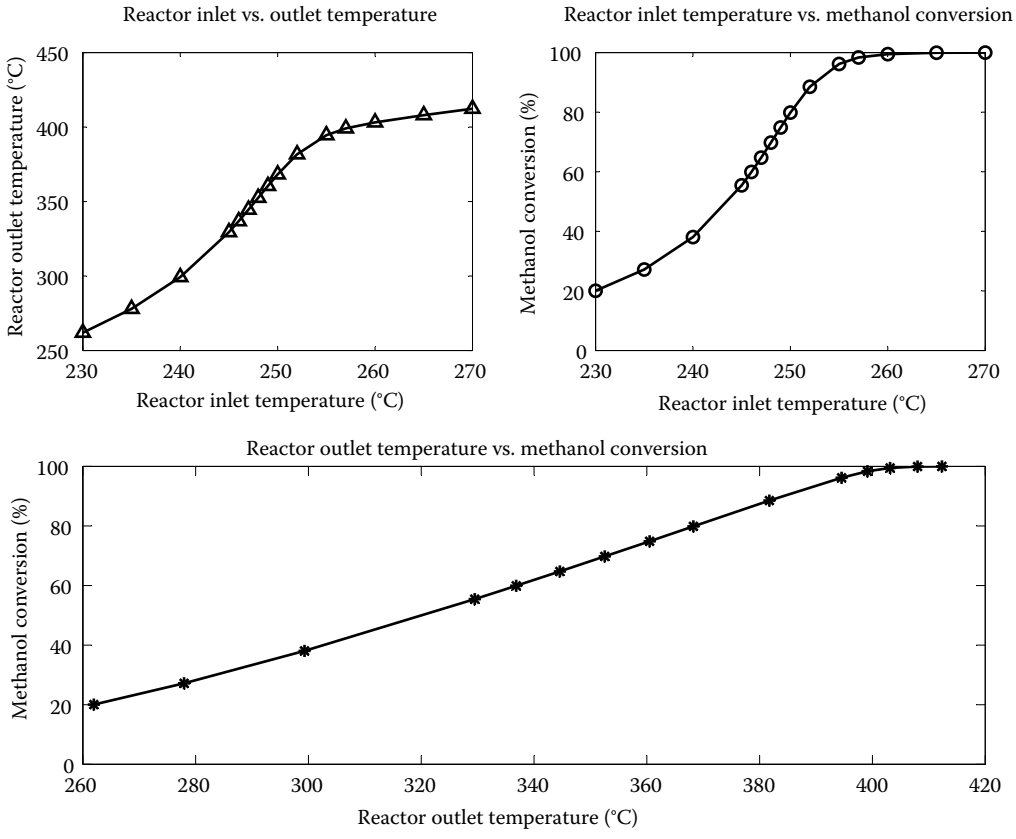

FIGURE 17.4 Analysis of the designed reactor operating conditions.

by heat integration. Due to the feed heat exchanger the inlet temperature is dependent on the exit reactor temperature. To further address perturbations in the exit temperature a bypass is added which provides an additional control degree of freedom.

2. *Step 2:* Identify the controlled variables and the control degrees of freedom.
   Engineering experience indicates that the variables to be controlled are liquid levels as they account for material and all inventories. Additional inventory variables include column pressures and temperatures and material recycle. The following 16 variables are identified as CVs:
   - *Reactor unit operation* (4): inlet and exit temperatures, pressure, and flowrate.
   - *DME column* (6): distillate product purity, feed flowrate, temperature, pressure, and condenser and reboiler liquid levels.
   - *Water column* (5): bottoms product purity, temperature, pressure, and condenser and reboiler liquid levels.
   - *Recycle stream* (1): pressure

   The maximum number of control degrees of freedom should be at least the same as the number of controlled variables. The control degrees of freedom in the design are: seven control valves, seven heating and cooling utilities, and two pumps. The pairing of the controlled and manipulated variables can be accomplished with experiential knowledge supported by quantitative metrics such as steady-state RGA and the NI. Table 17.3 lists the controlled and manipulated pairs for the reactor and product and water columns.

   The variables are grouped based on their related unit operation and an RGA analysis. The first group contains variables related to the reactor; the second and third are associated with

TABLE 17.3 Pairing Controlled and Manipulated Variables

Unit	Controlled Variables	Manipulated Variables
Reactor	Flowrate	Control valve
	Pressure	Pump (P1)
Product column	Flowrate	Control valve
	Temperature	Cooler (CL1)
	Top pressure	Condenser
Water column	Temperature	Cooler (CL2)
	Top pressure	Condenser
Recycle stream	Pressure	Pump (P2)

the product and water columns. The CVs are the reactor inlet and exit temperatures. The selected manipulated variables are the heat exchanger bypass valve and the heater duty. The CVs associated with the product column are the liquid levels of the condenser and reboiler, and the DME distillate composition. The manipulated variables are the distillate and bottoms control valves, and the reboiler duty. The CVs associated with the water column are the liquid levels of condenser and reboiler, and the bottom stream water composition. The manipulated variables are the bottoms stream control valve and the reboiler duty.

The results of the RGA analysis for these three groups are:

$$\begin{bmatrix} & T_{Rin} & T_{Rout} \\ Q_{HT1} & 0.510 & 0.490 \\ F_{By} & 0.490 & 0.510 \end{bmatrix} \begin{bmatrix} & L_{Con1} & L_{Reb1} & x_{DME} \\ F_{DME} & 0.783 & -0.140 & 0.357 \\ F_B^{B1} & -0.140 & 1.745 & -0.605 \\ Q_{Reb1} & 0.357 & -0.605 & 1.248 \end{bmatrix}$$

$$\times \begin{bmatrix} & L_{Con2} & L_{Reb2} & x_{WW} \\ F_{RCY} & 0.692 & 0.528 & -0.220 \\ F_{WW} & 0.528 & 1.928 & -1.456 \\ Q_{Reb2} & -0.220 & -1.456 & 2.676 \end{bmatrix}$$

The recommended pairings are to pair along the major diagonal; however, it is clear that there are significant interactions (off-diagonal terms).

3. *Step 3:* Additional control issues.

The first issue is the pairing of the reactor variables. Pairing by either the major or minor diagonal is not appropriate because the reactor inlet and outlet temperatures are coupled. To address this issue a cascade feedback control strategy is proposed to control the reactor exit temperature by the set point of the reactor inlet temperature. The reactor inlet temperature is regulated by manipulating the duty of the heater. The open-loop time constant of the inner loop is ~4.5 min while that of the outer loop is about 21.5 min.

The heat exchanger bypass remains closed except when the exit temperature exceeds the operating limit of 400.0°C. The closed-loop performance shows that the cascade feedback control structure has the ability to withstand large disturbances (up to a +5.0% step disturbance in the feed rate) in the reactor inlet temperature.

The DME composition controller and the water composition controller suffer from reliable online, real-time composition data. Most often an estimator is developed to infer the composition from measured data. From thermodynamic theory, the composition, temperature, and pressure are correlated. In the column the temperature profile across the trays is an indication of the extent of the physical separation. It is reasonable to use several temperatures to estimate the composition [60]. It is recognized that not all tray temperatures are measured but using those that are available can improve the composition estimates.

To apply the SOCVs method, a dynamic loss function L is defined. Here, L represents a measure of the total amount of off-specification product when the process is subjected to a set of expected disturbances while the set points of the candidate SOCVs are held constant:

$$L = \left| \int_{t_0}^{T_f} F_{DME}(t) \left[x_{DME}(t) - x^*_{DME} \right] dt + \int_{t_0}^{T_f} F_{WW}(t) \left[x_{WW}(t) - x^*_{WW} \right] dt \right| \tag{17.6}$$

where t_0 and T_f represent the initial and final times, respectively; F_{DME} and F_{WW} are the mass flowrates of the product streams DME and WW, respectively; x_{DME} and x_{WW} represent the DME composition by weight in the distillate and bottoms streams of the product and water columns respectively; and x^*_{DME} and x^*_{WW} are their respective set points.

Based on experiential knowledge, the expected set of disturbances are assumed to be change in the feed stream MeOH composition (−4.0 wt%) and throughput (+3.0%). The candidate CVs are chosen from the temperature of the column trays [60]. Dynamic simulations are done to determine the value of L. The simulated results show that the temperature of tray 2 in the product column is the best variable to determine the DME composition and the temperature of water column at tray 26 is the best measured variable to determine the bottom stream water composition.

4. *Step 4:* Plantwide control structure.
   The pairing between controlled and manipulate variables provides a control structure for the process. However, the control strategy needs both the control structure and a suitable control law. A typical control law that has been successful is the proportional-integral (PI) feedback control law [61]. This law can take more than one form [62,63]. This law will be applied for each control/manipulated variable pair. The process flowsheet with the full decentralized PI control structure is shown in Figure 17.5. The tuning parameters (controller gain and integral time constant) of these PI controllers are based on Ziegler–Nichols tuning rules and the internal model control (IMC) tuning rules [62]. The PI tuning constants are listed in Table 17.4.

   Three temperature controllers are used to regulate three compositions. First, controller Rout_TC (see Table 17.4) is the primary controller of the cascade loop to regulate the reactor exit composition. The other two composition controllers are labeled DME_CC and WW_CC. The former is used to regulate the DME product purity in the distillate of the product column and the latter to regulate the water purity in the bottom stream of the water column.

17.3.3 Results and Discussion

Two case studies are provided to demonstrate the suggested plantwide control structure. In the first study, a +5.0% change in the throughput is introduced and in the second study a −10.0% in the feed MeOH composition. Both are introduced 10.0 h after the plant has achieved steady state. Both changes are modeled as a step changes. The closed-loop performance is based on the time it takes for the controlled variables to return to their set points, the magnitude of overshoot, and the settling time (the time it takes to reach within 5% of the final value).

17.3.3.1 Case Study 1: Feed Throughput Change

The closed-loop responses of the reactor inlet and exit temperatures are shown in Figure 17.6. The controller Rin_TC responds to the disturbance by adjusting the reactor inlet temperature by 0.57%. The reactor exit temperature has an average transient of 0.03% but settles 6 h after the disturbance is introduced.

Figure 17.7 shows the composition of the DME product column and the composition of the water stream in the water column. There is an overshoot in the DME product of 0.27% and a settling time of 6 h. In the case of the water purity there is an undershoot of 0.5% with a settling time of 15 h. It is possible that different tuning parameters on the water controller may reduce the settling time.

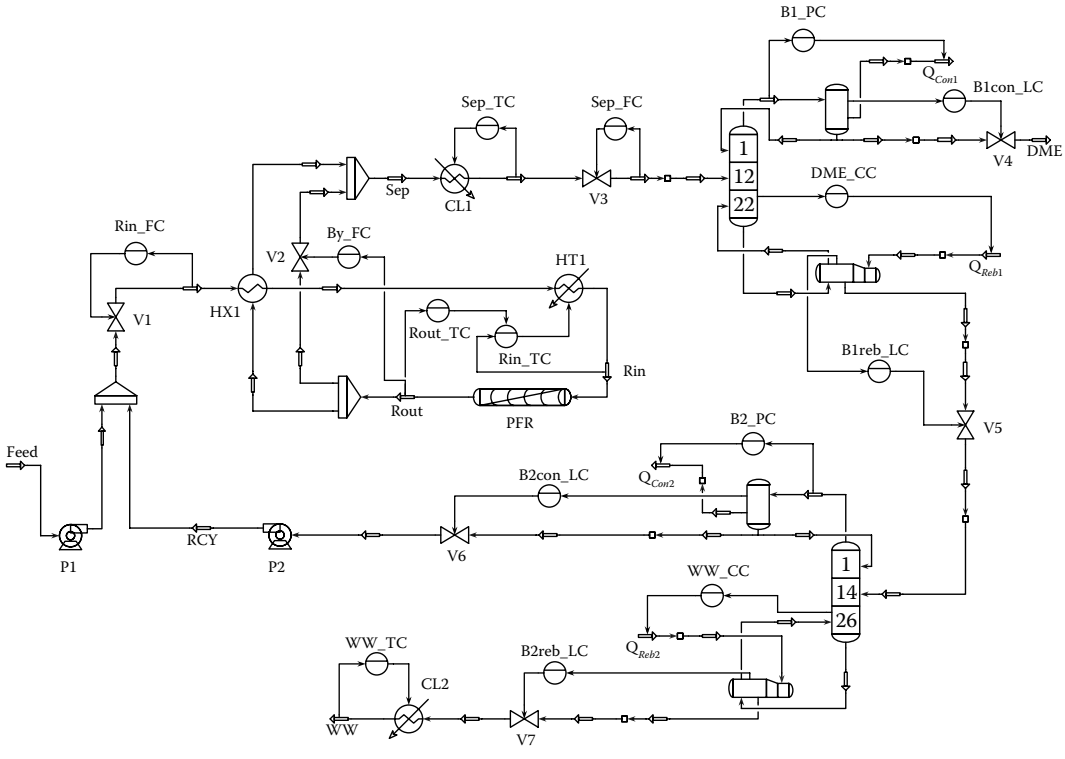

FIGURE 17.5 Decentralized PI control structure. Pressure controllers of streams Feed and RCY are omitted for simplicity.

TABLE 17.4 PI Controller Tuning Values

Controller	CV	MV	K_p	T_I (min)
Rin_FC	F_{Rin}	V1	−6.9[a]	0.7
Rin_TC	T_{Rin}	Q_{HT1}	−2.2	1.6
Rout_TC	T_{Rout}	SP of Rin_TC	−1.3	5.2
By_FC	F_{By}	V2	−12.5	0.9
Sep_TC	T_{Sep}	Q_{CL1}	−0.4	4.0
Sep_FC	F_{Sep}	V3	−13.8	0.8
B1_PC	B1 top pressure	Q_{Con1}	−4.2	28.5
B1con_LC	L_{Con1}	V4	4.4	25.0
DME_CC	T^{2}_{B1} [b]	Q_{Reb1}	−8.5	18.7
B1reb_LC	L_{Reb1}	V5	9.3	14.1
B2_PC	B2 top pressure	Q_{Con2}	−3.1	26.4
B2con_LC	L_{Con2}	V6	3.7	23.2
B2reb_LC	L_{Reb2}	V7	7.6	19.8
WW_CC	T^{26}_{B2} [c]	Q_{Reb2}	−10.2	34.5
WW_TC	T_{WW}	Q_{CL2}	−0.5	4.1

[a] Minus sign: reverse acting.
[b] Temperature of tray 2: product column.
[c] Temperature of tray 26: water column.

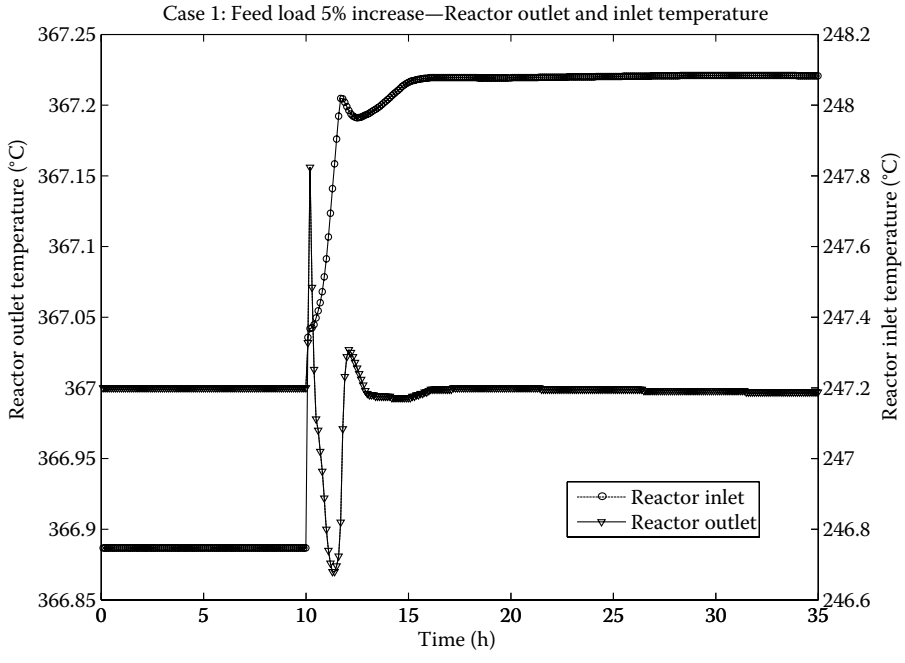

FIGURE 17.6 Closed-loop response of the reactor inlet and exit temperatures in response to a +5% change in feed throughput.

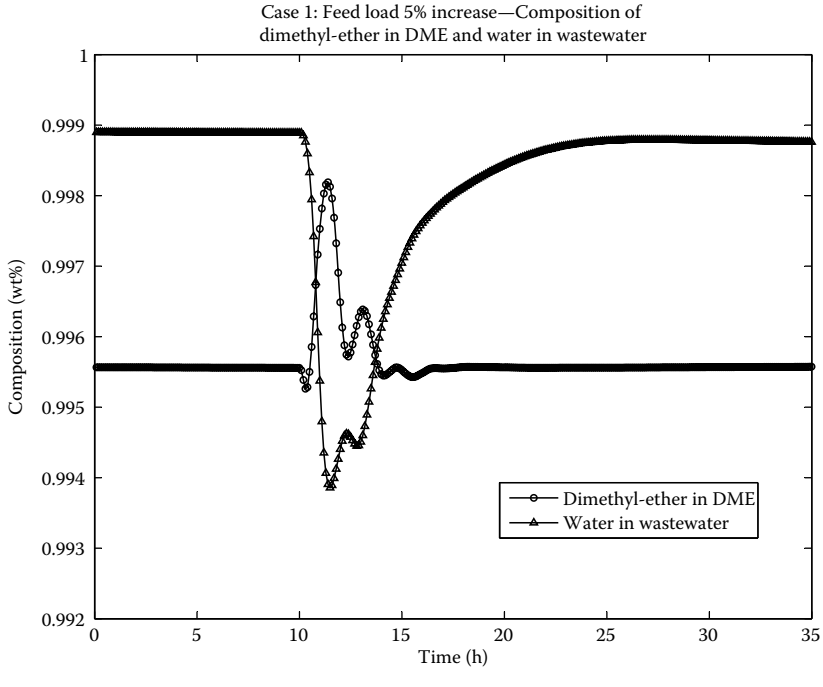

FIGURE 17.7 Closed-loop response of the DME and water product purities in response to a +5% change in feed throughput.

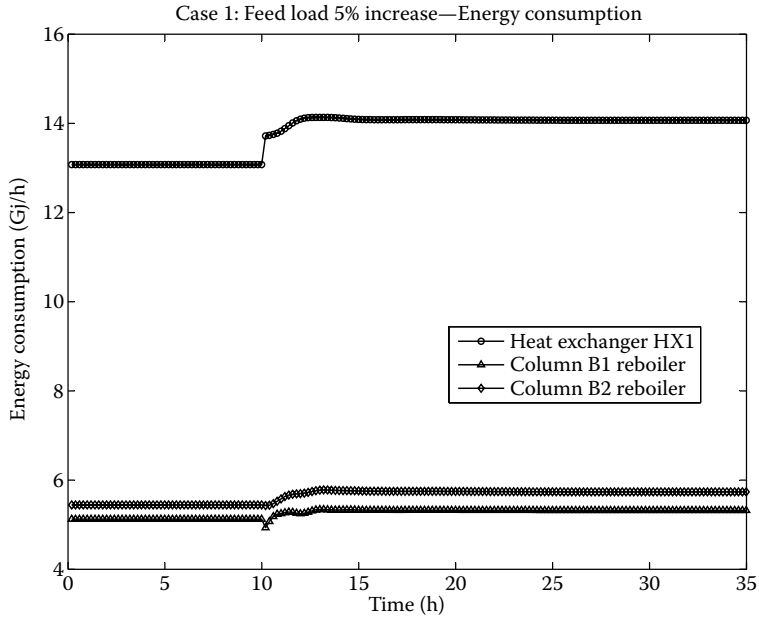

FIGURE 17.8 Utility response to a + 5% change in the throughput.

Key utility responses are shown in Figure 17.8. As the throughput of the process increases, the energy required for the system also increases. It can be seen from Figure 17.8, that the manipulated variables change smoothly and reach a steady value 2 h after the disturbance occurs. As before, it is possible to change the closed-loop performance by modifying the tuning parameters.

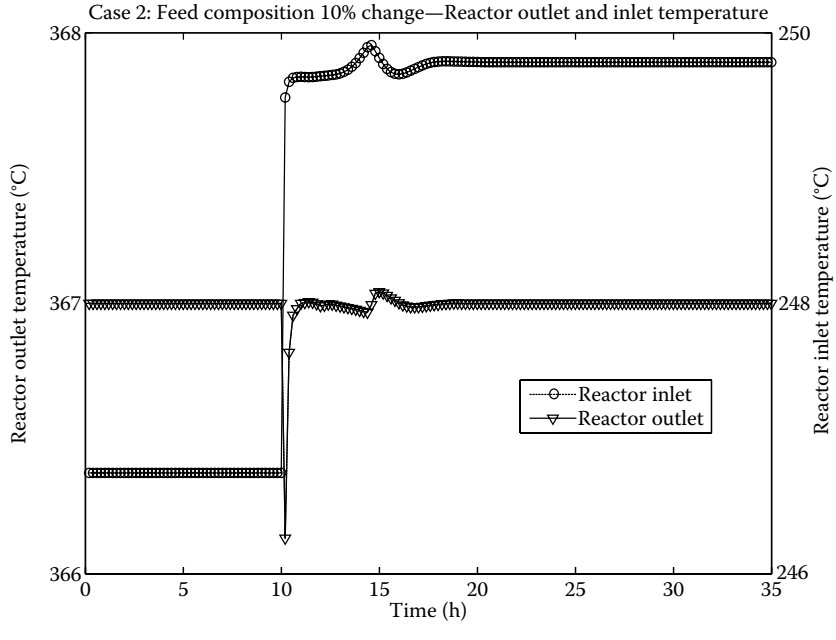

FIGURE 17.9 Temperature profiles of the reactor exit streams in response to a − 10% change in the feed methanol composition.

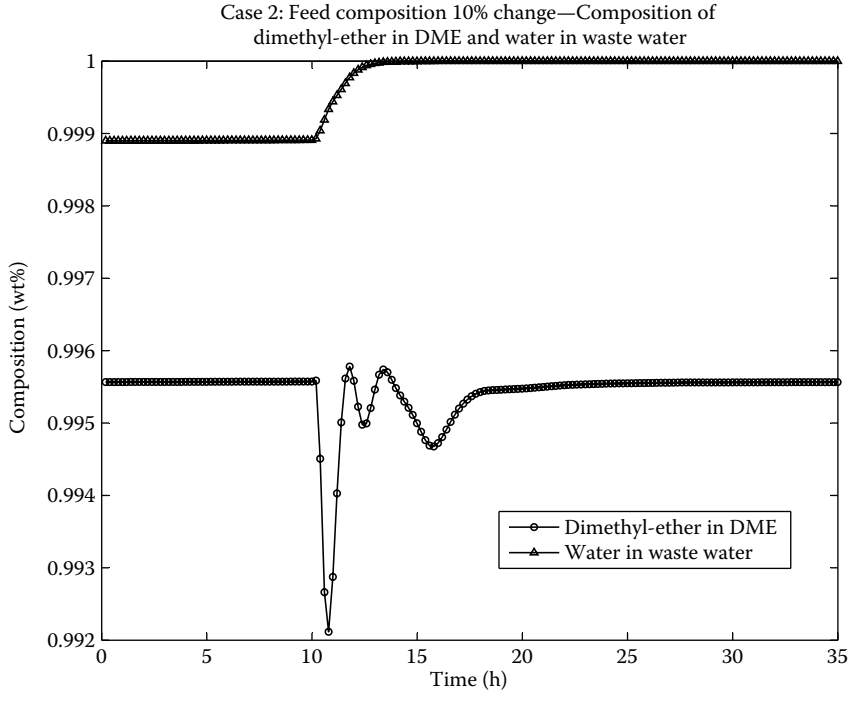

FIGURE 17.10 Closed-loop response of the DME and water product purities in response to a − 10% change in the feed methanol composition.

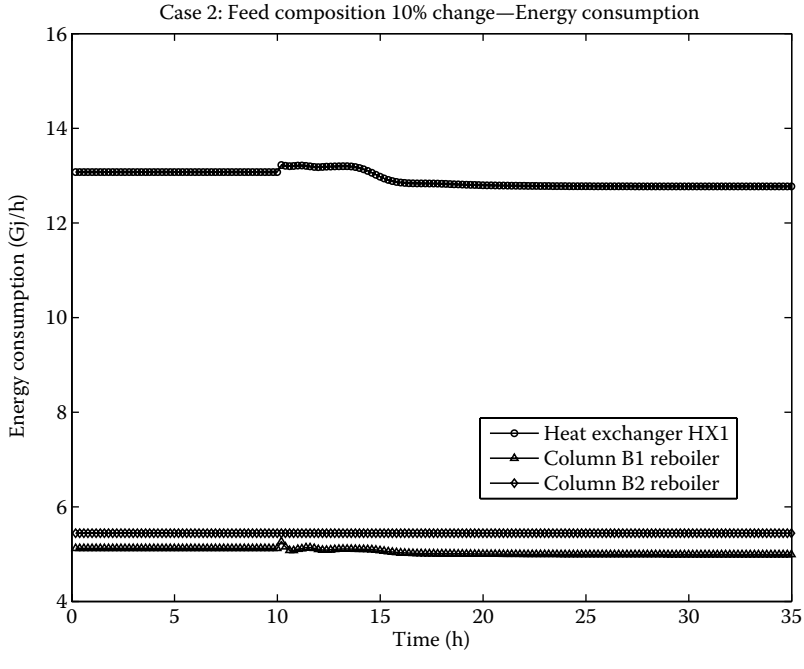

FIGURE 17.11 Utility profiles to regulate a − 10% change in the feed methanol composition.

17.3.3.2 Case Study 2: Feed Composition Change

The closed-loop performance of the reactor inlet and exit temperatures is shown in Figure 17.9. The outlet temperature has a 0.2% undershoot before settling 7 h after the disturbance is introduced. The inlet temperature's set point is 1.2% greater than its initial value at steady state.

Figure 17.10 shows the compositions of DME and water product streams. Near 100% water purity is established after 2.5 h. In the case of the DME product, there is a 0.35% undershoot and a settling time of 7 h.

The duties of the heater and the DME column reboiler are shown in Figure 17.11.

These two case studies show that as long as the reactor exit composition is regulated, the recycle stream will be mostly composed on unreacted methanol. The unusual control difficulties can be interpreted as requiring a more operable design that enables control objectives. More advanced control strategies such as model predictive controller may improve the overall performance of the plant. However, the development of such advanced control strategies, its cost, maintenance, and potential benefits should be investigated thoroughly before the investment of time and resources is expended.

17.4 Summary

In this chapter, a review of the some of the popular plantwide design and control methods is presented. A more in-depth review is limited by space considerations. To assist the reader the methods discussed are classified into two groups, engineering heuristics or experiential knowledge and a mathematical framework. In each category an attempt was made to point out advantages and disadvantages. The simple DME process was used as an example to illustrate some of the more practical approach to plantwide control design principles and methods. The proposed plantwide control structure is shown to be very effective even for two common disturbances, feed throughput (measured) and feed composition (unmeasured) disturbances. From the analysis of the DME plantwide control design, it may be concluded that making use of both experiential knowledge and mathematical constructs is a good compromise to choosing one over the other. It may be akin to the old adage of *always apply common sense*.

It can be appreciated that because of the natural complexity of the plantwide design and control problems, it is not trivial to find a universal solution that can solve all chemical processes. While the heuristics are always valuable and provide insightful guidelines, mathematical methods that support these guidelines are always good to have a strong, well-accepted foundation.

Nomenclature

Notation	Definition
CV	Control variable
DME	Dimethyl ether and/or the product stream of dimethyl ether
E_a	Activation energy of the DME reaction
F_B^{B1}	Flowrate of the bottom stream of column B1
F_{By}	Flowrate of the bypass stream of the heat exchanger HX1
F_{DME}	Flowrate of product stream DME
F_{RCY}	Flowrate of the recycle stream RCY
F_{WW}	Flowrate of wastewater stream WW
H_2O	Water
k	Kinetic parameter of the DME reaction
K_p	Proportional gain
L	The dynamic loss function

Notation	Definition
L_{Con1}	Liquid level of condenser in column B1
L_{Reb1}	Liquid level of reboiler in column B1
IMC	Internal model control
mAHP	Modified Analytic Hierarchical Process
MeOH	Methanol
MILP	Mixed integer linear programming
MINLP	Mixed integer nonlinear programming
MPC	Model predictive control/controller
MV	Manipulate variable
NI	Niederlinski Index
N_f	Feed tray
N_T	Total number of trays
PFR	Plug flow reactor
PID	Proportional integral derivative
P_{MeOH}	Partial pressure of MeOH in the DME reaction
P_{Rin}	Pressure of stream Rin
Q_{Con1}	Duty of condenser in column B1
Q_{Con2}	Duty of condenser in column B2
Q_{HT1}	Heating duty of heater HT1
Q_{Reb1}	Duty of reboiler in column B1
Q_{Reb2}	Duty of reboiler in column B2
R	Universal gas constant
RCY	Recycle stream of the DME process
RGA	Relative gain array
Rin	Inlet stream of reactor PFR
Rout	Outlet stream of reactor PFR
Rr	Molar reflux ratio
r_{MeOH}	Conversion rate of MeOH
Sep	Material stream feeds column B1 for separation
SOCV	Self-optimizing control variable
SP	Set point
T	Temperature
T_{Rin}	Temperature of stream Rin
t_0	Initial time
T_f	Final time
T_I	Integral time constant
WW	Wastewater stream
x_{DME}	Dimethyl ether composition in the product stream DME
x^*_{DME}	Setpoint of the dimethyl ether composition in the product stream DME
x_{WW}	Water composition in the wastewater stream WW
x^*_{WW}	Setpoint of the water composition in the wastewater stream WW

References

1. J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 65:433–444, 1943.
2. M. L. Luyben and W. L. Luyben. Design and control of a complex process involving two reaction steps, three distillation columns, and two recycle streams. *Ind. Eng. Chem. Res.*, 34(11):3885–3898, 1995.

3. M. L. Luyben, B. D. Tyreus, and W. L. Luyben. Plantwide control design procedure. *AIChE J*, 43(12):3161–3174, 1997.
4. M. L. Luyben and B. D. Tyreus. An industrial design/control study for the vinyl acetate monomer process. *Comp. Chem. Eng.*, 22(7-8):867–877, 1998.
5. W. L. Luyben. Design and control degrees of freedom. *Ind. Eng. Chem. Res.*, 35(7):2204–2214, 1996.
6. W. L. Luyben, B. Tyreus, and M. L. Luyben. *Plantwide Process Control*. McGraw-Hill, New York, NY, 1998.
7. W. L. Luyben. *Distillation Design and Control Using Aspen Simulation*. Wiley-Interscience, Hoboken, NJ, 2006.
8. M. L. Luyben and C. Floudas. Analyzing the interaction of design and control-I & II. *Comp. Chem. Eng.*, 18:933–993, 1994.
9. W. L. Luyben. Design and control of recycle processes in ternary systems with consecutive reactions. In *IFAC Workshop, Interactions between Process Design and Process Control*, pp. 65–74. Pergamon Press, Oxford, UK, 1992.
10. W. L. Luyben. Snowball effect in reactor/separator processes with recycle. *Ind. Eng. Chem. Res.*, 33(2):299–305, 1994.
11. R. M. Price and C. Georgakis. Plantwide regulatory control design procedure using a tiered framework. *Ind. Eng. Chem. Res.*, 32:2693–2705, 1993.
12. R. M. Price, P. R. Lyman, and C. Georgakis. Throughput manipulation in plantwide control structures. *Ind. Eng. Chem. Res.*, 33(5):1197–1207, 1994.
13. R. Shinnar. Chemical reactor modelling for purposes of controller design. *Chem. Eng. Commun.*, 9:73–99, 1981.
14. R. Shinnar, B. Dainson, and I. Rinard. Partial control, a systematic approach to the concurrent design and scale-up of complex processes: The role of control system design in compensating for significant model uncertainties. *Ind. Eng. Chem. Res.*, 39:103–121, 2000.
15. P. Buckley. *Techniques of Process Control*. John Wiley & Sons, New York, NY, 1964.
16. J. Douglas. *Conceptual Design of Chemical Process*. McGraw-Hill, St. Louis, MO, 1988.
17. W. Fisher, M. Doherty, and J. Douglas. Steady-state control as a prelude to dynamic control. *Ind. Eng. Chem. Res.*, 63:353–357, 1985.
18. W. Fisher, M. Doherty, and J. Douglas. The interface between design and control. 1. Process controllability 2. process operability. 3. selecting a set of controlled variables. *Ind. Eng. Chem. Res.*, 27:597–615, 1988.
19. W. Fisher, M. Doherty, and J. Douglas. The interface between design and control. 1. Process controllability. *Ind. Eng. Chem. Res.*, 27:597–605, 1988.
20. A. Zheng, R. V. Mahajanam, and J. M. Douglas. Hierarchical procedure for plantwide control system synthesis. *AIChE J.*, 45(6):1255–1265, 1999.
21. J. W. Ponton and D. Laing. A hierarchical approach to the design of process control systems. *Trans. IChemE*, 71:181–188, 1993.
22. M. Morari, Y. Arkun, and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes; Part i: Formulation of the problem. *AIChE J.*, 26(2):220–232, 1980.
23. M. Morari and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes; Part ii: Structural aspects and the synthesis of alternative feasible control schemes. *AIChE J.*, 26(2): 232–246, 1980.
24. M. Morari, Integrated plant control, A solution at hand or a research topic for the next decade?, *Chemical Process Control–II, Proc. of the Eng. Found. Conf.* (T.F. Edgar and D.E. Seborg, eds.), United Engineering Trustees, New York, pp. 467–496, 1982.
25. J. H. Lee and M. Morari. Robust measurements selection. *Automatica*, 27(3):519–527, 1991.
26. J. H. Lee, R. D. Braatz, M. Morari, and A. Packard. Screening tools for robust control structure selection. *Automatica*, 31(2):229–235, 1995.
27. J. H. Lee, P. Kesavan, and M. Morari. Control structure selection and robust control system design for a high-purity distillation column. *IEEE Trans. Control Systems Technol.*, pp. 402–416, 1991.
28. M. Morari and J. Lee. Model predictive control: Past present and future. *Comp. Chem. Eng.*, 24(4):667–682, 1999.
29. Y. Arkun and G. Stephanopoulos. Studies in the synthesis of control structures for chemical processes; Part iv: Design of steady-state optimizing control structures for chemical process units. *AIChE J*, 26(6):975–991, 1980.
30. T. Meadowcroft and G. Stephanopoulos. The modular multivariable constructures for chemical process units. i: Steady-state properties. *AIChE J.*, 38(8):1254–1278, 1992.

31. C. S. T. Ng. *A Systematic Approach to the Design of Plant-Wide Control Strategies for Chemical Processes*. Doctor of philosophy, Massachusetts Institute of Technology, MA, USA, 1997.
32. G. Stephanopoulos and C. Ng. Perspectives on the synthesis of plant-wide control structures. *J. Process Control*, 10:97–111, 2000.
33. S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. John Wiley & Sons, New York, NY, 1996.
34. E. H. Bristol. On a new measure of interactions for multivariable process control. *IEEE Transactions on Automatic Control*, 11:133–134, 1966.
35. G. Stanley, M. Marino-Galarraga, and T. J. McAvoy. Shortcut operability analysis. 1. The relative disturbance gain. *Ind. Eng. Chem. Process Des. Dev.*, 24:1181–1188, 1985.
36. H. P. Huang, M. Ohshima, and I. Hashimoto. Dynamic interaction and multiloop control system design. *J. Process Control*, 4:15–27, 1994.
37. M. Hovd and S. Skogestad. Simple frequency-dependent tools for control system analysis, structure selection and design. *Automatica*, 28(5):989–996, 1992.
38. M. Morari and E. Zafiriou. *Robust Process Control*. Prentice-Hall, Lebanon, IN, 1989.
39. A. Groenendijk, A. Dimian, and P. Iedema. Systems approach for evaluating dynamics and plantwide control of complex plants. *AIChE J.*, 46(1):133–145, 2000.
40. S. M. A. M. Bouwens and P. Kosters. Simultaneous process and system control design: An actual industrial case. In *IFAC Workshop on Interactions between Process Design and Process Control*. Pergamon Press, Oxford, UK, 1992.
41. M. Morari. Design of resilient processing plants—iii: A general framework for the assessment of dynamic resilience. *Chem. Eng. Sci.*, 38(11):1881–1891, 1983.
42. D. R. Vinson and C. Georgakis. A new measure of process output controllability. *J. Process Control*, 10(2–3):185–194, 2000.
43. P. Grosdidier, M. Morari, and B. R. Holt. Closed-loop properties from steady-state gain information. *Eng. Chem. Fundam.*, 24(1):221–235, 1985.
44. L. Narraway and J. D. Perkins. Selection of process control structure based on linear dynamic economics. *Ind. Eng. Chem. Res.*, 32(11):2681–2692, 1993.
45. J. B. Lear, G. W. Barton, and J. D. Perkins. Interaction between process design and process control: The impact of disturbances and uncertainty on estimates of achievable economic performance. *J. Process Control*, 5(1):49–62, 1995.
46. M. J. Mohideen, J. D. Perkins, and E. N. Pistikopoulos. Robust stability considerations in optimal design of dynamic systems under uncertainty. *J. Process Control*, 7(5):371–385, 1996.
47. C. Loeblein and J. D. Perkins. Structural design for on-line process optimization (1) dynamic economics of mpc. *AIChE J.*, 45(4):1018–1029, 1999.
48. C. Georgakis, D. R. Vinson, S. Subramanian, and D. Uzturk. A geometric approach for process operability analysis. *Comp. Aided Chem. Eng.*, 17:96–125, 2004.
49. C. L. E. Swartz. A computational framework for dynamic operability assessment. *Comp. Chem. Eng.*, 20(4):365–371, 1996.
50. P. A. Bahri, J. A. Bandoni, and J. A. Romagnoli. Effect of disturbances in optimizing control: Steady-state open loop back-off problem. *AIChE J.*, 42:983–994, 1996.
51. T. Larsson and S. Skogestad. Plantwide control: A review and a new design procedure. *Int. J. Control*, 21(4):209–240, 2000.
52. S. Skogestad and M. Morari. Effect of disturbance directions on closed-loop performance. *Ind. Eng. Chem. Res.*, 26:2029–2035, 1987.
53. S. Skogestad. Plantwide control: the search for the self-optimizing control structure. *J. Process Control*, 10:487–507, 2000.
54. S. Skogestad. Self-optimizing control: The missing link between steady-state optimization and control. *Comp. Chem. Eng.*, 24:569–575, 2000.
55. S. Skogestad. Control structure design for complete chemical plants. *Comp. Chem. Eng.*, 28:219–234, 2004.
56. S. Skogestad. Near-optimal operation by self-optimizing control: From process control to marathon running and business systems. *Comp. Chem. Eng.*, 29:127–137, 2004.
57. S. Skogestad. The do's and don'ts of distillation column control. *Trans IChemE Part A*, 85:13–23, 2007.
58. E. Vasbinder and K. Hoo. The use of decision-based approach to the evaluation of plant-wide control problem. *Ind. Eng. Chem. Res.*, 42:4586–4598, 2003.
59. R. Turton, R. Bailie, W. Whiting, and J. Shaeiwitz. *Analysis, Synthesis, and Design of Chemical Processes*. Prentice-Hall International, Upper Saddle River, NJ, 1998.

60. M. J. Piovoso and K. A. (Hoo) Kosanovich. Applications of multivariable statistical methods to process monitoring and controller design. *Int J Control*, 59(3):743–765, 1994.
61. K. J. Astrom and H. Hagglund. *PID Controllers: Theory, Design, and Tuning* (2nd ed.) Instrument Society of America, Research Triangle Park, NC, 1995.
62. B.A. Ogunnaike and W.H. Ray. *Process Dynamics, Modeling and Control.* Oxford University Press, New York, NY, 1994.
63. D. Seborg, T. F. Edgar, and D. A. Mellichamp. *Process Dynamics and Control* (2nd ed.). John Wiley, New York, NY, 2004.

18

Automation and Control Solutions for Flat Strip Metal Processing

Francesco Alessandro Cuzzola
Danieli Automation

Thomas Parisini
University of Trieste

18.1 Introduction

The final products realized in flat strip processing of metal are tinplates and galvanized sheets whose commercial availability is now worldwide since it is exploited in many applications, including automotive, food industry, building, and defense industries.

The performances required in this field concern not only the precision of the product to be realized, but also the need of guaranteeing limited consumption of electric energy and raw materials. Due to the constant increment of the performance required, the research efforts toward the realization of new mechanical, automation and control technologies are constantly synergistically increasing.

In this article, we will mainly focus on the realization of control and automation solutions for flat steel strip although most of the concepts can be applied also for the realization of copper and aluminum strips.

The automation technology applied in flat strip processing is traditionally divided in to three parts referred to as Level 1, 2, and 3, respectively. In all these three automation layers that need to cooperate hierarchically in order to guarantee the best final product performances and the highest productivity levels, a number of control technologies, mathematical models of physical phenomena and optimization algorithms are implemented.

In this contribution, whose first purpose is to present the most significant automation and control problems that can be found in all the main processes devoted to flat strip processing, we want to focus on the *Galvanizing* process since it represents a very complex system, still not well described in the literature, that has all the automation peculiarities typical for this sector. In particular, after describing the rolling process (both hot and cold cases) and the pickling process, we want to present the *coating control* problem to be managed in the galvanizing process as an example where Level 1 and Level 2 automation need to cooperate for guaranteeing the effectiveness of a closed-loop regulation system. In this review, the sensor and actuator technologies available today are discussed as well.

The chapter is organized as follows. In Section 18.2, we give a presentation of the metals flat production cycle and the structure of a typical automation system for the processes involved. In the subsequent sections, we define the main aspects concerning the control technologies for *hot rolling* (Section 18.3), *pickling* (Section 18.4) and *cold rolling* (Section 18.5). Finally, in Section 18.6, we give a thorough presentation of the *galvanizing* process and the advanced control technologies that have been introduced in the very latest years for the coating control problem.

18.2 Flat Metal Processing

18.2.1 Main Phases of Flat Strip Processing

Flat metal forming is realized through several consecutive processes whose complexity involves mechanical and automation technologies that are continuously subject to research [1–3].

In steel-making plants, liquid metal is first formed in a blast furnace by reducing the iron oxide. After further processing, the liquid metal is cast by a process called continuous casting into raw stock shapes. These are very large pieces of metal with typical rectangular cross-sections (*slabs*). These pieces have too large dimensions for practical applications and for this reason their thickness is reduced in rolling mills that compress them till reaching a desired thickness.

Rolling mills for flat products can be placed in two categories: *hot rolling mills* and *cold rolling mills*. In hot rolling, the material is heated to just below its melting point before being fed into the rolling process. In cold rolling the material has the environmental temperature and for this reason its hardness is much higher than in the hot rolling case.

Hot rolling and cold rolling are based not only on different mechanical solutions but also on significantly different control technologies. In turn, the typical performances required in terms of thickness targets and tolerances for the final product in hot rolling and cold rolling are significantly different.

For instance, concerning the steel case, the minimal thickness reached in modern *hot strip mills* (HSM) is about 1 mm whereas in *cold rolling mills* (CRM) the final target thickness is about 0.15 mm.

After hot rolling, in order to prepare the material for cold rolling the surface of the metal strip needs to be deprived of the oxide layer that is generated by environmental agents. This is realized through a *pickling line* (PKL).

After cold rolling the material can be treated again in *hot dip galvanizing lines* (HDGL) in order to recover both the mechanical properties of the material (since it has been a subject of a hardening effect during cold rolling due to the compression of the metal crystals) and also to improve the surface aspect and protect it from further oxidation.

All these processes involved in flat metal processing can be controlled through a standard software and automation architecture that includes three automation layers.

18.2.2 Realization of an Automation System in Flat Strip Metals Processing

The hierarchical structure of a control automation system usually adopted for flat strip processing and several other technological processes in the metals industry is depicted in Figure 18.1 [4,50].

The Level 1 automation directly interacts with low-level devices (actuators and transducers). Real-time control loops and logic sequences are implemented here. Fast sampling (1 ms) and high computing power are achieved, for instance, through VME (Versa Module European) architecture technology. Conventional PLC, instead, guarantees a minimal sample time of 10 ms. The human–machine interface (HMI) offers to the operator a real-time look at the process.

The Level 2 automation provides higher-level control functions and utilities, like optimal plant setup calculation, generation of production reports and statistical analysis of product quality. In particular

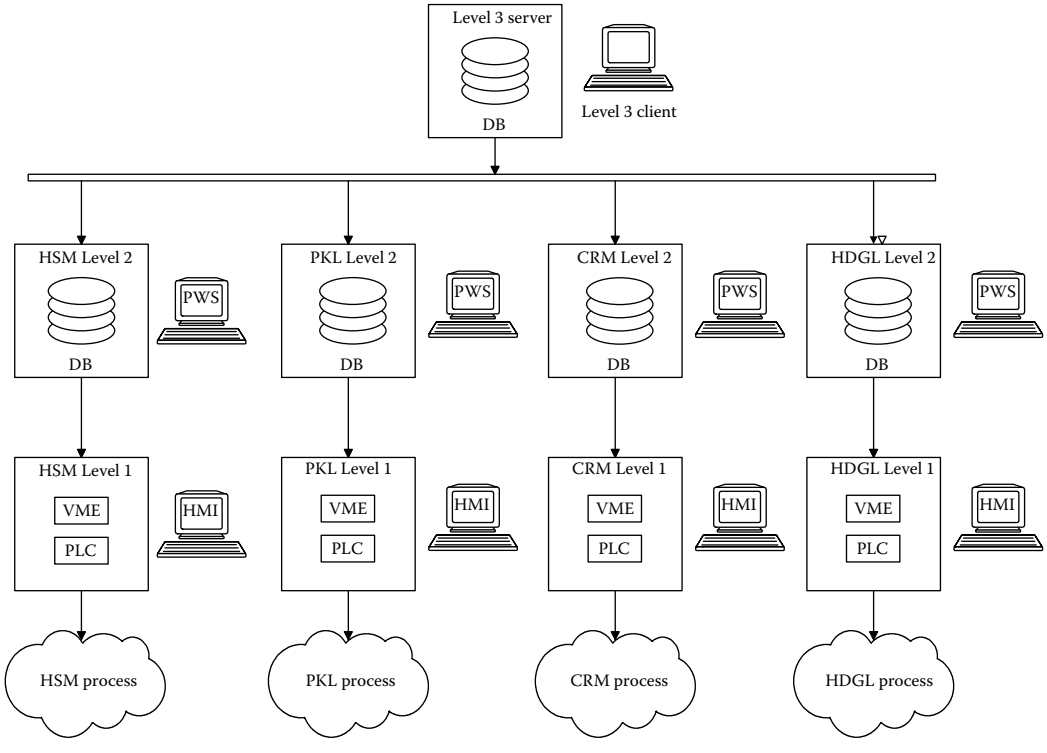

FIGURE 18.1 Structure of a typical automation system for flat metal processing.

mathematical models of technological processes are used to generate proper plant setups. Reliability of physical models, at different and even time-varying working conditions, is guaranteed by *self-adaptation* that is, identification techniques based on plant feedback that improve recursively the reliability of the model predictions. Technological information and historical archive of production are stored into the Database (DB), while the process workstation (PWS) offers a graphic interface to the Level 2 utilities.

In many cases, a Level 3 automation system is implemented in order to provide additional utilities for top-level production supervision (the so-called Manufacturing Execution System (MES) functions), storage yard management and coordination among Levels 2 of the different processes belonging to the same plant.

As put in evidence in Figure 18.1, the Level 3 automation system is in charge of coordinating the production scheduling between the hot plant area (represented by the HSM) and the cold plant area, that is, the CRM, the PKL and the HDGL.

18.3 Flat Strip Processing in HSM

Referring to the steel case, the HSM purpose is to process cast steel slabs having 250 mm thickness into steel flat strip with as little as 1.0 mm thickness.

The typical HSM process consists of the following steps (Figure 18.2):

- The casting machine where a continuous steel casting is produced and then cut into slabs.
- The slabs are then reheated in a furnace in order to reach the optimal temperature.
- A first rolling stand (the roughing mill) realizes a preliminary thickness reduction.
- Then, the finishing mill constituting 5/7 consecutive rolling stands has the purpose of reducing the thickness to the desired value.

In the finishing mill an important task is performed by a hydraulic arm, called the *looper*, placed in the middle between two consecutive stands and whose purpose is to keep the strip tension at a constant value. This mechanical system is subject to particularly unstable dynamics that makes the control problem tricky.

18.3.1 Control Technologies Applied to HSM

The use of advanced control and modeling solution for HSM has been subject to wide research efforts in many directions in the past 40 years (see [5] and references quoted therein):

- The use of multivariable control techniques has been proposed for the finishing mill since the 1970s (see [6] and references therein) and now it is considered a consolidated tool for controlling the generic rolling stand together with the downstream looper or the downstream coiler.
- Several models have been developed in order to predict the material characteristics according to the material temperature and the rolling process [7] and control is applied for regulating the coiling temperature.
- Advanced control techniques are applied in order to compensate friction phenomena [8].

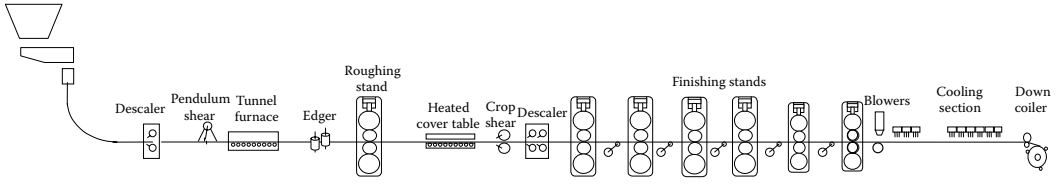

FIGURE 18.2 A conventional HSM process.

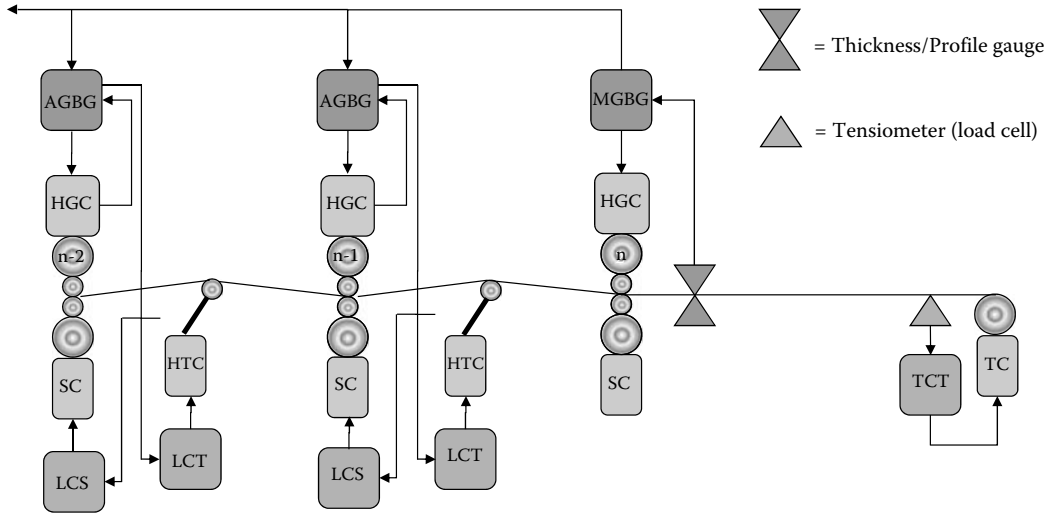

FIGURE 18.3 Thickness control in HSM.

- Models and controllers [9] are proposed in order to improve the material flatness and profile.
- Steering control techniques [10,11] are recently introduced in order to increase the productivity levels by reducing the probability of cobble events.

In the following sections we will concentrate on the presentation of the control technologies for thickness regulation [51–53].

In Figure 18.3 we report an example of a thickness regulator applied to the HSM case and we put in evidence that typically the HSM is provided with the following sensors:

- *Thickness and profile gauge* is based on x-ray technology and is aimed at measuring the thickness in the centerline of the material. Rarely they are mounted on a moving carriage and can measure the whole thickness profile along the width of the coil. In general only one thickness/profile measurement system is installed at the end of the mill.
- *Load cells* are provided in order to have a measurement of the *rolling force* that represents a fundamental measurement signal in HSM thickness regulation. In case the load cells are not provided then the measurement of the hydraulic force signal generated by the pressure transducers installed in the main cylinder can be exploited as an alternative measure.
- *Load cells* in some cases are mounted on the loopers in order to get a direct measurement of the *interstand strip tensions*. Also in this case an alternative measure is represented by the force signal generated by the pressure transducers mounted in the hydraulic cylinder acting on the looper.

In the following, we will distinguish between *basic* and *external* controllers, that is, controllers that are in charge of implementing references for physical actuators (basic controllers) and controllers that produce references for basic controllers in order to reach the desired target (external controllers).

The thickness controller is realized by means of the following *Basic* controllers:

- The *hydraulic gap controller* (HGC) receives a gap reference and measures the gap coming from position encoders placed in the hydraulic cylinder and produces the servo-valve command that indeed controls the oil mass flow generating the movement of the cylinder. Of course, the measured gap can be significantly different from the physical gap of the stand because of the stand elastic stretch.

- The *torque controller* (TC) controls the torque generated by the two reels. These controllers receive a torque reference that is produced by the *tension control by torque* (TCT) controller that aims at keeping constant the strip coiling/uncoiling tensions.
- The *speed controller* (SC) is in charge of regulating the stand speed. Of course, in order to guarantee the rolling stability, the speed reference must be coordinated with the other entities in the mill.
- The *hydraulic torque controller* (HTC) is in charge of controlling the torque generated by the looper.

18.3.2 Automatic Gauge Control: The Realization of Thickness Control in Hot Rolling

The acronym AGC (*automatic gauge control*) is the system in charge of regulating the thickness. In HSM applications the AGC strictly needs the acquisition of the *stretch* for each stand (Figure 18.4). As it will be detailed later in Section 18.5, in *cold rolling* the acquisition of the stand stretch is much less important.

The stand stretch represents the elastic behavior of the mechanical structure of the stand when a compressing force is generated by the main hydraulic cylinder (i.e., the HGC cylinder). This characteristic must be known in advance for implementing the AGC in HSM and for this reason a suitable control sequence is implemented and executed offline, that is, before rolling (the *stretch acquisition sequence* (SAS)).

The SAS is realized by putting the work rolls in contact and linearly modifying the position reference for the HGC from a minimum value to a maximum value. For each position reference the force measured by the load cells (or by the HGC hydraulic force measurement) is recorded in order to build a stretch characteristic like the one depicted in Figure 18.4. The records are, in general, performed twice: the first one with increasing HGC position references (up readings) and the second one with decreasing HGC position references (down readings).

The differences between the up readings and the down readings are connected with a nonnegligible hysteresis in the elastic behavior of the stand. Finally, a best-fit polynomial curve of the following form is stored in order to perform the AGC task:

$$Stretch\,(F) = a^1 + a^2 F + a^3 F^{1/2} + a^4 F^{1/3} + \cdots + a^n F^{1/n-1} \tag{18.1}$$

where F is the measured force.

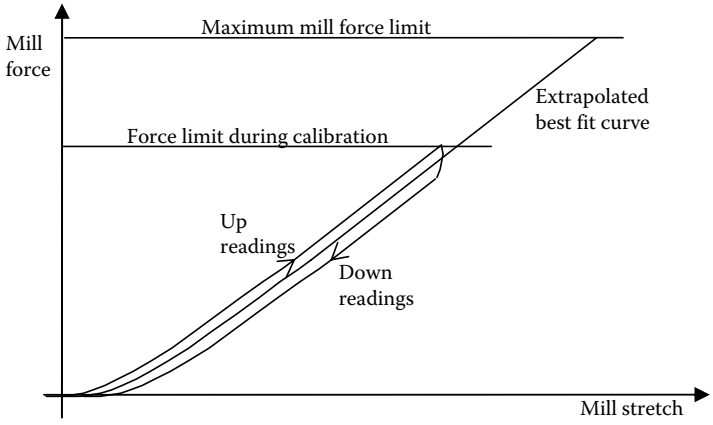

FIGURE 18.4 Mill stretch characteristic.

It needs to be pointed out that the acquisition of the stretch characteristic *stretch* (F) can be exploited during rolling to derive an indirect measurement of the material exit thickness h as follows:

$$\hat{h} = S + Stretch\ (F) \tag{18.2}$$

where

- h is the strip exit thickness for the considered stand and $\hat{h}$ is its estimate derived from the previous equation.
- S is the measured gap for the considered stand derived from the encoders mounted in the hydraulic cylinder.
- F is the measured rolling force (from load cells or from the HGC pressures).

Equation 18.2, referred in the literature to as *Gaugemeter* equation, is often simplified by introducing the so-called *Mill Modulus* M_m of the stand, that is, the elastic constant of the stand:

$$\hat{h} = S + \frac{F}{M_m}. \tag{18.3}$$

In general the real implementation of conventional AGC is based on the Equation 18.2 whereas, the advanced controller synthesis based on models can exploit the linear version represented by Equation 18.3 [12,13].

The AGC in HSM has the purpose of keeping constant the thickness of the material by acting on the position references for all the HGC by compensating several phenomena—for instance the hysteresis of the stand stretch, the variation of the material hardness caused by possible fluctuations of the material temperature, and so on.

To do this it is necessary to take into account that the presence of a looper between one stand and the following one implies that the regulation performed by one stand does not influence the regulation performed by the adjacent stands provided an effective interstand tension control is guaranteed by the looper. This fact represents the main reason why the control architecture of AGC for *hot rolling* and *cold rolling* are significantly different.

The AGC for HSM is realized by some *External* controllers cooperating during rolling. In particular two regulators are in charge of controlling the looper [49]:

- *The looper control by torque* (LCT): The LCT realizes the regulation of the *interstand tension* by acting on the torque reference exploited by the HTC. In general the LCT is fed by the tension error generated by a load cell mounted on the looper or, alternatively, by the estimation of the interstand tension derived by the looper hydraulic force.
- *The looper control by speed* (LCS): The LCS aims at regulating the *looper angular position* by acting on the speed reference of the upstream stand (i.e., by acting on the reference for the SC acting on the upstream stand). This regulator is also referred to as the *Mass Flow* regulator.

The proper thickness regulation is realized in a different way for the intermediate stands and for the final stand respectively. Indeed, *for the intermediate stands* a direct thickness measurement is not available and consequently an indirect measurement of the thickness is achieved from the *Gaugemeter* principle of Equations 18.2 and 18.3.

Consequently, the AGC represented in Figure 18.4 is composed of the following two regulators:

- The *absolute gauge control, feedback via gap* (AGBG): The AGBG is applied to all the intermediate stands that are not provided with a direct thickness measurement device and is based on the *gaugemeter* principle and generates a trim for the gap reference of the corresponding HGC. This controller is also in charge of making some feedforward compensations connected to the variation of the oil film for the backup roll bearings, the thermal expansion of the work roll due to the contact with the strip and the variation of the roll diameters due to wear.

- The *monitor gauge control, feedback via gap* (MGBG): It aims at keeping the strip thickness exiting the last stand of the finishing mill according to the proper target value by using the feedback of thickness coming from the x-ray located at the mill exit. The deviation signal is used to correct the gap references for the HGC of all the stands. Indeed, a dedicated algorithm defines how to distribute the corrections among all the finishing stands.

The main problem in implementing the MGBG is that it is strictly necessary to take into account the transport delays between the x-ray and the stand that implements the required correction.

Finally, as put in evidence in Figure 18.3, the LCT can receive a trim from the AGBG regulator in order to reduce the interactions between the LCT and the AGBG.

18.3.3 Speed Master

The speed of the stands and reels must be coordinated in order to guarantee the stability of the mill; this feedforward controller is known as *speed master*.

In order to prevent instability problems for the hot rolling process, one stand is selected as the *pivot stand* and the speed variations of the pivot are compensated in feedforward through suitable speed variations for the other stands.

In order to do this, it is fundamental to know, as precisely as possible, the *forward slip* (FS) for all the stands, that is, the following coefficient representing the relation between the stand motor angular speed Ω and the exit strip speed V_{out}:

$$FS = \frac{V_{out}}{R\Omega} \tag{18.4}$$

where R is the work roll radius.

In general, the FS coefficients are estimated through suitable mathematical models installed in the Level 2 automation system together with its sensitivities with respect to the tension set points and the strip speed.

18.3.4 The Multivariable Control Applied for the Finishing Mill

In the last few years, advanced control technologies are implemented and are now considered well established in the control of the thickness in the HSM finishing mill. As widely presented in the literature (see, e.g., [12–17]) the main purpose is to provide a multivariable framework in order to integrate the main controllers acting in the HSM process (more precisely, the AGBG, LCT, and LCS) in only one controller that reduces possible interferences between the various tasks and allows not only to increase the performances but also to decrease the probability of cobbles during the realization of ultra-thin gauges.

The multivariable control is consequently applied in the intermediate stands in order to perform together the AGBG realized by the ith stand together with the LCT/LCS applied to the downstream looper (Figure 18.5).

Another reason for using advanced control is represented by the necessity of introducing *a priori* robustness about a possible uncertainty associated to the knowledge of the stand stretch: indeed, it is possible to prove that the presence of a strong uncertainty in the knowledge of the mill modulus could cause the AGBG instability (see [12] and references quoted there in). On the other hand, as previously explained, the measurement of the stretch is performed off-line and it is subject to time-variability together with the stand wear.

18.4 Modeling and Control of Steel Pickling Process

After leaving the HSM, the material reacts both with air and cooling water, thus leading to the formation of an oxide layer. Its properties depend on many factors: the steel's chemical composition, the so-called

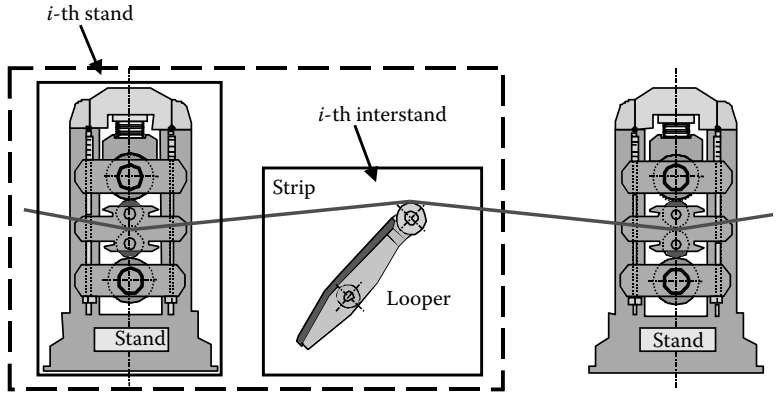

FIGURE 18.5 The contest of application of a multivariable control for HSM.

down-coil temperature (i.e., the strip temperature when leaving the HSM), and the cooling process duration [55,56].

18.4.1 Pickling of Carbon Steel

Descaling of hot rolled strips is achieved in pickling lines (PKLs) usually consisting of many processing tanks, where the steel strip comes into contact with a corrosive solution (in general, hydrochloric acid). Acid reacts with the oxide layer. Iron chlorides and iron ions are dissolved into the acid baths. The processing capability of the line is maintained by reintegrating baths with fresh acid and discharging the exhaust solution. In most cases, the PKL is coupled with an *acid regeneration plant* (ARP), which regenerates the exhaust solution by removing the iron contents, thus saving the consumption of fresh acid.

Usually, the scale thickness is 10–20 μm. Its so-called *crack structure* is not uniform, but it mainly consists of a layer of wustite (FeO). According to the descaling model described in [18], the acid penetrates the scale structure, thus reaching the free metal surface. The reaction occurring between acid and free metal generates a local current flowing from the scale-free metal portion, acting as an anode, to the conductive wustite layer, playing the role of cathode.

$$\mathrm{Fe} \longrightarrow \mathrm{Fe}_2^{+} + 2e - . \tag{18.5}$$

The local current reduces the trivalent ions in the wustite, transforming them into acid-soluble bivalent iron ions. The scale is quickly dissolved. The chemical reaction is generally accelerated by heating the acid solution up to 65–85°C. Different design concepts of strip processing tanks characterize different pickling systems (Figure 18.6). They are briefly presented in the sequel.

Deep-type PKLs consist of deep working tanks where the acid solution presents an extremely slow flow motion.

Shallow-type PKLs rely on a different tank design, which guarantees higher pickling effect on the bottom surface of the strip, with respect to conventional deep-type lines, even with slow motion of acid flow. Unfortunately, some sealing problems characterize this kind of plants.

Turbulence PKLs maintain the acid liquor in movement thanks to directly injected acid. The higher kinetic energy of the directly injected acid accelerates the descaling process and increases the heat transfer coefficient between strip and acid.

Turboflo PKLs consist of pickling tanks divided into a number of cells of 2 m length with special tank covers [19]. In this case higher strip speeds (up to 400 m/min for light gauge strips) can be reached without compromising the pickling effect.

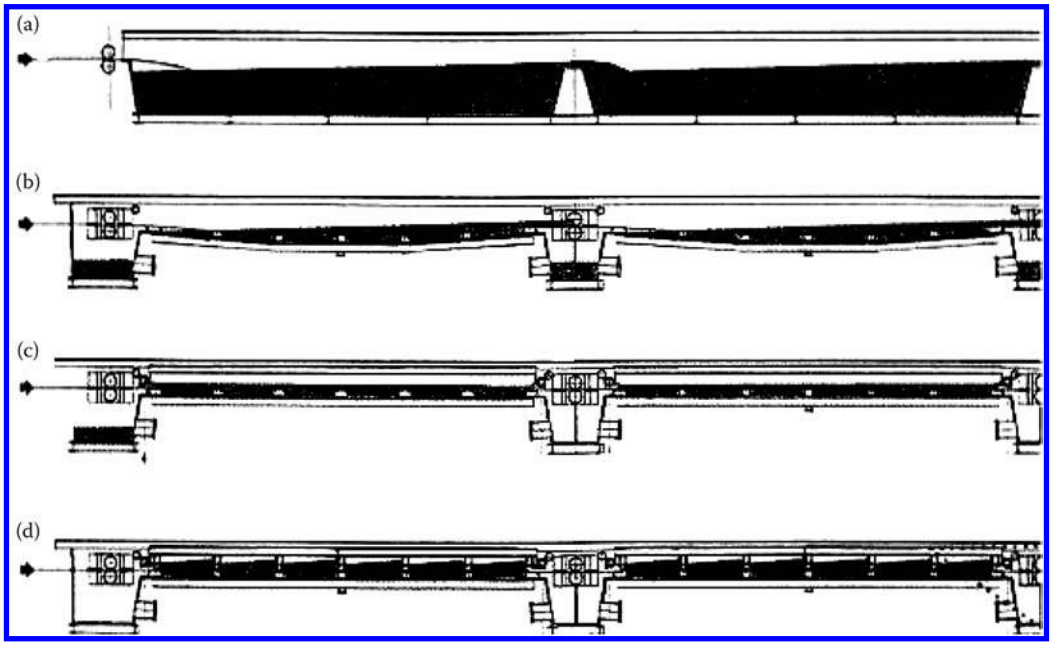

FIGURE 18.6 Different design of PKL tanks. (a) Deep-type tanks; (b) shallow-type tanks; (c) turbulence tanks; (d) turboflo tanks.

18.4.2 Pickling of Stainless Steel

Stainless steel is produced in many different qualities depending on the different material application and comprises of austenitic, ferritic, and martensitic grades. It is available in the form of cold-rolled or hot-rolled strips.

Recrystallizing treatment in annealing furnaces is required for both hot- and cold-rolled strips, before being used or cold rolled again. Thus, repeated descaling of stainless-steel strips needs to be performed, in order to remove mill scale and scale resulting from annealing process as well.

With respect to carbon steel, the scale from stainless steel is more difficult to be removed. Moreover, hot rolling and annealing processes lead to the diffusion of chromium from the upper layer of base material into the scale. Also the resulting chromium depleted layer on the strip surface needs to be removed by the pickling process.

Annealing and descaling treatment of stainless steel are often combined in a single processing line [20,54]. Since the strip speed is governed by the annealing section, the pickling process must guarantee a suitable operational flexibility to avoid unwanted *underpickling* (i.e., incomplete scale removal) or *overpickling* phenomena. This requirement led to the development of specific descaling processes for stainless steel.

18.4.3 Management and Control of Pickling Processes

In the context of steel manufacturing, many efforts are devoted to the development of advanced control methodologies for rolling mills. Processing lines (pickling and annealing) instead, are characterized by slow dynamics. For these reasons, usually they are not supplied with sophisticated automation systems. PKLs in particular, are often managed through semimanual operating practices, defined on the basis of operator experience and plant specific knowledge.

Nevertheless, in recent years the field has become quite competitive and modern control and automation solutions are required in order to achieve significant improvements in quality, production, and

reduction of consumption. The increasing interest of model-based applications [21,22] is motivated by providing additional process monitoring capabilities and offering valid support to the operators' decisions. Some of the main motivations for improving automation systems of PKLs are recalled in the sequel.

1. Modern Turboflo PKLs [19] are characterized by fast dynamics, which require higher control performance with respect to traditional push-pull or even low-speed continuous lines [22,23].
2. Real-time chemical analysis of pickling baths is usually unavailable. Process monitoring capabilities of plant automation systems can be extremely useful for plant operators [24].
3. ARPs usually guarantee the best performances and efficiency if the process control systems are able to keep the concentrations of metals in the exhaust acid solution, coming from the PKL, almost constant.
4. The increasing demand for high quality steel products requires that the correct grade of descaling is ensured by process automation. *Underpickling* is responsible for corrosion damages of steel strips. *Overpickling*, instead, reduces the strip quality and increases its roughness, thus leading to different friction values during cold rolling.

Through the support of valid process control systems it is possible to improve the plant's efficiency by reducing the consumption of steam and fresh acid.

18.4.4 Pickling Lines Main Components

As previously mentioned, PKLs present a modular structure. The descaling process takes place in many equally sized consecutive stages. The typical structure of a pickling stage is shown in Figure 18.7. The list of its main components follows.

1. In the *working tank* (WT) the strip comes into contact with the acid reactant (different designs of WT are shown in Figure 18.6). The WT is continuously refilled with heated acid solution, while the exhaust solution is drained, and the fluid level is kept constant. Part of the pickling medium

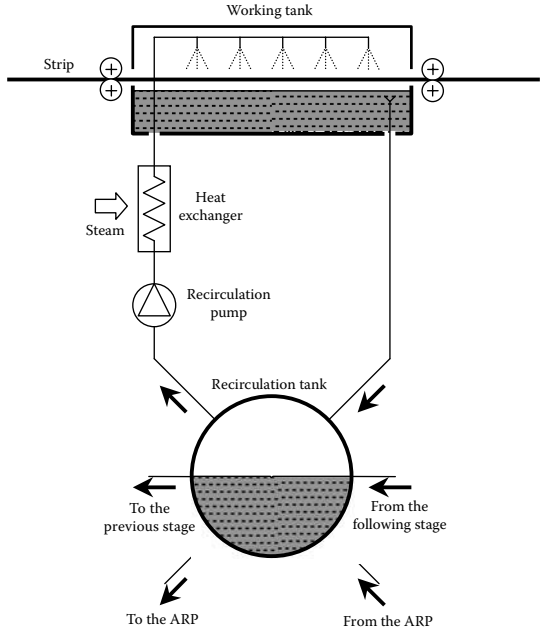

FIGURE 18.7 Schematic graph of a pickling stage. PKL models.

can be lost because of evaporation, due to temperature and spraying. Part of the solution, instead, is transported, despite sealing, on the strip surface into the following treatment tank.

2. The *recirculation tank* (RT) provides acid solution to the WT, and receives, from this one, the pickling liquor enriched with metals. Depending on both the line structure and the sequential position of the stage, the RT can receive fresh acid solution directly from the ARP and/or from the adjacent stages. In the same way, it can be directly drained, otherwise part of the fluid can be sent to the previous or to the next stage.
3. The *recirculation pump* (RP) guarantees the continuous flow of pickling solution between working and recirculation tank. Conventional PKLs are provided with fixed-speed recirculation pumps. In this case, the recirculation flow rate must be high enough to keep the concentration of acid and metals in the working and in the RT at the same values. Modern Turboflo, instead, usually rely on variable-speed recirculation pumps. The possibility of changing the pump speed introduces a fast control action on the acid concentrations inside the WT, especially useful in case of fast transients or plant stops.
4. Through the heat exchanger, thermal power provided by hot steam is transferred to the pickling liquor. The temperature of the acid solution downstream the exchanger is regulated by means of a closed-loop controller of steam flow.

18.4.5 Pickling Line Models

Nomenclature:

T	Acid solution temperature
x	Acid concentration
L_{WT}	Length of the working tank
w_s	Strip width
q_S	Mass flow of scale removed from the strip
δ_s	Mass density of the oxide layer
h_{sin}	Inlet scale thickness
h_{sout}	Outlet scale thickness
v_s	Strip speed
I	Specific current per surface unit

18.4.5.1 Purpose of the Mathematical Model

For PKLs, Level 1 provides real-time control of strip speed and tensions, acid solution level inside treatment tanks, temperature of pickling baths and electric current for electrolytic processes. Moreover, Level 1 manages all the automatic sequences for tanks refilling.

Level 1 receives the proper set-point values of the above mentioned variables from Level 2, depending on the features of the coil to be processed and on the status of the line in terms of acid solution concentrations and temperatures. In this case, the support of a mathematical model aims at guaranteeing:

- Optimal setup calculation.
- Coil-to-coil estimation of bath's degradation even if real-time chemical analysis of pickling baths is unavailable.
- Optimal tank refilling with consequent saving of fresh or regenerated acid.

18.4.5.2 Structure of the Mathematical Model

The mathematical model is based on first principles and relies on mass balance equations expressed for the RT and WT. These mass balanced equations can be easily derived, not only for global volumes but also for each chemical species including the mass flows lost due to evaporation and the chemical mass flows deriving from chemical reactions.

In particular, for PKLs using hydrochloric acid, the most important chemical reaction is represented by:

$$FeO + 2HCl \rightarrow Cl_2Fe + H_2O.$$

As proposed in [22], a proper mathematical description of the descaling process can be achieved by introducing a quantity called *average reaction speed* (ARS). It is defined as the rate of removed oxide quantity per surface and time unit. ARS can be determined by means of laboratory tests and it is also widely used to express the effectiveness of a line operated at specified concentrations and temperatures.

ARS is a quadratic function of acid concentration x and temperature T:

$$ars\,(x, T) = k_{x1}x + k_{x2}x^2 + k_{T1}T + k_{T1}T^2. \tag{18.6}$$

The flow rate of removed oxide taking place in a treatment tank of length L_{WT}, processing a strip of width w_s, can be obtained as

$$q_s = ars\,(x, T)w_sL_{WT}$$

Then, the scale thickness at the output of the treatment tank can be computed as follows:

$$h_{sout} = h_{\sin} - \frac{q_s}{w_s\delta_s}\frac{1}{v_s}.$$

In electrolytic PKLs for stainless steel the ARS formula (Equation 18.6) needs to be replaced because the descaling process is driven by electric current. For lines using a neutral electrolyte, like Na_2SO_4, the quantity of removed oxide layer is essentially proportional to the quantity of electric charge moved through the strip surface. If an acid reactant, like H_2SO_4, is used, even its descaling action has to be properly taken into account. Temperature, instead, does not significantly affect the pickling process.

A model of ARS for electrochemical pickling can be obtained by simply replacing temperature with specific current I in Equation 18.6:

$$ars\,(x, I) = k_{x1}x + k_{x2}x^2 + k_{T1}I + k_{T1}I^2. \tag{18.7}$$

18.5 Cold Rolling: Control Applied in Reversing and Tandem Rolling

Cold rolling [57] is performed in order to further reduce thickness and achieve material properties that are suitable for the realization of products where higher thickness precision, suitable flatness profiles and higher surface quality must be guaranteed.

Strip thickness reduction by means of cold rolling can be realized mainly by means of three types of processes that requires different automation solutions in terms of sensors and control technologies:

- *Single stand cold reversing mills* (SCRM): the flat metal strip is processed in several passes (from 3 to 7) and the coil is uncoiled–recoiled by two reels installed in proximity of the stand.
- *Double stand cold reversing mills* (2CRM): the reduction of the thickness is achieved with a reversing process but the number of passes (from 1 to 3) is reduced by increasing the number of stands.
- *Tandem cold mill* (TCM): the thickness reduction is achieved with a number of nonreversing stands (typically from 3 to 5 stands) [25].

In some cases the TCM process is coupled with the pickling process in order to increase the productivity [26,27]. In this case the process is known as *continuous tandem cold mill* (CTCM) because coils are welded together and the process is expected to stop only for maintenance reasons. In this case even the weld between a coil and the following one is subject to rolling (*flying* setup), see, for example, [28].

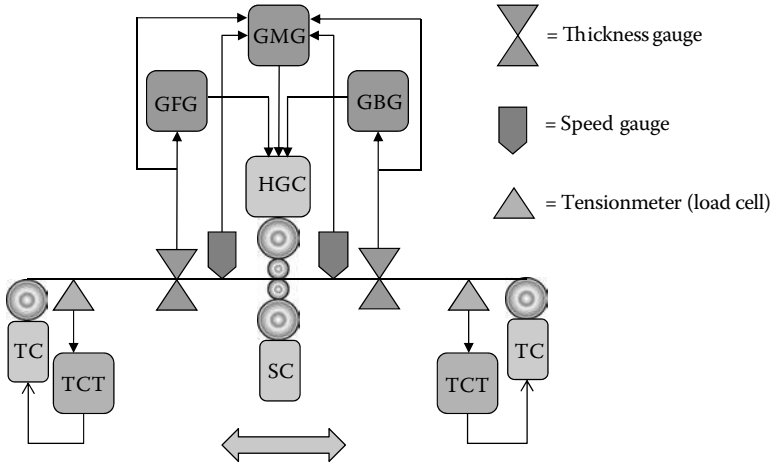

FIGURE 18.8 Thickness control in SCRM.

In Figure 18.8 we depict an example of a possible thickness controller for SCRM and the most common configuration of sensors:

- *Thickness gauges* like for HSM case are based on x-ray technology and are aimed at measuring the thickness in the centerline (and seldom the thickness profile).
- *Speed gauges* are sensors based on laser technologies or are simply encoders. In general the use of laser technology (much more expensive) is preferred when the required measuring precision must be guaranteed also in the presence of fast acceleration/declaration periods, that is, when an encoder can lose contact with the material.
- *Load cells* are in general installed in each interstand in order to get a direct measurement of the interstand tension.

As depicted in Figure 18.8, it is quite common to see SCRM installations provided with thickness and speed sensors (possibly encoders) on both sides of the mill.

18.5.1 AGC—The Realization of Thickness Control in Cold Rolling

In cold rolling (and in particular in tandem rolling) this regulation effect is realized with sophisticated controllers that need to take into account that loopers are not present and consequently the regulation activity of all the stands must be coordinated in order to guarantee stability of the rolling process.

As we have done for HSM rolling in Section 18.3, in the following we will distinguish between *basic* and *external* controllers. The basic controllers {HGC, SM, TC} do not depend on the type of rolling process considered (see Section 18.3) whereas the external controllers can change significantly according to the structure of the process and the availability of sensors.

For the SCRM the external controllers are:

- The *tension control* via *torque* (TCT): The entry/exit tensions are kept constant through the torque regulated by the TC that, in turn, exploits the motors applied to the coiler/uncoiler reels.
- The *gauge control, feedback* via *gap* (GBG): This controller generates a trim for the HGC reference on the basis of the thickness measurement H_{out}^{Xray} available downstream the stand.
- The *gauge control, feedforward* via *gap* (GFG): This controller generates a trim for the HGC reference in order to anticipate the thickness deviations of the incoming strip to be rolled through the x-ray installed on the entry side and producing the measurement H_{in}^{Xray}.

- *Gauge control, mass flow* via *gap* (GMG): This controller aims at compensating the deviations of thickness H_{out} by exploiting the mass flow principle and so the speed measurements of the strip at the entry side and exit side (V_{in} and V_{out}). More precisely, since strip width variations are negligible the following mass flow balance equation is expected to be satisfied:

$$H_{in}^{Xray} V_{in} = H_{out}^{Xray} V_{out}. \tag{18.8}$$

  On the basis of Equation 18.8, it is possible to track the measurement of H_{in}^{Xray} at the entry side of the stand and then get another measurement of the thickness at the exit of the considered stand:

$$H_{out}^{MF} := H_{in}^{Xray} \frac{V_{in}}{V_{out}}. \tag{18.9}$$

  The GMG, by controlling the signal H_{out}^{MF} instead of the signal H_{out}^{Xray}, guarantees a wider stability margin and better performances than the GBG, since there is no transport delay affecting the measure represented by H_{out}^{MF}.

In the 2CRM the HGC applied on stand #1 (Figure 18.9) does not aim at regulating the stand #1 exit thickness directly. Indeed, some regulators are introduced in the 2CRM case in order to keep, as much as possible, constant the interstand tension between stand #0 and stand #1 in order to avoid the generation of disturbances for the GMG/GBG acting on stand #0.

Moreover, the thickness at the exit of stand #1 is regulated by the *gauge control, feedback* via *speed* (GBS). This regulator acts on the speed reference used by the SC applied on stand #1 and, possibly, on the speed reference used by the SC applied on stand #0.

The interstand tension is indeed controlled by two mutually exclusive controllers:

- The *tension control* via *speed* (TCS): This controller regulates the interstand tension by varying the speed reference for the SC applied on the stand #0.
- The *tension control* via *gap* (TCG): This controller acts on the gap reference for the HGC applied on stand #1.

The selection between keeping active the TCG or the TCS depends on the mill speed. Indeed, at low speed the TCS results are in a more prompt controller but, of course, it can interfere with the GBS that is in charge of guaranteeing the final thickness. Consequently, a suitable logic is implemented in order

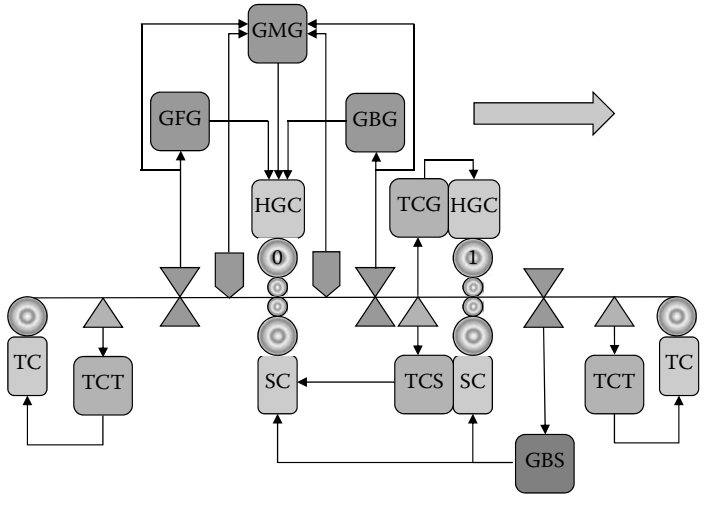

FIGURE 18.9 Thickness control in 2CRM.

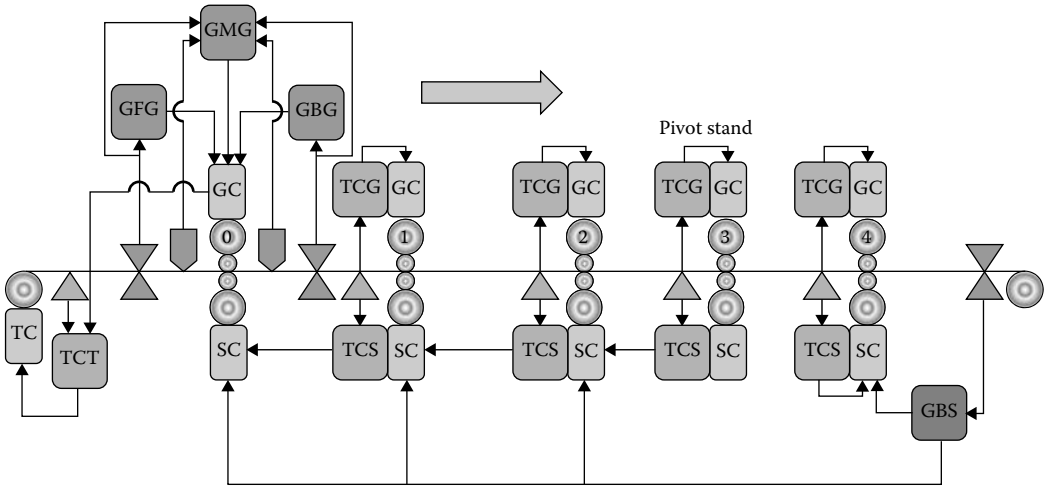

FIGURE 18.10 Thickness control in TCM.

to switch, as soon as possible, from TCS to TCG when the speed reaches a threshold. Of course, in the 2CRM when the rolling direction is inverted the roles of the stands #0 and #1 are inverted and the external controllers are applied with a symmetrical logic.

In TCM [24–27, 29–31] the control logic applied to 2CRM is further extended in order to take into account the contribution of more stands (Figure 18.10) and the corresponding availability of sensors.

Indeed, a typical TCM installation is provided with the following sensors:

- Thickness x-ray at the entry side of stand #0 and at the exit of stand #0.
- Thickness x-ray at the exit of last stand.
- Laser speed meters are in general installed only on entry/exit of stand #0.
- All the interstand speeds and the coiling speeds are measured through encoders.
- All the interstand tensions are measured by load cells.

As in the 2CRM case, GMG/GBG/GFC is applied to the first stand of the tandem (stand #0 in Figure 18.10) whereas the GBS, in charge of regulating the final thickness, can act on the speed references for all the stands. Moreover, as in the 2CRM case, all the interstand tensions are regulated by TCG or TCS.

Finally, as in the HSM case the Speed Master controller (Section 18.3.3) must be implemented in order to coordinate the speeds of the various entities in the mill. This is particularly important in 2CRM/TCM where the interstand tension regulation guaranteed by TCG/TCS is not as fast as that guaranteed by the TCT or by the LCT implemented in HSM through the looper.

18.5.2 Automatic Flatness Control (AFC): Flatness Control in Cold Rolling

The control tasks realized in the Level 1 closed-loop control for *cold rolling* concerns not only the thickness (AGC) but also the flatness (AFC) [32,33].

In *hot rolling*, similar controllers are not so widespread due to the more difficult realization of effective sensors: contact sensors become quickly unreliable due to the wear generated by the high temperature of the strip whereas contactless sensors, when based on optical technologies, give information only about the presence of manifested flatness defects. In any case, most of the concepts presented here are valid for Hot Rolling case as well.

For a strip subject to cold rolling, the flatness is defined as the amount of internal stress difference along the width of the material. The measurement of the strip internal stresses (the so-called *shape*)

during coiling can be taken through suitable sensors named *shapemeters* or *stressometers* that until now represent a significant investment. Due to the cost of these sensors seldom a plant is equipped with more than one flatness sensor, that is, the shapemeter installed at the exit of the mill.

The AFC task is usually performed by exploiting in closed-loop the flatness actuators of the last stand only, since it is the nearest to the shapemeter and it has the most immediate and predictable effect on the coil final flatness.

Differently to the hot rolling case, which exploits almost exclusively stands of 4HI type (i.e., stands with 4 rolls), the rolling stands used for performing cold rolling can have more advanced flatness actuators: in general in TCM/2CRM the stands could be of 4HI type or 6HI type (i.e., stands with 6 rolls). SCRM process can be realized (in particular for stainless steel) with stands of 20HI type also ("Cluster mills") [34,35,58].

18.5.2.1 Shape Measurement System

A conventional shapemeter used in cold rolling consists of an array of load cells distributed along the width of the strip (the interested reader is referred to [3] for other nonconventional sensors). Each load cell produces a signal representing the pressure exercised by the slice of strip in contact with it. Consequently, the shapemeter produces an array of tension signals whose dimension is the amount of load cells placed on the sensor:

$$Shape = [T_1 \; T_2 \; \ldots \; T_L] \, .$$

Recently, contact-less sensors based on ultrasound are available and provide a quite similar array of signals.

It is worth pointing out that the presence of a gradient in the specific tension associated to two different strip slices implies that the two slices will present different elongation values (see Figure 18.11). In turn, an excessive difference in the elongation between the strip slices could imply a manifested flatness defect that should be corrected.

18.5.2.2 The Conventional AFC System based on Least Mean Squares

The problem of correcting a possible flatness defect (i.e., a deviation of the measured shape with respect to the desired shape target) can be faced if the influence of each flatness actuator on each tension measurement T_i is known, where $i = 1, \ldots, L$.

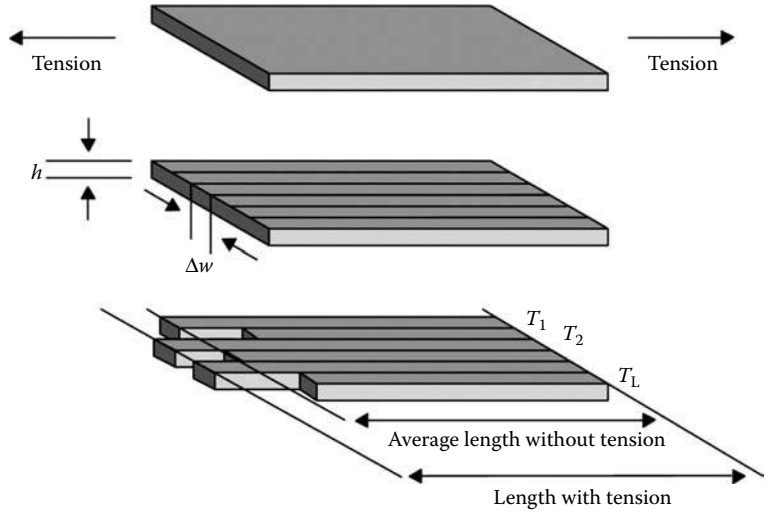

FIGURE 18.11 The flatness defect is induced by differential internal tensions causing differential elongation.

In other words, it is necessary to know the relation

$$\Delta Shape\,(Act) = \begin{bmatrix}\Delta T_1(Act) & \Delta T_2(Act) & \dots & \Delta T_L(Act)\end{bmatrix} = M(Act)\Delta Act \tag{18.10}$$

for each actuator, where *Act* represents the generic flatness actuator and $M(Act)$ is a vector of dimension L representing the *sensitivity matrix* of the generic actuator *Act* on the shape.

The sensitivity matrices $M(Act)$ are computed by the Level 2 automation flatness models that allow the prediction of the deflection as a result of the implementation of the actuator references (Roll Stack Deflection Model, RSDM) and how this deflection modifies the shape of the material (Shape Model (SM)) (the interested reader is referred to [36] and references therein).

Once the matrices $M(Act)$ are known, the closed-loop control problem is conventionally solved by means of LMS. More precisely at each time instant the measured shape error (i.e., the difference between the shape measurement and the shape target) is provided to an optimization tool based on LMS, which computes, by inverting Equation 18.10, the optimal variation of each actuator reference with respect to the current status [35]:

$$\Delta Act^{Corr} = M(Act)^{+}\Delta Shape\_Err.$$

The use of LMS in the field is actually complemented of additional logic that is in charge of avoiding several problems: as an example, it is necessary to execute the control algorithm even when two flatness actuators have quite similar but not identical sensitivities. In principle, in such a situation the bare LMS algorithm could produce, as optimal result, a flatness actuator configuration having some actuators uselessly fighting themselves or a configuration that cannot be actually implemented due to actuator saturation.

18.6 The Use of a Multivariable Controller for Deposited Zinc in HDGL: Introduction and Problem Settling

Nomenclature:

U	Strip speed
M	Zinc coating weight
L	Distance between the air knives and the coating gauge
ρ	Molten zinc density
μ	Molten zinc viscosity
g	Gravity acceleration constant
h	Air knives gap—strip horizontal position
X	Air knives gap—zinc pot vertical position
d	Air knives opening gap
P_0	Air knives supply pressure
T_s	Strip temperature
T_{zp}	Zinc pot temperature

These subscripts will be applied to M with the following meaning:

ref	Zinc coating target
err	Zinc coating regulation error
hot	Zinc coating at the air knives when the zinc is still in molten status
solid	Zinc coating after solidification
solid_meas	Zinc coating measurement produced by a cold coating gauge
solid_pred	Zinc coating estimation produced by a mathematical model
solid_pred_del	Synchronized estimation of the zinc coating with the measurement.

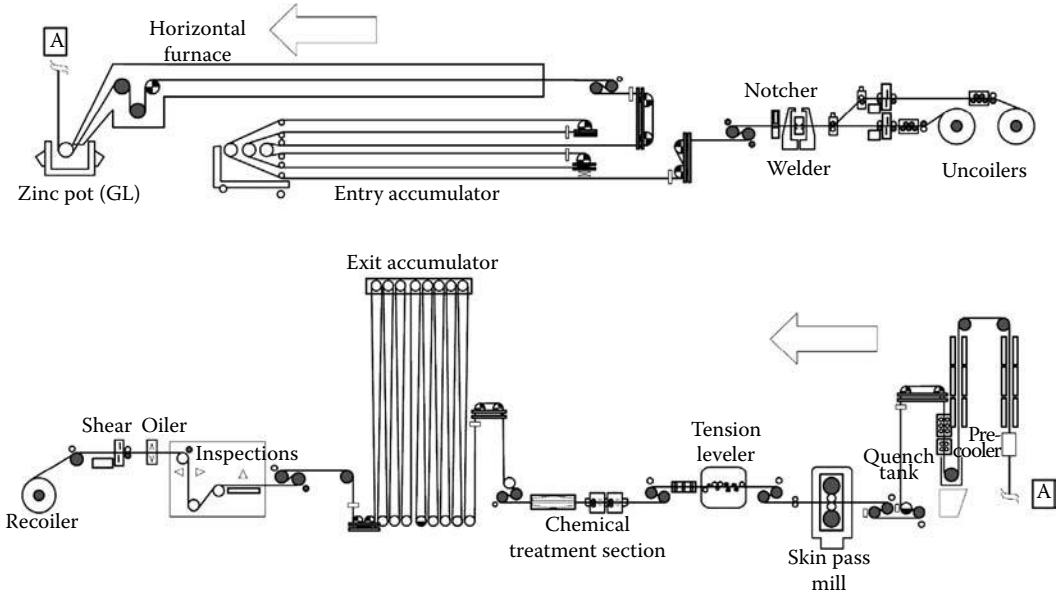

FIGURE 18.12 A possible layout of a HDGL.

The HDGL is constituted of several devices installed consecutively in a continuous line aiming at improving the quality of a flat steel strip. As pointed out in Figure 18.12 the main devices of the HDGL are represented by:

- A coil welder used to weld consecutively several coils together to guarantee the continuity of the process.
- An annealing furnace that guarantees the mechanical properties of the material through material recrystallization.
- A skin pass mill mainly used to correct possible strip flatness defects.
- The coating system.

As far as the zinc coating process is concerned, the HDGL section dedicated to this process (see Figure 18.13 for a detail) includes the following devices installed after the furnace:

- When the strip leaves the furnace, it is then immersed in a zinc pot of a molten bath of a zinc alloy.
- As soon as the strip emerges from the zinc pot the *air knives* (AK) (i.e., the device is constituted of two air jets, one for each surface of the strip) then aim at reducing the thickness of the zinc alloy film (coating weight) through air jet stripping action.
- The strip is then transported along a cooling turret in order to solidify the zinc alloy film.
- Finally, the coating weight is measured by means of an x-ray device (*cold coating gauge*).

18.6.1 The Purpose of a Coating Weight Closed-Loop Control System and Performance Definition

The target M_{ref} for a possible closed-loop controller must be chosen taking into account the standard deviation guaranteed by the system, $\sigma(M_{solid\_meas})$, and by the minimal coating weight to be guaranteed for the final application, $M_{\min\_ref}$.

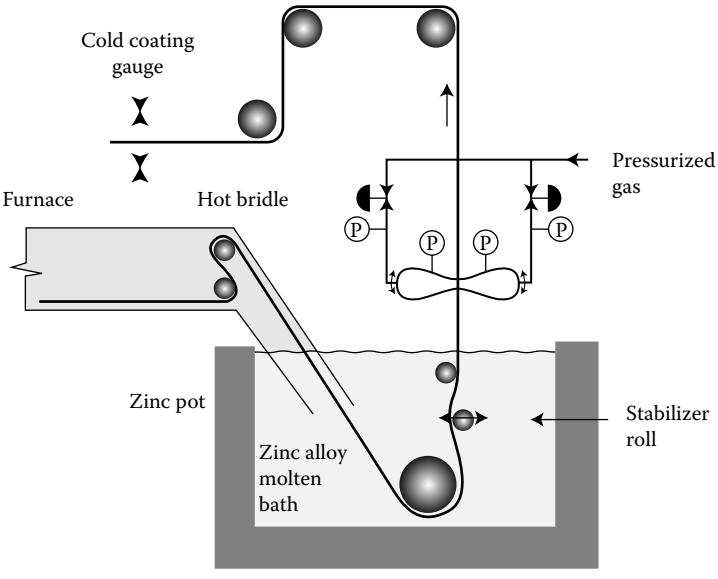

FIGURE 18.13 The zinc coating process—P represents the installation of a pressure transducer.

Indeed, it is a common practice to measure the zinc coating efficiency according to the following rule

$$Over\text{-}Coat\ ratio = 1000\, Average \frac{(M_{solid\_meas})}{M_{\min\_ref}} \tag{18.11}$$

and, consequently, it is an advisable practice to choose the target M_{ref} with a rule of the following type:

$$M_{ref} = M_{\min\_ref} + 2\sigma(M_{solid\_meas}).$$

Since the attitude of an operator acting in manual mode on the coating process is to use more zinc than what is strictly necessary to avoid the possible occurrence of defects (*under coating*), the purpose of a closed-loop controller is to guarantee without supervision the *Over-Coat ratio* to be as much as possible near to 1000 (but never under 1000) so as to save as much zinc alloy as possible while avoiding possible under coating.

18.6.2 The Use of a Cold Coating Gauge in Closed-Loop Control

A possible coating weight closed-loop controller must take into account a number of requirements. One of the main points is represented by the necessity of compensating the considerable measurement delay represented by the distance between the Cold Coating Gauge and the AK (see [37] for an analysis of the robustness margin of a possible closed-loop controller).

In order to solve this stability problem definitively the most exploited method is the use of the Smith Predictor concept (Figure 18.14), that is, the introduction of a prediction model inside the closed loop [38,39,59].

The use of the Smith Predictor concepts aims at increasing the stability margin and consequently the possibility of reaching the desired coating weight target in a reduced amount of strip meters. With such a type of control rationale it is expected that the final target is reached for the first time in 1–3 times the distance between the actuator and the measurement device.

High-frequency disturbances can be compensated by means of a Smith Predictor based closed-loop approach and for this reason, as will be detailed later, feedforward compensation concepts must be superimposed to guarantee disturbance rejection.

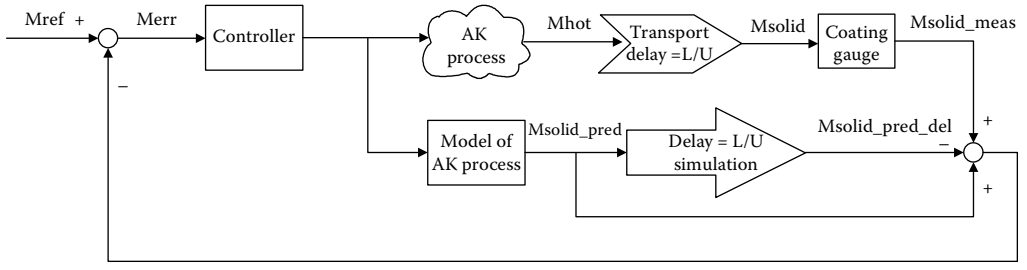

FIGURE 18.14 The Smith Predictor concept applied to the closed-loop coating control.

In the real practice, the vibration of the strip due to the elastic deformation of the strip introduces significant disturbance with a high-frequency content that in general are hard to compensate by the AK actuators. This is the main reason why in the recent years magnetic damping systems have been introduced in the field [40].

The use of a *hot coating gauge*, that is, a coating measurement device installed just after the AK device, has been also considered in the field and has been installed in some plants. Of course this type of measurement device significantly increases the stability margin of a possible closed-loop control without the use of the Smith Predictor type controller, since it drastically reduces the transport delay but, on the other hand, it is still not commonly used mainly due to the costs associated to the frequent necessary maintenance and for the difficulties related to the installation (the hot coating gauge needs to be installed together with a strip vibration damping system to get a reliable measurement).

The installation of the vibration damping system, on the contrary, is not strictly required with a cold coating gauge system because, in general, it is installed on the bridle just before the skin pass mill (Figure 18.12) where the strip is not subject to vibrations anymore.

In the following part of this article, we will treat only the case of the cold coating gauge because it definitely corresponds to the most widespread and consolidated coating sensor technology used in the field [38,39].

18.6.3 The Purpose of a Closed-Loop Multivariable Controller

The control problem to be solved has a multivariable nature because of the high number of control variables and measured variables. It is quite important also to note that the multivariable nature of this control problem mainly derives from the fact that the coating weight regulation cannot be realized independently for the two faces of the strip. More precisely, an excessive difference for the actuation references applied to the control variables for the two jets has several consequences that in turn could imply the instability of coating process.

For instance an excessive difference between the two AK in terms of supply pressure P_0, horizontal position h or AK gap opening d, implies a movement of the strip with respect to the machine centerline of the AK device due to a bending effect impressed to the strip. To the best knowledge of the authors the coordination of the two air jets, the main obstacle to achieve a full automatic control system, has never been treated till now in the literature.

18.6.3.1 Selection of the Control Variables

The set of the control variables to be selected can change significantly according to the available mechanical solution:

- The inlet AK jets air pressure, P_0.
- The horizontal position of the AK apparatus with respect to the strip, h.

- The vertical position of the AK apparatus with respect to the Zinc Pot, X.
- The AK opening gap [38], d.

The privileged control variables for this closed-loop control problem are represented by P_0 and h both because they represent the most general control variable choice (they correspond to suitable actuators always available in any installation) and also because they have an easily predictable effect on the coating weight.

In many installations h, representing the distance of the AK nozzle from the strip, is not directly measured and it is only estimated by using absolute encoders. Consequently, the available estimation of h should be considered subject to low-frequency disturbances due to:

1. A possible wrong calibration of the measurement encoders.
2. A possible wrong calibration of the strip pass line.
3. A possible curvature of the strip caused, as reported before, by an excessive difference between the setup applied for the two AK or by the curvature resistance of the strip that tends to deviate from the centerline at the stabilizer roll (Figure 18.13).

The vertical position alone does not provide a flexible control variable because its sensitivity with respect to M in some cases could turn out small and, consequently, ineffective for a closed-loop controller.

The jet gap d seldomly can be actuated automatically and in general is left to a mechanical pre-tuning of the AK device. Nowadays, there exist some mechanical installations where the gap can be automatically set through servomotors. The control procedure proposed in [38] allows regulating the coating weight profile through several gap actuators installed along the AK lip. This type of mechanical solution is, in principle, very promising but it still remains extremely rare.

In the following the section presents the implementation of a multivariable controller using as control variables the pair $\{P_0$ and $h\}$ because this choice adheres to the most widespread mechanical devices and guarantees to maintain uniform performance in all possible conditions without any supervision.

18.6.3.2 Definition of the Controlled Variables

The most widespread cold coating gauges are based on a single x-ray source mounted on tracks so as to continuously travel back and forth, transverse to the movement of the strip. Consequently, the coating weight profile is available not only along the length of the coil but also along the width of the coil and for both the surfaces of the strip (*front* surface and *rear* surface).

In general, according to the travel speed of the track several measurements are taken on the strip along its width (*coating weight profile*). Let us introduce the following convention: $M_{solid\_meas}(p)$ is the measurement collected at the normalized coordinate $p \in [-1, +1]$ with respect to the strip centerline where -1 represents the nonblower side and $+1$ the blower side.

Since, only very particular and expensive mechanical devices [38] allow to correct deviations of the whole vector of measurement $M_{solid\_meas}(\cdot)$ from the target M_{ref} we will concentrate on the possibility of regulating two main signals derived from the vector $M_{solid\_meas}(\cdot)$:

$$M_{avg\_solid\_meas} = \underset{p\in[-1,+1]}{Average}\{M_{solid\_meas}(p)\} \tag{18.12}$$

and

$$M_{dif\_solid\_meas} = M_{solid\_meas}(-1) - M_{solid\_meas}(+1). \tag{18.13}$$

In the following, for the sake of simplicity, the symbol M_{solid_meas} will be used to represent the vector of signals $[M_{avg\_solid\_meas}, M_{dif\_solid\_meas}]$ because this represents the set of controlled variables.

As it will be explained better later, a number of controlled variables can be exploited successfully for the control of $M_{avg\_solid\_meas}$ (between them also the average horizontal position h). On the other hand,

definitively the most widespread mechanical solutions can regulate $M_{dif\_solid\_meas}$ in closed loop by acting on the differential horizontal position represented by the following variable:

$$h_{dif} = h(-1) - h(+1). \tag{18.14}$$

18.6.4 Purpose of Feedforward Compensation

As pointed out before, the closed-loop controller based on a Cold Coating Gauge is not able to compensate high-frequency disturbances. Furthermore, the coating line could be subject to sudden variations of the operating conditions.

In practical situations, it is advisable to consider compensating as fast as possible, in a feedforward way, variations of:

- The strip speed U.
- The strip temperature at the exit of the furnace T_s.

In general, step variations on these process variables correspond to line faults but due to the continuous nature of the process it is important to guarantee the quality of the coating process even in these particular situations.

A step variation in the *line processing speed* can be compensated in a feedforward way through a coordinated variation of the AK supply pressures and vertical position. Indeed, a variation of the line speed could imply a drastic variation of the zinc mass flow rate to be removed from the strip surfaces and the compensation with supply pressures only could turn out impractical. Typical requirements are to manage line accelerations/decelerations from 50 mpm (low speed) to 200 mpm (full speed).

The variation of the furnace strip exit temperature are not step-like but could have significant variations that have strong impacts on the strip surface zinc adhesion (for instance, in case of a furnace fault the strip temperature could be subject to a variation of 50°C in some minutes).

In order to implement these two feedforward compensation actions the controller depicted in Figure 18.14 must be complemented with suitable routines as described in Figure 18.15.

The feedforward compensation routines are actually based to the same prediction mathematical model used for implementing the Smith Predictor concept: these routines implement a *gradient method* aiming at maintaining the current coating weight M_{solid_meas} independently of the measured variation on the strip speed and temperature.

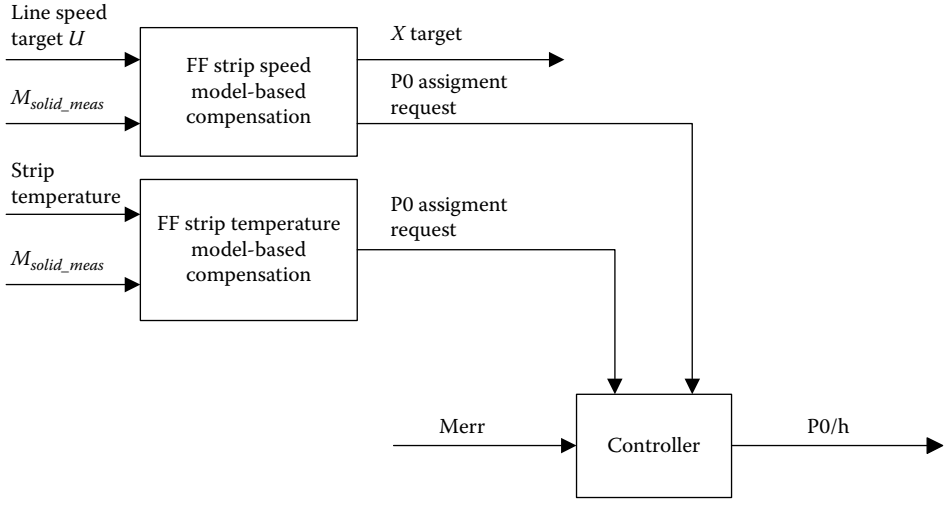

FIGURE 18.15 Feedforward (FF) compensations.

18.6.5 Coating Mathematical Model and Its Implementation

It is a common conviction that it is almost impossible to generate highly accurate physical film process models (see [41] as a recent survey on the steel/paper industry) or that implementation of accurate physical models is not suitable for real-time closed-loop controls.

This is the main reason why in several previous contributions in the literature the Smith Predictor concept is implemented through extremely simplified models [42] or black-box models [43,44]).

The physical coating process has been actually studied for many years [45,46] and it is still under study [47]. The recent advances in terms of computer hardware performance do not put any particular limit on the real implementation of complex and reliable coating prediction models for this field.

The mathematical model used for the implementation of the Smith Predictor and the feedforward compensation functions of Figures 18.14 and 18.15 is actually derived from [47] and it is complemented of a recursive identification function.

18.6.5.1 Implementation Issues

As in many other automation systems for flat metal production, the realization of complex control loops is obtained through the cooperation of Level 1 automation with the Level 2 automation (see Figure 18.16).

The Level 1 automation is in charge of the implementation of the communication with the sensors and actuators and it is the natural place to implement the necessary material real-time *tracking functions* for the real implementation of the Smith Predictor concept. For instance, Level 1 automation is in charge of simulating the transportation delay and to track the model predictions to the real measurement point (Figure 18.14).

On the other hand, the Level 2 is in charge of executing the mathematical model before the events occurring in the process. For instance, the execution of the model for Smith Predictor implementation is realized every 3 m of strip (i.e., every second, assuming the line at a full speed of 200 mpm). The CPU potentialities actually allow even fast closed-loop controllers and, in many installations, the real bottle-neck is represented by the acquisition speed of the coating weight sensor.

18.6.5.2 Model Main Characteristics

The mathematical model developed in [46,48] and refined in [47], where a heat-transfer model has been included for estimating the solidification rate, represents a static function allowing to predict the coating weight as a function of the main AK process variables. More precisely, the coating mathematical model for driving the prediction M_{solid_pred} can be represented as a function of the following type (see Figure 18.17

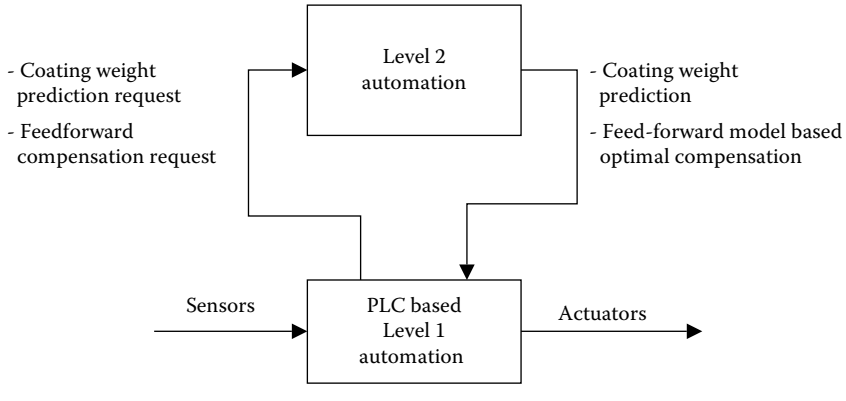

FIGURE 18.16 Coating weight closed-loop implementation.

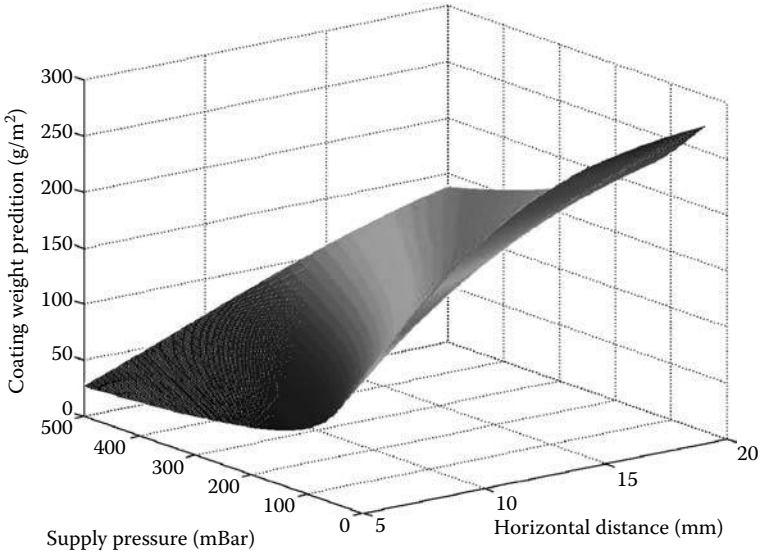

FIGURE 18.17 Coating weight predictions for $d = 1.2$ mm and $X = 400$ mm.

for typical coating weight predictions produced by the mathematical model):

$$M_{solid\_pred} = f(U, h, d, P_0, \rho, \mu, T_s, T_{zp}). \tag{18.15}$$

The derivation of the mathematical model is based on a simplified two-dimensional Navier–Stokes equation for a thin film:

$$\mu \frac{d^2u}{dy^2} - \left(\rho g + \frac{dp}{dx}\right) = 0$$

with the boundary conditions

$$u|_{y=0} = U, \quad \mu \left.\frac{du}{dx}\right|_{y=W_{solid\_pred}} = \tau, \quad W_{solid\_pred} = \rho\, M_{solid\_pred}$$

where x represents the vertical coordinate along the strip length, y is the coordinate perpendicular to the strip, p is the pressure along the strip, u is molten zinc speed and τ is the shear stress imposed on the film by the air jet.

This mathematical model can be solved in a reasonable computational time fully compatible with a real-time closed-loop controller. Indeed, after some simple elaboration the estimation of the coating weight can be retrieved as the solution of a second-order algebraic equation:

$$\frac{1}{\rho} M_{solid\_pred} = \frac{S \pm \sqrt{S^2 + 4G}}{2G}$$

where S is the nondimensional shear stress and G is the nondimensional effective gravitational acceleration. The interested reader is referred to [47] for further details.

As depicted in Figure 18.17, the coating process mathematical model produces an estimation of the coating weight in a wide range of operating conditions. In particular it is apparent that once some main process parameters are fixed (in particular d, U, X, T_s, and T_{zp}) the same coating weight target can be reached with an infinite number of combinations for the two main control variables P_0 and h.

It is also quite important to observe that all these solutions are not equivalent from a control point of view for two main reasons:

- Not all possible values for P_0 can be actuated with the same precision due to the nonlinear characteristic of the valve used for controlling the inlet pressurized air
- The sensitivities of the control variables with respect to the predicted coating weight must be kept limited between a minimal and a maximal value in order to guarantee good regulation margins:

$$\min < \frac{dM_{solid\_pred}}{dP_0} < \max, \quad \min < \frac{dM_{solid\_pred}}{dh} < \max. \tag{18.16}$$

The need to satisfy the constraints (18.16) implies that the mathematical model to be used must guarantee numerical precision also in terms of prediction for the sensitivities dM_{solid_pred}/dh and dM_{solid_pred}/dP_0 (possible values for these predicted sensitivities are reported in Figures 18.18 and 18.19, respectively).

The values reported in Figures 18.18 and 18.19 need to be evaluated taking into account the accuracy reachable by the internal regulators for P_0 and h. Of course, the accuracy of these internal regulators strongly depends on the mechanical solution but it is worthwhile evaluating the real control capability of each control variable on the coating weight.

The following considerations can lead to quite different results from installation to installation but are strictly necessary in order to have a deep understanding of the performance reachable by a closed-loop controller:

- Assuming that the operative working point corresponds to $dM_{solid\_pred}/dP_0 = -0.3$ g/m^2/mBar and assuming that the internal controller on P_0 guarantees a precision of 1.0 mBar in the same conditions, then a controller acting on the supply pressure is able to correct errors in the coating weight of not more than 0.3 g/m^2.

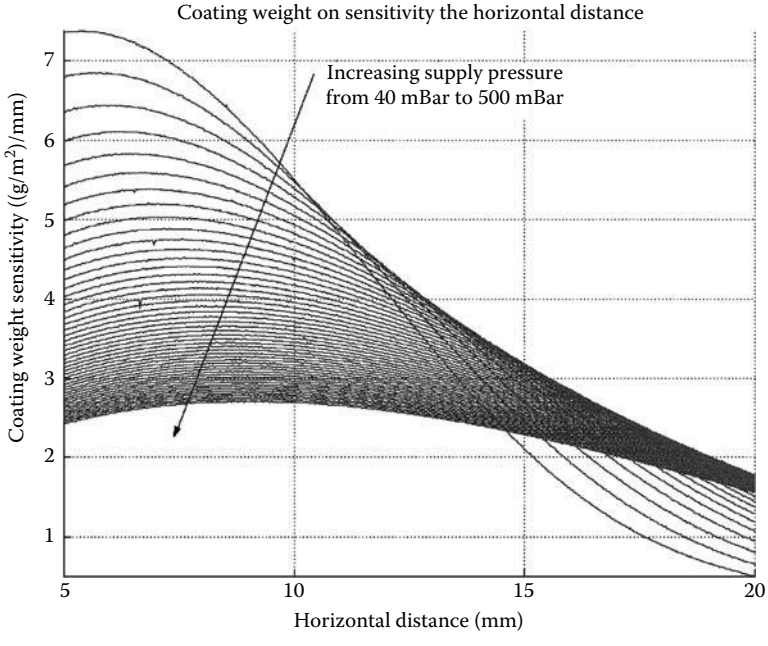

FIGURE 18.18 Predicted coating weight sensitivity on the horizontal distance.

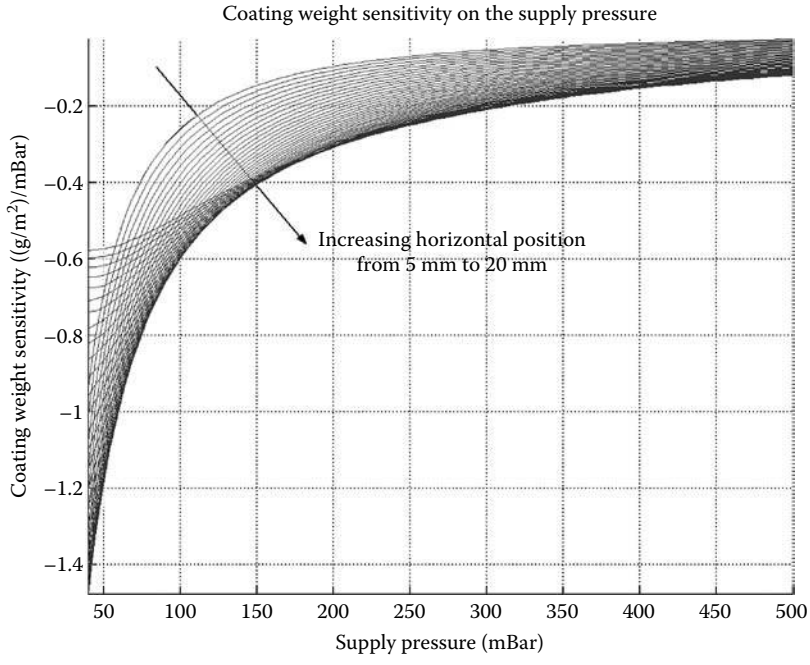

FIGURE 18.19 Predicted coating weight sensitivity on the supply pressure.

- It is necessary to take into consideration that the precision of the internal regulator for P_0 cannot be considered independent of the operative working point due to the nonlinear characteristic of the valve.
- A typical value for dM_{solid_pred}/dh is 4.0 g/m^2/mBar whereas the accuracy that can be reached by the internal controller for h is 0.1 mm (this accuracy in general does not depend on the operative working point but it could depend on long-term wear effects). In turn, this implies that a controller acting on the horizontal pressure is able to correct errors on the coating weight of not more than 0.4 g/m^2.

It is practical plant experience to consider P_0 as the most precise control variable but of course this strongly depends on the mechanical installation characteristics.

18.6.5.3 Coating Weight Model Prediction Performance

Most of the plants are not equipped of direct measurement of the distance between the AK and the strip surface. As already explained, the estimation of h is subject to an uncertain offset that can be estimated by means of a long-term autoregressive technique (that can be of course quite different for the *front* side and the *rear* side of the strip).

More precisely, in order to perform closed-loop control and feedforward compensation it is advisable to introduce in Equation 18.15 a recursively estimated parameter as follows:

$$M_{solid\_pred} = f(U, h + Offset, d, P_0, \rho, \mu, T_s, T_{zp}).$$

As explained before, most of the uncertainty associated to h has a low-frequency content and consequently the *Offset* term introduced in the model execution can be estimated from coil to coil. In the end the estimation performance obtained in case a direct measurement of h is not available is presented in Figure 18.20. Of course, the availability of a direct measurement of h leads to definitively better prediction results.

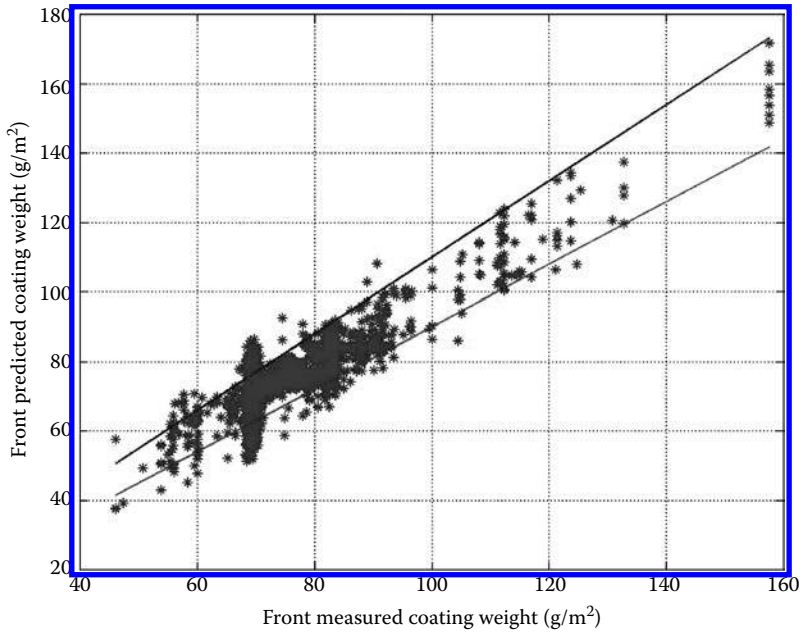

FIGURE 18.20 Coating weight prediction performance.

The autoregressive estimation technique is based on the following simple algorithm concept:

```
Iter=0; Offset=0;
For each coil, as soon as the coil tail leaves the mill
{
  Compute OptOffset such that;
```

$$\min_{\texttt{OptOffset}} \texttt{J} = \sum_{\texttt{for all collected samples}} \times (\texttt{Msolid\_meas} - \texttt{f}(\texttt{U}, \texttt{h} + \texttt{OptOffset}, \texttt{d}, \texttt{P}_0, \rho, \mu, \texttt{T}_\texttt{s}, \texttt{T}_{\texttt{zp}}))^2$$

$$\texttt{Offset} := (1.0 - \lambda)\texttt{Offset} + \lambda\texttt{OptOffset};$$

$$\texttt{Iter} := \texttt{Iter} + 1$$

```
};
```

The algorithm needs to be executed taking into account two points:

- The coefficient $\lambda \in [0, 1]$ can be adjusted automatically in the time, that is, according to the counter *Iter* and according the variance of prediction error.
- The *Offset* optimization parameter should be considered varying for different classes of coating weights and coil thicknesses. This is advisable in order to compensate not only the uncertainty in the measurement of *h* but also the uncertainty in the model itself. Indeed, as presented also in [47] the reliability of the bare model prediction degrades for high values of the coating weight.

18.6.6 Basic Controllers: Supply Pressure Control and Horizontal Position Control

The main problem in the actuation of the AK horizontal position *h* is related to the stick-slip friction often present in a mechanical position actuator based on an electrically commanded servomotor.

In order to compensate possible problems associated with friction phenomena, often slowly varying in the time according to the wear and lubrication, one of the most effective and simple control technologies is represented by the use of *sliding mode* (see [8] and references quoted therein). In this type of application a first-order sliding mode controller has been applied.

The problem of regulating the supply pressure P_0 could turn out to be more complex under several aspects:

- First of all, P_0 is actuated not only through the inlet valve opening command but also by the air blower set point.
- The inlet valve often has a nonlinear characteristic that cannot be easily compensated and can be different according to the valve supplier.
- The electrical power consumption associated to the air blower is not a negligible parameter.

The reasons just presented lead to the consideration that a cascade controller such as the one proposed in Figure 18.21 leads to a twofold advantage of keeping the valve in an operating point corresponding to the linear part of the valve characteristic and the reduction of the electrical blower electrical power consumption.

18.6.7 Structure of the Multivariable Controller

The multivariable controller consists of three different algorithms aiming at solving the following three problems:

1. The *setup control algorithm* is implemented in the Level 2 automation. The trigger time instant is represented by the violation of the constraints (Equation 18.16). It consists in solving the following optimization problem through a gradient-based optimization technique.

$$\underset{X,h,P_0}{Min} |M_{solid\_pred} - M_{ref}| \tag{18.17}$$

   subject to constraints (Equation 18.16).
2. The *feedforward compensation* algorithm is implemented in the Level 2 automation and it is triggered by a request generated by the Level 1 automation according to the perceived variations in the line speed reference U or in the measured strip temperature outside the furnace T_s:

$$\underset{X,P_0}{Min} |M_{solid\_pred} - M_{solid\_meas}| \tag{18.18}$$

   subject to constraints (Equation 18.16).
   The problem of Equation 18.18 is different from that of Equation 18.17 because it does not aim at reaching the final target M_{ref} but aims at maintaining the current measured value M_{solid_meas}.

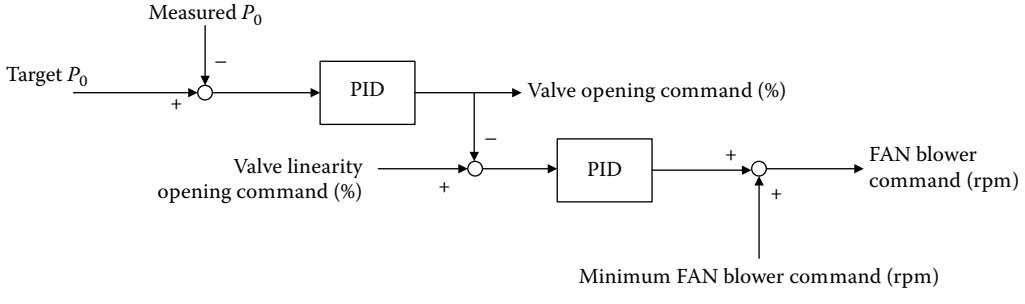

FIGURE 18.21 AK supply pressure controller.

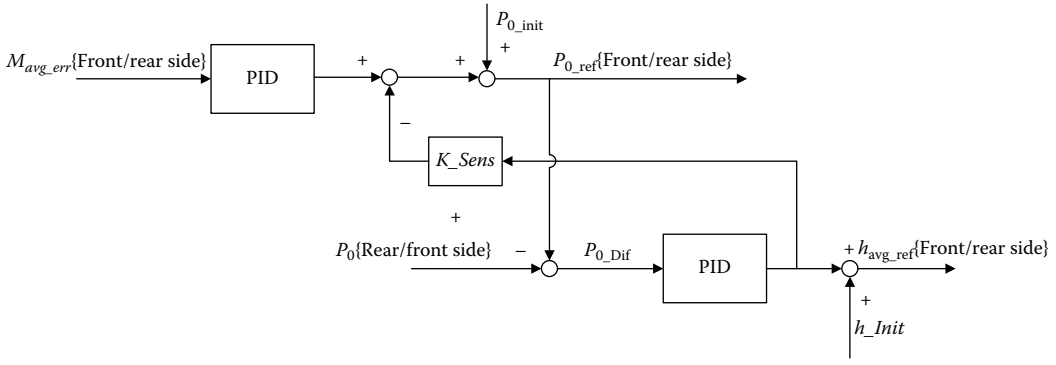

FIGURE 18.22 Multivariable closed-loop controller.

3. The *multivariable closed-loop controller*: it is implemented in the Level 1 automation and consists in the execution of the "Controller" block in Figure 18.14. The details about the structure of this block are given in Figure 18.22.

As depicted in Figure 18.22, this controller produces some references for the supply pressure (P_{0_Ref}) and for the horizontal position (h_{avg_ref}) to be transmitted to the internal controllers. More precisely, P_{0_Ref} is decided in order to compensate the signal M_{avg_err} generated through the Smith Predictor rationale pointed out in Figure 18.14 whereas h_{avg_ref} is decided in order to reduce the pressure difference between the *rear* and *front* sides.

Finally, the constant *K_Sens* is used to decouple the effects produced by the two PID controllers and can be defined by exploiting the coating weight sensitivities produced from the model as follows:

$$K\_Sens := \frac{(dM_{solid\_pred}/dP_0)}{(dM_{solid\_pred}/dh)}.$$

The error, M_{avg_err} feeding the closed-loop controller of Figure 18.22 is obtained through the Smith Predictor concept (see Figure 18.14 and Equation 18.12) as follows:

$$M_{avg\_err} := M_{ref} - (M_{avg\_solid\_meas} - M_{avg\_solid\_pred} + M_{avg\_solid\_pred\_del}). \tag{18.19}$$

It is worthwhile noting that the implementation of the signal of Equation 18.19 corresponds to the following requirements:

- The Level 2 automation should run the mathematical model of the coating process at least three times, that is, not only for the central horizontal reference $h_{ref}(0)$ (to produce $M_{avg\_solid\_pred}$) but also for the following two positions $h_{ref}(-1)$ and $h_{ref}(+1)$ (to produce $M_{dif\_solid\_pred}$).
- The Level 1 automation must be provided of a suitable *tracking* system in order to track the estimation $M_{avg\_solid\_pred}$ referred to the AK device position to the position where the Cold Coating Gauge is installed, that is, by "simulating" even during acceleration/deceleration periods the transport delay. The output of this tracking system is represented by the signal $M_{avg\_solid\_pred\_del}$.

18.6.8 Performances Achieved

In this section, we present the results obtained with the coating weight controller presented on a real plant where the distance between the AK device and the cold coating gauge is about 100 m.

First of all, we considered switching on the controller in the middle of a coil in order to measure the settling performance. In Figure 18.23, we report the measured average coating historical record. The target to be imposed is 80 g/m^2. The tolerance band of 1 g/m^2 is reached in about 150 m of the strip.

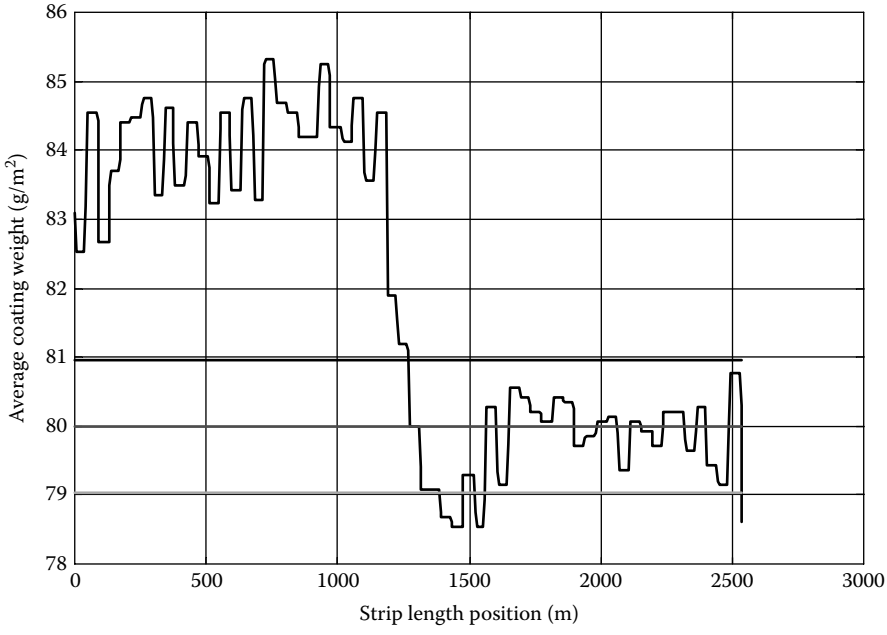

FIGURE 18.23 Measurement of the settling performance.

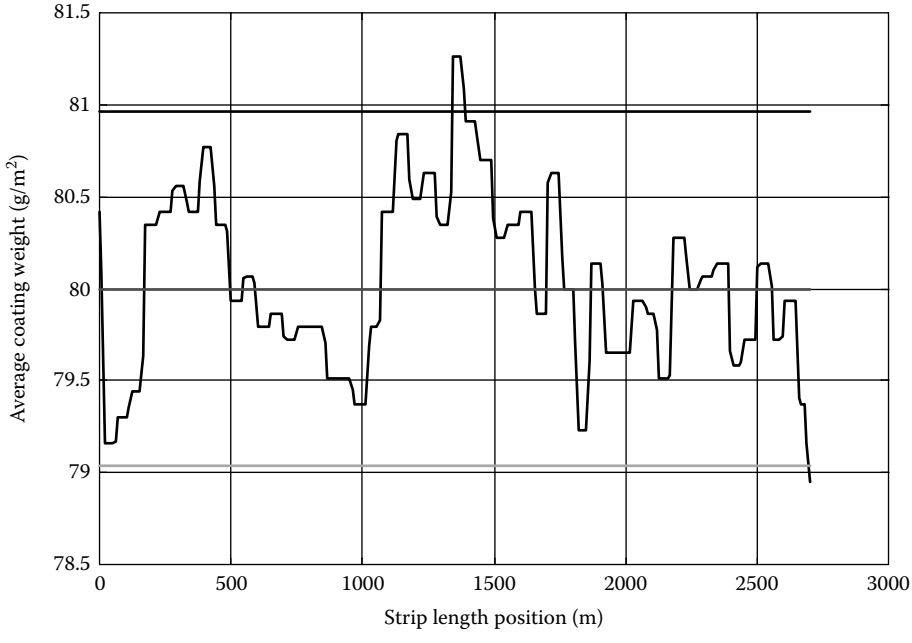

FIGURE 18.24 Long-term performance (average measured coating weight).

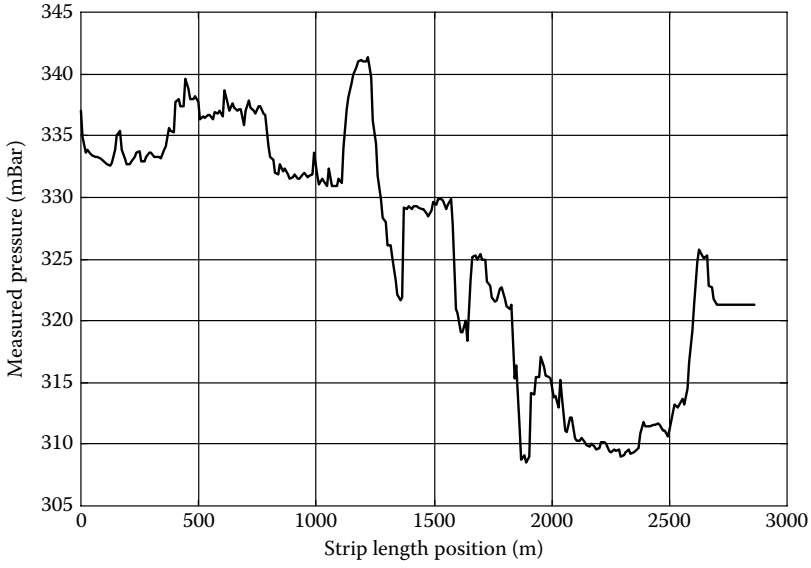

FIGURE 18.25 Long-term performance (AK measured supply pressure).

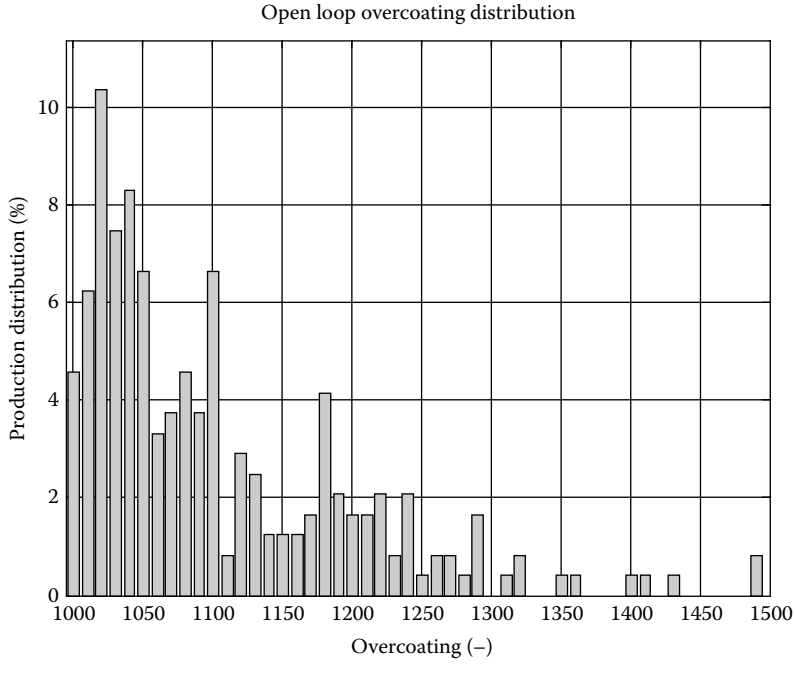

FIGURE 18.26 Measured over-coating during open-loop production.

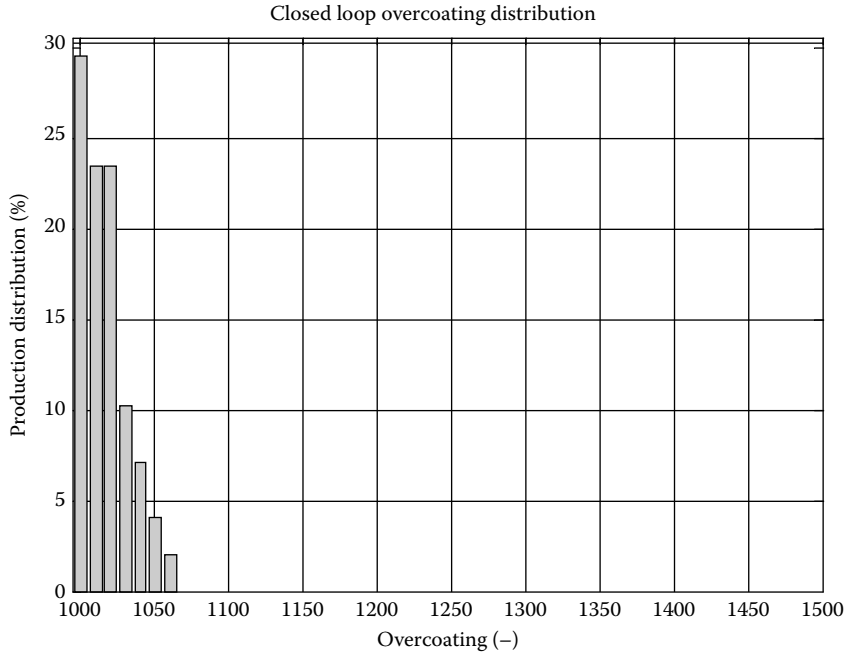

FIGURE 18.27 Measured over-coating during closed-loop production.

In Figures 18.24 and 18.25, we present the performance that can be guaranteed in long-term campaigns of production (i.e., the controller is kept switched on during the production of coils having the same characteristics).

As for long-term performance, the most interesting aspect is represented by the amount of possible over-coating (see definition in Equation 18.11). In Figures 18.26 and 18.27, we present the statistical distribution of the production in terms of over-coating ranges from 1000 to 1500 in open-loop and in closed-loop, respectively.

The difference pointed out in these last figures derives not only from the natural attitude of the operator of being cautious but also from the difficulties in proper manual intervention when any unexpected situation happens.

18.7 Conclusions

This article reviews the automation solutions and the control problems that need to be addressed for the realization of plants for the production of flat metal strip. In particular it presents a multivariable control architecture for the regulation of the zinc coating weight in HDGL together with all the topics that need to be tackled in order to achieve an effective implementation of such a closed-loop controller. Indeed, how a mathematical model is implemented and exploited for the closed-loop task is presented together with a discussion of the typical plant characteristics and on the possible control variables. Finally, the performances achieved with such a controller are presented.

Acknowledgment

The authors thank Filippo Bertoli, Marco Roddaro, and Massimo Filippo (Danieli Automation, Italy) for their assistance during the commissioning of the multivariable coating controller for galvanizing lines.

References

1. A. Tselikov, *Stress and Strain in Metal Rolling*, MIR Publishers, Moscow, 1967.
2. W.L. Roberts, *Flat Processing of Steel*, Marcel Dekker, New York, 1988.
3. V.B. Ginzburg, Ed., *Flat-Rolled Steel Processes: Advanced Technologies*, Taylor & Francis Group, Broken Sound Parkway NW, 2009.
4. C. Aurora, D. Cettolo, and F.A. Cuzzola, Cut scheduling optimization in plate mill finishing area through mixed-integer linear programming, *IEEE Transactions on Control System Technology*, Vol. 18, No. 1, pp. 118–127, 2010.
5. R. Takahashi, State of the art in hot rolling process control, *Control Engineering Practice*, Vol. 9, No. 9, pp. 987–993, 2001.
6. K. Fukushima, Y. Tsuji, S. Ueno, Y. Anbe, K. Sekiguchi, and Y. Seki, Looper optimal multivariable control for hot strip finishing mill, *Transactions Iron Steel Inst. Japan*, Vol. 28, pp. 463–469, 1988.
7. G. van Ditzhuijzen, The controlled cooling of hot rolled strip: a combination of physical modeling, control problems and practical adaption, *IEEE Transactions on Automatic Control*, Vol. 38, No. 7, pp. 1060–1065, 1993.
8. R. Furlan, F.A. Cuzzola, and T. Parisini, Friction compensation in the interstand looper of hot strip mills: A sliding mode control approach, *Control Engineering Practice*, Vol. 16, No. 2, pp. 214–224, 2008.
9. T. Mroz, G. Hearns, T. Bilkhu, K.J. Burnham, and J.G. Linden, Predictive profile control for a hot strip mill, *19th International Conference on Systems Engineering*, pp. 260–265, 2008.
10. I. Malloci, J. Daafouz, C. Iung, R. Bonidal, and P. Szczepanski, Robust steering control of hot strip mill, *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 4, pp. 908–917, 2010.
11. F.A. Cuzzola and N. Dieta, Camber and wedge compensation in hot strip rolling, *IFAC Workshop on New Technologies for Automation of Metallurgical Industry*, Shanghai, China, 2003.
12. G. Hearns and M.J. Grimble, Robust multivariable control for hot strip mill, *Transactions Iron Steel Inst. Japan*, Vol. 40, No. 10, pp. 995–1002, 2000.
13. F.A. Cuzzola, A multivariable and multi-objective approach for the control of hot-strip mills, *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 128, No. 4, pp. 856–868, 2006.
14. T. Hesketh, Y.A. Jiang, D.J. Clements, D.H. Butler, and R. van der Laan, Controller design for hot strip finishing mills, *IEEE Transactions on Control Systems Technology*, Vol. 6, No. 2, pp. 208–219, 1998.
15. M. Okada, M. Murayama, A. Urano, Y. Iwasaki, A. Kawano, and H. Shiomi, Optimal control system for hot strip finishing mill, *Control Engineering in Practice*, Vol. 6, pp. 1029–1034, 1998.
16. Y. Seki, K. Sekiguchi, Y. Anbe, K. Fukushima, Y. Tsuji, and S. Ueno, Optimal multivariable looper control for hot strip finishing mill, *IEEE Transactions on Industry Applications*, Vol. 27, No. 1, pp. 124–130, 1991.
17. G. Hearns, P. Reeve, P. Smith, and T. Bilkhu, Hot strip mill multivariable mass flow control, *IEE Proceedings, Control Theory and Applications*, Vol. 151, No. 4, pp. 386–394, 2004.
18. B. Frisch, W.R. Thiele, and N. Muller, Determination of the pickling time during the descaling of steel in HCl, *Stahl und Eisen.*, Vol. 93, No. 15, pp. 673–679, 1973.
19. F. Pempera, D. Gruchot, and M. Turchetto, The Turboflo concept realized in the new Corus pickling and Bregal pickling and hot dip galvanising lines, *Metallurgical Plant and Technology International*, Vol. 24, No. 6, 2001.
20. T. Nakamura, H. Okoshi, Y. Kani, and H. Sugawara, Continuous annealing, pickling and galvanizing for production of surface-treated steel sheet, *Hitachi Review*, Vol. 45, No. 6 pp. 283–288, 1996.
21. D. Annika, Model-based control system for pickling lines, *Iron and Steel Engineer*, Vol. 74, No. 1, pp. 47–50, 1997.
22. L. Isopescu, P.M. Frank, and G. Gonsior, Modelling of a turbulence pickling line for adaptive control, *Proceedings of the European Control Conference*, Karlsruhe, Germany, 1999.
23. D. Annika and T. Heinz, Model for turbulence pickling lines, *Metallurgical Plant and Technology International*, Vol. 17, No. 2, pp. 70–75, 1994.
24. W.J. Edwards, Design of entry strip thickness controls for tandem cold mills, *Automatica*, Vol. 14, pp. 429–441, 1978.
25. C.F. Bryant, *Automation of Tandem Mills*, British Iron and Steel Institute, London, UK, 1973.
26. M. Tomasic and J. Felkl, Rolling of transitions in a continuous Tandem cold mill, *Proceedings of the 9th International and 4th European Steel Rolling Conference*, Paris, France, 2006.
27. C. Binroth and A. Fedosseev, Behavior of weld seams during cold rolling processes, *Proceedings of the 9th International and 4th European Steel Rolling Conference*, Paris, France, 2006.
28. J.S. Wanga, Z.Y. Jiang, A.K. Tieu, X.H. Liu, and G.D. Wang, A flying gauge change model in tandem cold strip mill, *Journal of Materials Processing Technology*, Vol. 204, No. 1–3, pp. 152–161, 2008.

29. J.R. Pittner and M.A. Simaan, An optimal control method for improvement in Tandem cold metal rolling, *IEEE IAS 2007 Conference Record of the 42nd Annual Meeting*, New Orleans, 2007.
30. J.R. Pittner and M.A. Simaan, Optimal control of Tandem cold rolling using a pointwise linear quadratic technique with trims, *ASME Transactions on Dynamic Systems, Measurement and Control*, Vol. 130, No. 3, 2008.
31. J.R. Pittner and M.A. Simaan, State-dependent Riccati equation approach for optimal control of a Tandem cold metal rolling process, *IEEE Transactions on Industry Application*, Vol. 42, No. 3, pp. 836–843, 2006.
32. T. Saito, T. Ohnishi, T. Komatsu, S. Miyoshi, H. Kitamura, and M. Kitahama, Automatic flatness control in Tandem cold rolling mill for ultra-thin gauge strip, *Kawasaki Steel Technical Report*, Vol. 24, pp. 41–46, 1991.
33. S.R. Duncan, J.M. Allwood, and S.S. Garimella, The analysis and design of spatial control systems in strip metal rolling, *IEEE Transaction on Control System Technology*, Vol. 6, No. 2, pp. 220–232, 1998.
34. J.V. Ringwood and M.J. Grimble, Shape control in Sendzimir mills using both crown and intermediate roll actuators, *IEEE Transactions on Automatic Control*, Vol. 35, No. 4, pp. 453–459, 1990.
35. M.J. Grimble and J. Fotakis, The design of strip shape control systems for Sendzimir mills, *IEEE Transactions on Automatic Control*, Vol. 27, No. 3, pp. 656–666, 1982.
36. R.M. Guo, Development of an optimal crown/shape level-2 control model for rolling mills with multiple control devices, *IEEE transactions on Control Systems Technology*, Vol. 6, No. 2, pp. 172–179, 1998.
37. G.R. Galvan, L.A. Garcia-Garza, and I. Peres-Vargas, Robustness margin of the hot-dip galvanising control system, *Proceedings of the 2003 IEEE Conference on Control Applications*, Instanbul, Turkey, 2003.
38. S.R. Yoo, I.S. Choi, P.K. Nam, J.K. Kim, S.J. Kim, and J. Davene, Coating deviation control in transverse direction for a continuous galvanizing line, *IEEE Transactions on Control Systems Technology*, Vol. 7, No. 1, pp. 129–135, 1999.
39. C. Schiefer, F.X. Rubenzucker, H.P. Jorgl, and H.R. Aberl, A neural network controls the galvannealing process, *IEEE Transactions on Industry Applications*, Vol. 35, No. 1, pp. 114–118, 1999.
40. H.L. Gerber, Magnetic damping of steel sheet, *IEEE Transactions on Industry Applications*, Vol. 39, No. 5, pp. 1448–1453, 2003.
41. J.G. Van Antwerp, A.P. Featherstone, R.D. Braatz, and B.A. Ogunnaike, Cross-directional control of sheet and film processes, *Automatica*, Vol. 43, pp. 191–211, 2007.
42. Y.T. Kim, An automatic coating weight control for continuous galvanizing line, *Proceedings of the Conference on Control, Automation and Systems*, Seoul, Korea, 2008.
43. C. Fenot, F. Rolland, G. Vigneron, and I.D. Landau, A successful black box design: digital regulation of deposited zinc in hot-dip galvanising at Sollac Florange, *Proceedings of the 2nd IEEE Conference on Control Applications*, Vancouver, Canada, 1993.
44. T. Watanabe, H. Narazaki, Y. Uchiyama, and H. Nakano, An adaptive fuzzy modeling for continuous galvanizing line, *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, 1997.
45. J.A. Thorton and H.F. Graff, An analytical description of the jet finishing process for hot dip metallic coatings on strip, *Metallurgical Transactions B*, Vol. 7, pp. 607–618, 1976.
46. C.H. Ellen and C.V. Tu, An analysis of jet stripping of liquid coatings, *ASME Journal of Fluids Engineering*, Vol. 106, pp. 399–404, 1984.
47. E.A. Elsaadawy, G.S. Hanumanth, A.K.S. Balthazaar, J.R. McDermid, A.N. Hrymak, and J.F. Forbes, Coating weight model for the continuous hot-dip galvanizing process, *Metallurgical and Materials Transactions B*, Vol. 38, No. 3, pp. 413–424, 2007.
48. C.H. Ellen and D.H. Wood, Wall pressure and shear stress measurements beneath an impinging jet, *Experimental Thermal and Fluid Science*, Vol. 13, No. 4, pp. 364–373, 1996.
49. V. Asano, K. Yamamoto, T. Kawase, and N. Nomura, Hot strip mill tension-looper control based on decentralisation and coordination, *Control Engineering in Practice*, Vol. 8, pp. 337–344, 2000.
50. F. Yamada, K. Sekiguchi, M. Tsugeno, Y. Anbe, Y. Andoh, C. Forse, M. Guernier, and T. Coleman, Hot strip mill mathematical models and set-up calculation, *IEEE Transactions on Industry Applications*, Vol. 27, No. 1, 131–139, 1991.
51. D.F. Garcia, J.M. Lopez, F.J. Suarez, J. Garcia, F. Obeso, and J.A. Gonzalez, A novel real-time fuzzy-based diagnostic system of roll eccentricity influence in finishing hot strip mills, *IEEE Transactions on Industry Applications*, Vol. 34, No. 6, 1342–1350, 1998.
52. G.W. Rigler, H.R. Aberl, W. Staufer, K. Aistleitner, and K.H. Weinberger, Improved rolling mill automation by means of advanced control techniques and dynamic simulation, *IEEE Transactions on Industry Applications*, Vol. 32, No. 3, pp. 599–607, 1996.
53. Y.-L. Hsu, C.-P. Liang, and S.-J. Tsai, An improvement of HAGC response for CSC No.1 HSM, *IEEE Transactions on Industry Applications*, Vol. 36, No. 3, pp. 854–860, 2000.

54. L.F. Lia, P. Caenenc, M. Daerdenc, D. Vaesc, G. Meersc, C. Dhondtd, and J.P. Celisa, Mechanism of single and multiple step pickling of 304 stainless steel in acid electrolytes, *Corrosion Science*, Vol. 47, No. 5, pp. 1307–1324, 2005.
55. R.Y. Chen and W.Y.D. Yuen, A study of the scale structure of hot-rolled steel strip by simulated coiling and cooling, *Oxidation of Metals*, Vol. 53, No. 5–6, pp. 539–560, 2000.
56. K.W. Gohring, N.D. Swain, and R.L. Sauder, Pickler line simulation model, *Proceedings of the 5th Annual Simulation Symposium*, Tampa, FL, 1972.
57. W.L. Roberts, *Cold Rolling of Steel*, Marcel Dekker, New York, 1978.
58. R.M. Guo, Optimal profile and shape control of flat sheet metal using multiple control devices, *IEEE Transactions on Industry Applications*, Vol. 32, No. 2, pp. 449–457, 1996.
59. G.C. Goodwin, S.J. Lee, A. Carlton, and G. Wallace, Application of Kalman filtering to zinc coating mass estimation, *Proceedings of the 3rd IEEE Conference on Control Applications*, Strathclyde University, Glasgow, UK, 1994.

IV

Biological and Medical

19

Model-Based Control of Biochemical Reactors

Michael A. Henson
University of Massachusetts Amherst

19.1 Introduction

The biotechnology industry is expanding rapidly due to continuing advances in the understanding of complex biological systems and the high demand for biologically manufactured products such as foods and beverages, pharmaceuticals, and commodity and specialty chemicals. The impact of the biotechnology industry on the global economy is substantial. For example, the revenues of the top 10 U.S. pharmaceutical manufacturers totaled USD 217 billion in 2002 with profits of USD 36 billion [1]. A rapidly growing biotechnology market is the large-scale production of ethanol as a renewable liquid fuel. The production of ethanol in 1998 was 31.2 billion liters worldwide and 6.4 billion liters in the United States with roughly two-thirds of the ethanol produced targeted for biofuel applications [2].

A typical biochemical manufacturing process consists of a reaction step in which a large number of cells is used to synthesize the desired product followed by a series of separation steps, in which the product is recovered from other constituents of the reaction liquid. The key requirement of the manufacturing process is the identification of a cell type that converts relatively inexpensive chemical species to the desired biochemical product. Advances in recombinant DNA technology facilitate the design of genetically engineered cell strains to enhance the yield of a target product [3]. While many industrial processes are based on microbial cells such as bacteria and yeasts, other cell types obtained from plants and animals are typically utilized to produce high-value pharmaceutical products such as therapeutic proteins [4].

Because each cell produces only a miniscule yield of a given product, a very large number of cells are needed to obtain commercially viable production rates. The cells are grown and products are harvested in large vessels known as biochemical reactors (bioreactors). The liquid removed from the *bioreactor* contains a mixture of biochemical species that must be separated to recover the desired *product*. The recovery step is usually achieved through a series of separation units [5]. The development of process control strategies for these separation systems is an important research problem not covered in this chapter.

19.2 Biochemical Reactor Technology

A schematic representation of a bioreactor in *continuous operating mode* is shown in Figure 19.1. The cells are innoculated into the bioreactor to initiate cell growth. Inoculation is achieved through a multistep procedure in which cells grown in a shake flask are transferred to increasingly larger bioreactors. This procedure is necessary to achieve a sufficiently large cell density ($\sim 10^{13}$ cells/L) to achieve rapid growth. The cells are continuously fed with a liquid medium stream containing chemicals that act as carbon, nitrogen, and phosphorous sources as well as other components including salts, minerals, and vitamins that replicate the natural growth environment. These chemicals are called nutrients or substrates. Careful preparation of the medium is essential since most cells are highly sensitive to changes in their growth environment. In *aerobic operation*, the cells utilize oxygen as a substrate, and air must be continuously supplied to the bioreactor to maintain the necessary dissolved oxygen concentration. By contrast, *anaerobic operation* does not require oxygen to achieve cell growth and product formation. Usually, the medium is prepared such that a single substrate such as glucose limits growth. This nutrient is known as the *growth limiting substrate*.

An agitator is used to continuously mix the liquid contents, thereby minimizing spatial gradients in substrate concentrations and cell density that can reduce bioreactor *productivity*. The agitator speed is chosen to provide adequate mixing while avoiding excessive shear forces that can rupture the cells. Figure 19.1 shows a stream being continuously removed from the reactor to achieve a constant liquid

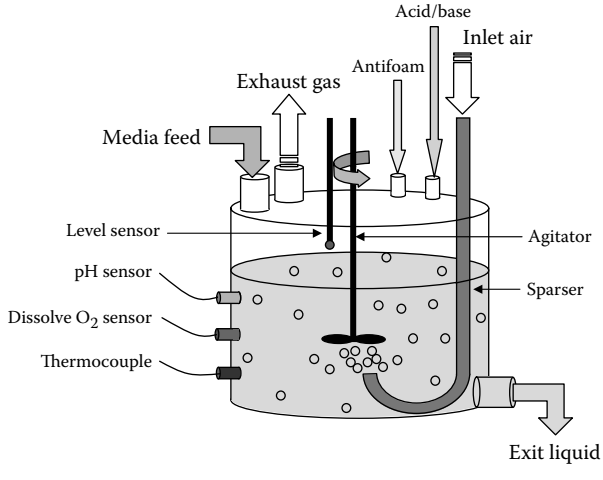

FIGURE 19.1 Continuous biochemical reactor (bioreactor) for aerobic manufacturing of biological products. A liquid media stream containing substrates and an air stream containing oxygen are continuously supplied to the bioreactor to sustain cell growth. The liquid level is maintained constant by continuously removing a stream containing unconsumed substrates, cellular biomass, and products of cellular metabolism. The level, temperature, pH, and oxygen content of the liquid are measured online and used as feedback signals for regulatory control.

volume, which is termed the *continuous operating mode*. The liquid removal rate is characterized by the *dilution rate*, which is the ratio of the volumetric feed flow rate to the liquid volume. The effluent stream contains unconsumed media components, cellular biomass, and products excreted by the cells. The desired product, which can be the cells themselves or a product of cell metabolism such as ethanol, is separated from the other components by a series of recovery and purification operations. Off-gases such as carbon dioxide are also generated as by-products of cell metabolism. Effective operation of an industrial bioreactor requires not only supplying the necessary nutrients and extracting the desired products but also maintaining sterility of the medium and processing equipment. A miniscule amount of microbial contamination can lead to production of the foreign microbe rather than the desired microbe, resulting in complete loss of productivity and an unscheduled shutdown of the bioreactor.

Many industrial bioreactors are operated in batch or *fed-batch operating mode* to allow more efficient media utilization and to avoid sterility problems caused by continuous liquid removal. In batch operation, the bioreactor is initially charged with cells and medium. The bioreactor then evolves to a predetermined final time with no media feed or liquid withdrawal. Fed-batch operation differs from batch operation in that fresh media feed is continuously supplied. Because there is no liquid withdrawal, the reactor volume increases until the final batch time. An advantage of fed-batch operation is that nutrient levels are continuously varied to achieve favorable growth conditions without significant risk of culture contamination.

A bioreactor system manufactured by New Brunswick Scientific is shown in Figure 19.2a. The 1500-L bioreactor constructed from a cylindrical stainless steel vessel is equipped with numerous stainless-steel tubes, valves, and electronic instruments to facilitate the manipulation of feed and withdrawal stream flow rates and continuous monitoring of growth conditions. The headplate located at the top of the bioreactor, and shown in Figure 19.2b, has openings to allow the insertion of tubes for feed and withdrawal streams and online probes for measuring temperature, pH, level, and oxygen concentration of the liquid mixture (see Figure 19.1).

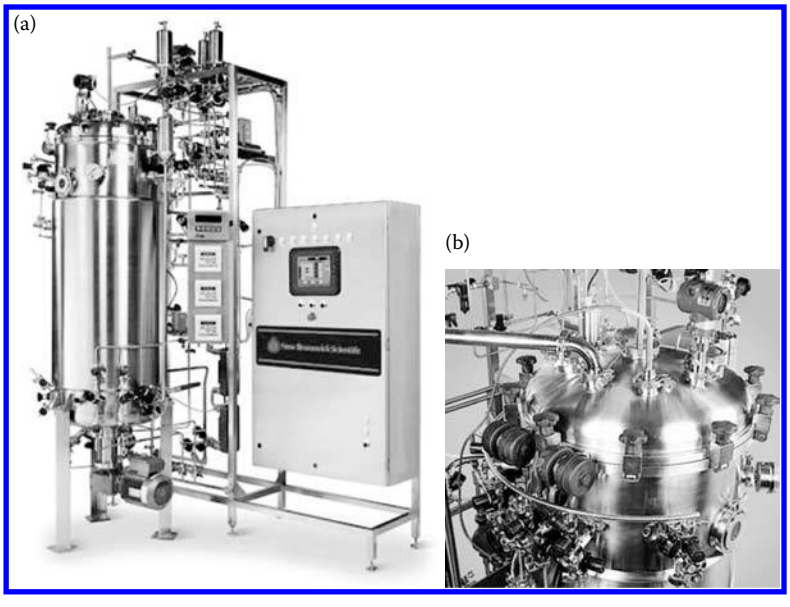

FIGURE 19.2 Industrial-scale bioreactor for manufacturing of biological products. The complete bioreactor system shown in (a) includes a 1500-L reaction vessel and stainless steel tubes, valves, and electronic instruments that allow flow rate manipulation and continuous monitoring of growth conditions. The bioreactor headplate in (b) provides openings for inserting feed/withdrawal tubes and submerged sensors for measuring properties of the reaction liquid. (Courtesy of New Brunswick Scientific.)

19.3 Bioreactor Monitoring and Control

Process control has played a more limited role in the biotechnology industry than in the petroleum and chemical industries. However, this situation is changing due to the expiration of pharmaceutical patents and the continuing development of global competition in biochemical manufacturing. In the United States, all aspects of pharmaceutical manufacturing processes are subject to validation mandated by the Food and Drug Administration. These requirements place stringent demands on the process control system to achieve reproducible operating conditions and consistent product quality. Process control is expected to be particularly important for producing commodity biochemicals such as ethanol that depend on economy of scale.

A unique feature of bioreactors is their unusually slow dynamics, which are characterized by the residence time (inverse of the dilution rate) for continuous operation. A typical dilution rate of 0.2 h^{-1} is equivalent to an open-loop time constant of 5 h. These very slow dynamics have important implications for control system design. Conventional bioreactor control systems are designed to supply the prescribed flow of nutrients while avoiding growth conditions that adversely affect productivity. Each cell type has a unique and narrow range of environmental conditions that supports cell growth. Most bioreactors use simple proportional-integral-derivative (PID) feedback control loops to maintain liquid temperature, pH, and oxygen concentration at predetermined setpoints. This simple regulatory structure is preferred due to the availability of cheap, accurate, and reliable sensors for these environmental variables [6] in contrast to physiological variables such as the growth rate, which provide a more direct measure of the cellular state. With regard to key output variables such as cell density and product concentration, this regulatory structure represents an open-loop control strategy that fails to account for cellular and media variations present in an industrial manufacturing environment.

An obstruction to process control has been the lack of online sensors that allow effective monitoring of the biochemical process state. Many physiological measurement techniques are limited to offline analysis in a research laboratory environment [7]. However, recent advances in online measurement technology have driven the development of model-based control strategies that offer the potential for improved bioreactor performance. For example, online spectrophotometers are now routinely used to measure the cellular biomass concentration [7]. Substrate and product concentrations in the liquid medium can be obtained from biochemical analyzers with automatic sampling systems as well as online gas chromatography and high-performance liquid chromatography [6,7]. More sophisticated measurement technologies that provide online measurements of intracellular species concentrations and heterogeneities across the cell population are under development [7].

19.4 Dynamic Modeling of Biochemical Reactors

Mathematical modeling of bioreactors is a challenging problem due to the complexity of cellular metabolism. The appropriate degree of model complexity is determined by factors such as the amount of fundamental knowledge, data requirements for model construction and validation, computational requirements, and the intended use of the model. Dynamic bioreactor models are classified according to the level of detail used to describe an individual cell. The most mechanistic descriptions of cellular metabolism are based on structured kinetic models, where the rates of individual enzyme-catalyzed reactions are embedded within dynamic mass balance equations for the intracellular species [8]. Due to the experimental difficulties associated with large-scale identification of enzyme kinetics, these ordinary differential equations models are effectively limited to primary metabolic pathways and are not well suited for capturing whole cell metabolism that impacts cellular growth and product synthesis rates. As a result, these models have not yet been used for bioreactor control.

Segregated models account for cell population heterogeneities by differentiating individual cells according to internal variables such as cellular mass or DNA content. While control strategies based on segregated models have been explored in simulation studies [9], the construction and validation of these partial differential equations models are difficult in practice. Due to their mathematical simplicity, dynamic models based on unstructured descriptions of cellular metabolism and unsegregated representations of the cell population are best suited for model-based controller design [10]. Rather than model individual enzyme-catalyzed reactions, lumped descriptions of cellular metabolism are employed. Cellular heterogeneities are ignored, and the model equations represent the dynamics of an "average" cell. The use of such *unstructured models* for bioreactor control is the focus of this chapter.

19.5 Continuous Operating Mode

A representative unstructured dynamic model for a continuous bioreactor consists of the following ordinary differential equations [11]:

$$\begin{aligned}\frac{dX}{dt} &= -DX + \mu(S,P)X,\\ \frac{dS}{dt} &= D(S_f - S) - \frac{\mu(S,P)}{Y_{xs}}X,\\ \frac{dP}{dt} &= -DP + \left[Y_{ps}\frac{\mu(S,P)}{Y_{xs}} + \frac{1}{Y_{xp}}\right]X,\end{aligned} \tag{19.1}$$

where X is the concentration of the cellular *biomass*, S is the concentration of the growth limiting substrate, P is the concentration of the desired product, S_f is the concentration of the growth-limiting substrate in the feed stream, and $D = F/V$ is the dilution rate, where F is the volumetric flow rate of the feed stream and V is the constant liquid volume in the bioreactor. Cellular growth is characterized by the *specific growth rate* function μ. The *yield parameter* Y_{xs} represents the cell mass produced from a unit mass of substrate. Similarly, the growth associated yield Y_{ps} is the mass of product produced from a unit mass of substrate and the nongrowth associated yield Y_{xp} is the mass of product produced per unit mass of biomass independent of growth. Although yield coefficients often vary with environmental conditions, they are usually treated as constants for simplicity.

The accuracy of the dynamic model depends strongly on identification of a growth rate function that adequately describes cellular growth over a range of environmental conditions. The function

$$\mu(S,P) = \frac{\mu_m(1 - P/P_m)S}{K_m + S + S^2/K_i}, \tag{19.2}$$

where μ_m is the maximum growth rate, K_m is the substrate saturation constant, K_i is the substrate inhibition constant, and P_m is the product inhibition constant, is sufficiently general to describe many situations of practical interest [12]. A simple saturation function is obtained when substrate and product inhibitory effects are negligible in the limit of large K_i and P_m. In this case, the growth rate increases monotonically with substrate concentration, and μ_m represents the maximum growth rate obtained in the limit of infinite substrate concentration. The more general expression $\mu(S,P)$ in Equation 19.2 is needed when high substrate and/or product concentrations inhibit cellular growth. For example, the product ethanol is known to inhibit yeast growth when its concentration is sufficiently large. Yield and specific growth rate parameters are available for common cell types grown under standard conditions. Otherwise, these parameters must be determined from experimental data using offline parameter estimation techniques [4].

The steady-state behavior of the continuous bioreactor model (Equation 19.1) is characterized by two types of equilibrium solutions. The first type corresponds to the undesirable trivial or *washout* solution

$$\bar{X} = 0, \quad \bar{S} = S_f, \quad \bar{P} = 0, \tag{19.3}$$

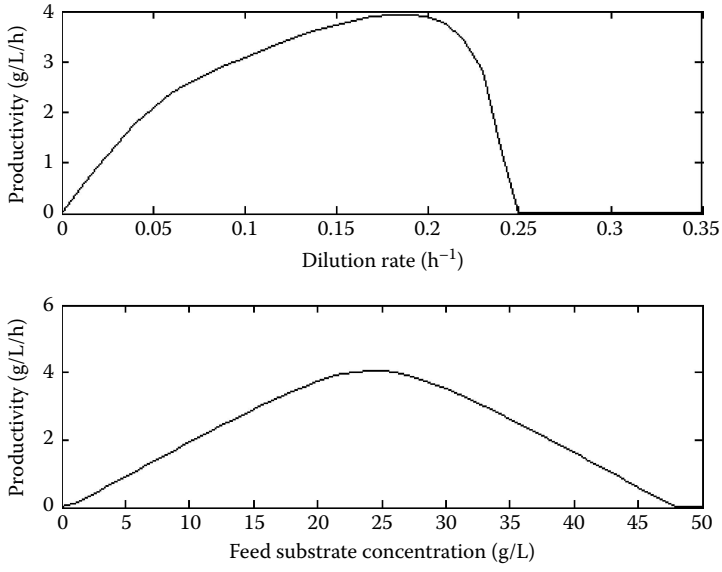

FIGURE 19.3 Predicted steady-state ethanol productivities as a function of the dilution rate D and feed glucose concentration S_f from the continuous yeast bioreactor model.

where the overbar denotes a steady-state solution. Washout occurs when the cellular growth rate is exceeded by the removal rate of cells from the bioreactor. In this case, cells eventually disappear from the reactor and none of the substrate entering the bioreactor is consumed. The number of nontrivial steady-state solutions for which the biomass and product concentrations are strictly positive depends on the specific growth rate function. In the absence of substrate and product inhibition, the growth rate function (Equation 19.2) yields a single nontrivial solution. When the dilution rate D is less than a critical dilution rate D_c that depends on model parameter values, the nontrivial steady-state solution is stable [13]. Otherwise, the washout steady-state is stable because the residence time is not sufficiently large to sustain cell growth. Consequently, there is a tradeoff between stability margin which requires small D and high-throughput which requires high D that must be considered during control system design.

Example 19.1: Anaerobic Yeast Growth in a Continuous Bioreactor

The dynamic models (Equations 19.1 and 19.2) have been used to describe the anaerobic growth of *yeast* cells on the limiting substrate glucose in a continuous bioreactor [12]. In this case, X is the yeast concentration, S is the glucose concentration, and P is the ethanol concentration. The function (Equation 19.2) describes the inhibitory effects of high glucose and ethanol concentrations on the cell growth rate and introduces steady-state input multiplicity in the model [12]. Bioreactor performance can be assessed in terms of the ethanol productivity $Q = DP$, which represents the mass of ethanol produced per unit volume of reactor liquid and per unit time. For a given set of model parameters [12] not reported here for the sake of brevity, input multiplicity is characterized by a maximum in the P versus D curve at which point the steady-state gain changes sign (Figure 19.3). The presence of a zero steady-state gain poses difficulties for control design when the objective is to stabilize the bioreactor near the point of maximum productivity by manipulating D. Similar behavior is observed with the feed substrate concentration S_f as the input variable.

The bioreactor models (Equations 19.1 and 19.2) also exhibit a significant degree of dynamic nonlinearity. Figure 19.4 shows the evolution of the ethanol concentration P from a common initial condition near the optimal productivity for different step changes in the dilution rate D and the feed glucose concentration S_f. The largest positive and negative D steps produce highly asymmetric

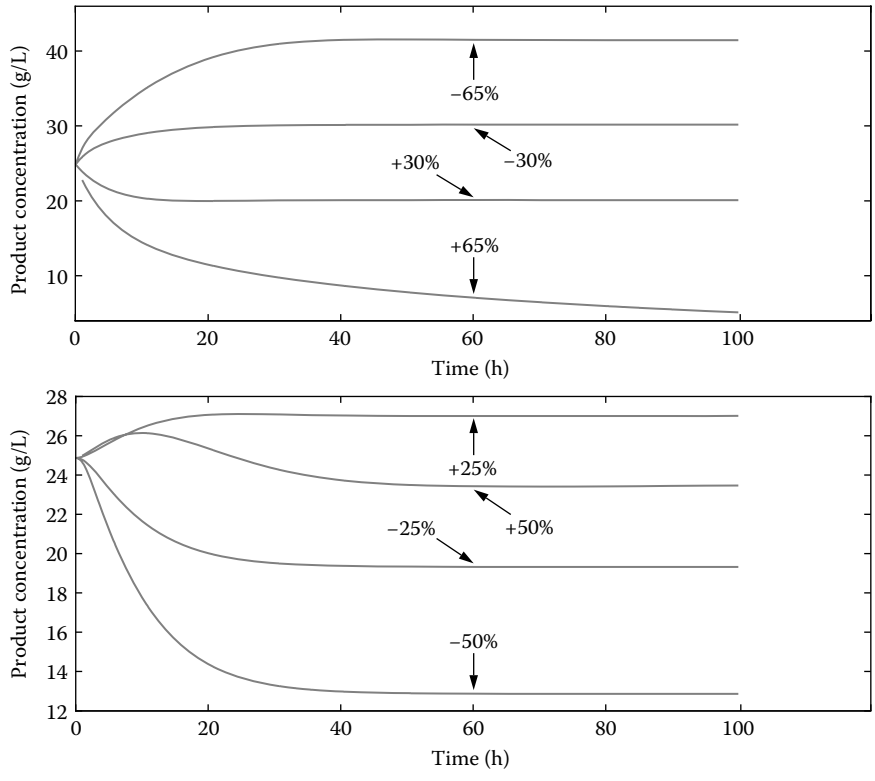

FIGURE 19.4 Dynamic simulations of the continuous yeast bioreactor model for symmetrical step changes in the dilution rate D (top) and feed glucose concentration S_f (bottom) from a common initial condition.

responses because the washout steady-state becomes stable for the positive change. For the smallest positive and negative S_f steps, significant differences in characteristic time constants and steady-state gains are observed. Even more pronounced asymmetries are evident for the largest S_f steps, as the positive change produces an inverse response due to the gain singularity at the maximum productivity. These strong nonlinearities must be considered in the controller design process.

19.6 Batch and Fed-Batch Operating Modes

The continuous bioreactor model (Equation 19.1) can be rewritten to describe batch or fed-batch operation by appropriate modification of the flow-dependent terms in the model equations [4]. Batch bioreactors are operated by initially charging the vessel with media and pregrown cells, allowing cell growth to proceed, and then removing the reaction liquid at a predetermined time to recover the desired product. Fed-batch bioreactors are operated similarly except that fresh media is supplied as cell growth progresses. The concept of a steady-state operating point is not meaningful for batch and fed-batch bioreactors owing to their inherently dynamic operation.

Example 19.2: Aerobic and Anaerobic Yeast Growth in a Fed-Batch Bioreactor

Fed-batch operation can be used to limit the inhibitory effects of high ethanol concentrations on yeast growth and enhance total ethanol production. In addition to glucose feeding throughout the

batch, a typical operating strategy involves aerobic growth during the initial phase of the batch to maximize biomass production followed by an anaerobic growth phase to maximize ethanol production. An unstructured dynamic model that describes both aerobic and anaerobic growth consists of the ordinary differential equations

$$\begin{aligned}
\frac{dV}{dt} &= F, \\
\frac{d(VX)}{dt} &= (\mu_f + \mu_o)VX, \\
\frac{d(VS)}{dt} &= FS_f - \left(\frac{\mu_f}{Y_{sf}} + \frac{\mu_o}{Y_{so}}\right)VX \\
\frac{d(VP)}{dt} &= Y_{ps}\left(\frac{\mu_f}{Y_{sf}}\right)VX,
\end{aligned} \tag{19.4}$$

where μ_f and μ_o are the growth rates for anaerobic (fermentative) and aerobic (oxidative) growth, respectively, Y_{sf} and Y_{so} are the corresponding substrate yield coefficients, and the nongrowth associated product yield $Y_{xp} = 0$. The two growth rate functions have the form

$$\begin{aligned}
\mu_f &= \mu_{fm}\frac{S}{K_{sf} + S + S^2/K_{isf}}\frac{1}{1 + P/K_{ipf}}\frac{1}{1 + O/K_{iof}}, \\
\mu_o &= \mu_{om}\frac{S}{K_{so} + S + S^2/K_{iso}}\frac{1}{1 + P/K_{ipo}}\frac{O}{K_O + O},
\end{aligned} \tag{19.5}$$

where O is the liquid oxygen concentration relative to the saturation value, with $O = 0\%$ corresponding to anaerobic conditions and $O = 100\%$ corresponding to fully aerobic conditions, and K_{ij} are constant model parameters. The oxygen concentration is assumed to remain constant despite the consumption of oxygen by cells through the action of an appropriately designed controller that manipulates air flow to the bioreactor. In addition to the usual substrate dependence, the anaerobic growth rate μ_f is inhibited by high ethanol concentrations and high oxygen concentrations since anaerobic growth occurs in the absence of oxygen. By contrast, the aerobic growth rate μ_o is an increasing function of oxygen since aerobic growth occurs only in the presence of oxygen.

The dynamic models (Equations 19.4 and 19 5) can be used to simulate both batch and fed-batch bioreactors through appropriate specification of the volumetric feed flow rate $F(t)$. In particular, batch operation is achieved by specifying $F(t) = 0$. For a given set of model parameters not reported here, for the sake of brevity, Figure 19.5 shows batch simulations for anaerobic and partially aerobic ($O =$ 50%) conditions. Anaerobic growth is characterized by slower glucose consumption, less biomass production, and more ethanol production than aerobic growth.

19.7 Process Control of Biochemical Reactors

Regardless of the bioreactor operating mode, the control objective is to maximize total production of the desired product. Most bioreactors are equipped with sensors for online measurement of liquid temperature, pH, level, and oxygen concentration. Simple PID regulatory loops are used to maintain the pH and temperature at constant setpoints predetermined to promote cell growth and product formation. The primary manipulated inputs available for higher-level controllers are the nutrient flow rates and concentrations. For the simple case of a single rate-limiting substrate, the available manipulated inputs are the feed flow rate F and the feed substrate concentration S_f.

The appropriate strategy for achieving the control objective depends strongly on the operating mode, the availability of online measurements, and the accuracy of the dynamic bioreactor model. The single most important determinant is the operating mode since continuous operation involves regulation at an equilibrium point whereas fed-batch operation requires tracking dynamic trajectories. The continuous

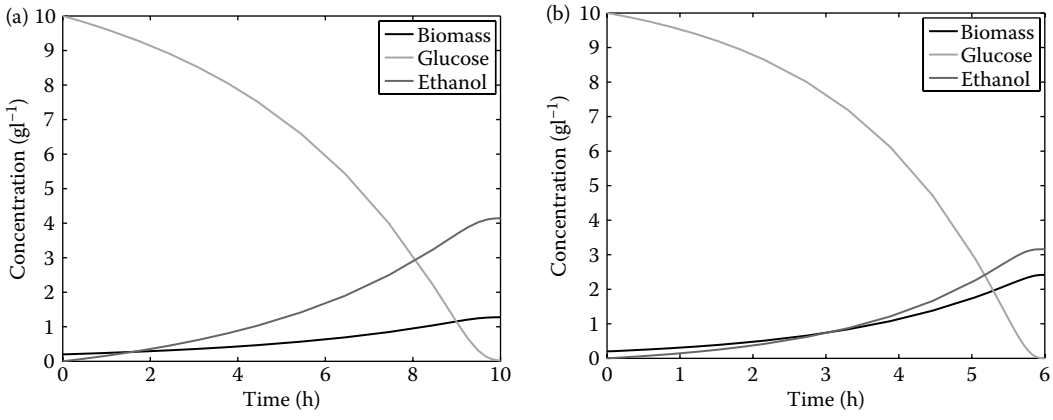

FIGURE 19.5 Dynamic simulations of the batch yeast bioreactor model for anaerobic conditions (a) and partially aerobic ($O = 50\%$) conditions (b).

and fed-batch control problems are considered separately below due to their fundamentally different nature. Batch bioreactors are not discussed further since their lack of feed and withdrawal streams does not allow feedback control during the batch. Instead, run-to-run control strategies have been successfully applied to batch operation [14].

19.8 Continuous Biochemical Reactors

A typical control objective is regulation at an operating point that maximizes the steady-state productivity. Determination of an appropriate operating point is nontrivial due to the complex effects of the environment on cellular metabolism that may result in highly nonlinear behavior (see Figure 19.3). Common industrial practice is to determine the operating point through a time-consuming and expensive experimental design procedure [15]. When a sufficiently accurate bioreactor model is available, the optimal operating point can be determined offline using simple optimization techniques [12]. Online optimization strategies based on adaptive extremum seeking control can address the significant errors present in many unstructured bioreactor models [16].

Once the desired operating point is determined, the next step is to design a feedback controller that achieves regulation despite unmeasured disturbances that arise from sources such as nutrient variations, inadequate liquid mixing, imperfect pH and temperature control, and unmodeled cellular behavior. Both simple PID and model-based control strategies have been used for this purpose. Controller design is heavily influenced by the online measurements available as feedback signals. Due to their high reliability and relatively low cost, analyzers that provide liquid oxygen concentration and carbon dioxide gas concentration measurements are commonly used for control system design [17]. The main limitation of this approach is that these concentration measurements provide indirect measures of the cellular state, and thus regulation of these measurements at predetermined steady-state values cannot be expected to result in optimal productivity.

The availability of online analyzers that provide direct measurements of substrate and product concentrations has enabled the development of more effective bioreactor control strategies. Commercial instruments can provide these concentration measurements every few minutes, while the time constant of a typical continuous bioreactor is several hours. The rapid analyzer sampling rate allows the controller design problem to be treated in continuous time. Both nonadaptive and adaptive nonlinear control strategies have received attention due to highly nonlinear and uncertain bioreactor dynamics [17].

Continuous bioreactor models are well suited for applying nonlinear controller design methods based on differential geometry [18], and both state-space linearization [19] and *input–output linearization* [20] techniques have been investigated. Because offset-free regulation of the controlled output at a specified setpoint is required, input–output linearization is usually the preferred method [12].

Feedback linearizing control strategies suffer from practical limitations such as the need for accurate descriptions of cellular growth and product yields as well as the limited availability of online measurements of the biomass, substrate, and product concentrations. While reasonably accurate values of yield coefficients can often be obtained, the determination of a rate function that captures cellular growth over a wide range of bioreactor operating conditions is notoriously difficult. Adaptive versions of input–output linearization in which the growth rate μ is treated as an unknown, time-varying parameter are experimentally evaluated in [11,21]. Although model-based controller implementation is facilitated by recent advances in biochemical measurement technology, industrial manufacturing processes often lack online biomass, substrate, and product concentration measurements. In this case, simple nonlinear state estimators [22,23] can be combined with adaptive input–output linearizing controllers to yield satisfactory closed-loop performance [21].

Example 19.3: Input−Output Linearizing Control of a Continuous Yeast Bioreactor

The dynamic models (Equations 19.1 and 19.2) have been used to design feedback linearizing controllers for stabilization of predetermined steady-state operating points and tracking of ethanol productivity setpoints [12]. Here the linearizing controller design procedure is outlined when the dilution rate D is the manipulated input and the ethanol productivity $Q = DP$ is the controlled output. Because the output depends explicitly on the input, this system is relative degree zero and controller design proceeds as follows [12]:

$$Q = v = \frac{1}{\varepsilon}\int_0^t [r(\tau) - Q(\tau)]\, d\tau \;\Rightarrow\; DP = \frac{1}{\varepsilon}\int_0^t [r(\tau) - D(\tau)P(\tau)]\, d\tau, \tag{19.6}$$

where $v(t)$ is the manipulated input of the feedback linearized system, $r(t)$ is the productivity setpoint, and ε is a controller tuning parameter. Under the assumption that $Q(0) = r(0)$, the control law (Equation 19.6) yields the following closed-loop transfer function:

$$\frac{Q(s)}{r(s)} = \frac{1}{\varepsilon s + 1}. \tag{19.7}$$

The control law (Equation 19.6) is an implicit function of the manipulated input D. An explicit control law can be obtained by differentiating Equation 19.6 with respect to time to yield

$$\frac{dD}{dt} = \frac{1}{P}\left[\left\{DP - \left(\frac{Y_{ps}\mu}{Y_{xs}} + \frac{1}{Y_{xp}}\right)\right\} D + \frac{1}{\varepsilon}(r - DP)\right]. \tag{19.8}$$

Figure 19.6 shows setpoint tracking performance for an initial condition near the optimal productivity. The controller provides symmetrical productivity responses for small changes ($\pm$5%) and a linear response for the large negative change ($-$10%). However, the controller encounters a singularity for the large positive change ($+$10%) due to the process gain sign change at the maximum productivity. As a result, the dilution rate becomes unbounded and the control system becomes unstable. Figure 19.7 shows regulatory performance for step disturbance changes in the feed substrate concentration at time zero. Because this disturbance has relative degree 2, the controller provided perfect regulatory performance for the positive change ($+$25%) and yields stable closed-loop performance (Figure 19.8). However, the negative change ($-$25%) causes the gain singularity to be encountered and the control system becomes unstable (Figure 19.8) due to the dilution rate becoming unbounded. A simple approach for overcoming this singularity problem has been proposed [12].

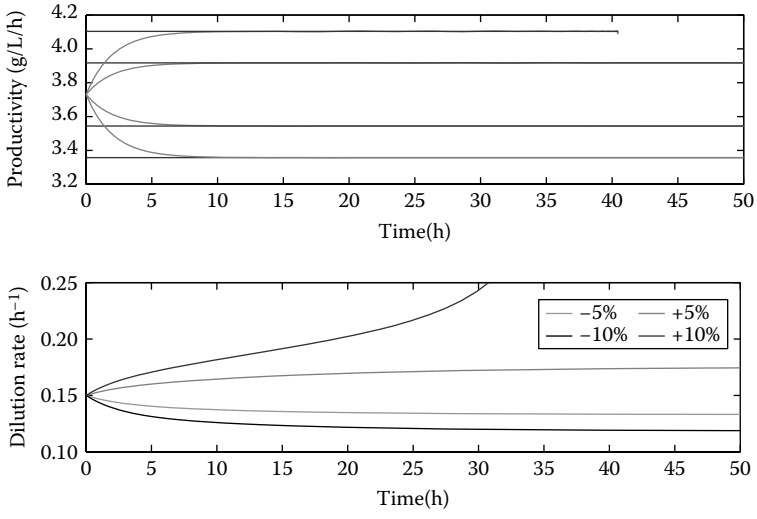

FIGURE 19.6 Setpoint tracking performance for input–output linearizing control of the continuous yeast bioreactor model. The manipulated input is the dilution rate D and the controlled output is the ethanol productivity Q. The productivity setpoint r is changed $\pm 5\%$ and $\pm 10\%$ from its nominal value at time zero.

19.9 Fed-Batch Biochemical Reactors

Because they are operated dynamically over a finite batch time, fed-batch bioreactors offer a unique set of challenges for control system design. Rather than stabilize a fixed operating point, the control objective is to maximize the amount of product at the final batch time. The productivity depends on the initial batch

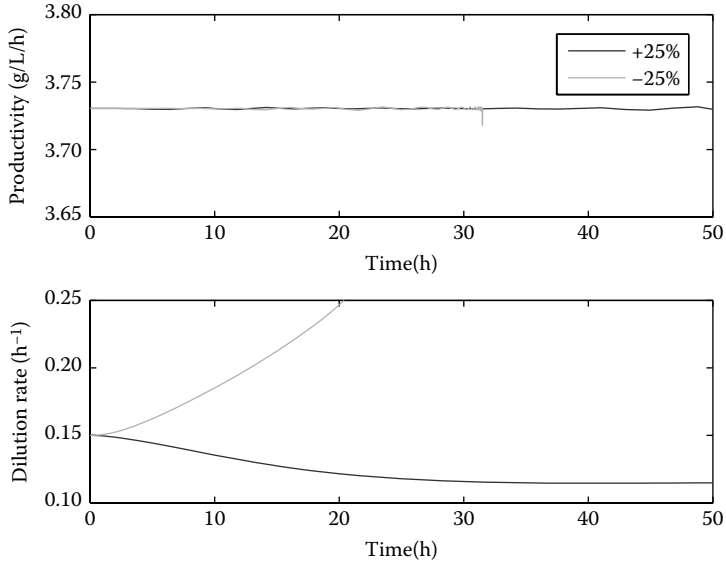

FIGURE 19.7 Disturbance rejection performance for input–output linearizing control of the continuous yeast bioreactor model. The manipulated input is the dilution rate D and the controlled output is the ethanol productivity Q. The feed substrate concentration S_f is changed $\pm 25\%$ from its nominal value at time zero.

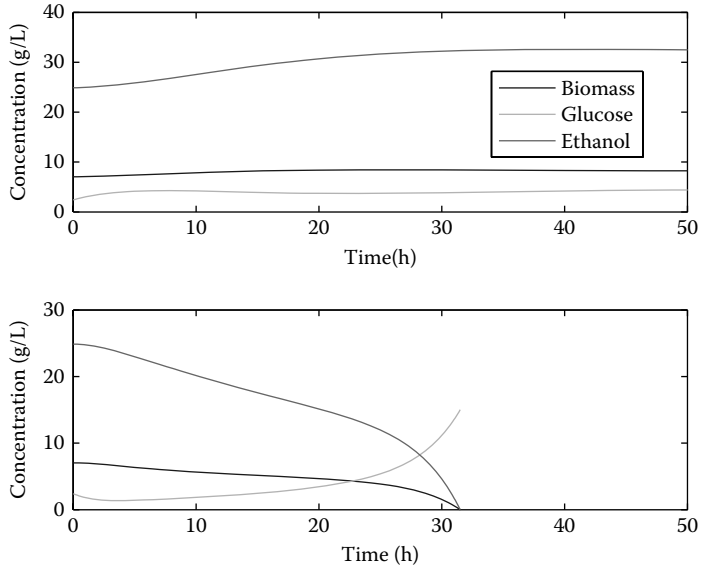

FIGURE 19.8 Closed-loop trajectories for input–output linearizing control of the continuous yeast bioreactor model when the feed substrate concentration S_f is changed +25% (top) and −25% (bottom) from its nominal value at time zero.

conditions, the substrate feeding policy, and the batch duration. A simple class of fed-batch bioreactor control strategies is based on regulating a substrate or product concentration at a predetermined setpoint that maximizes the predicted cellular growth rate [23,24].

Computational methods are required to rigorously determine optimal fed-batch control policies because the effects of manipulated variables on cellular growth and product formation are complex. The open-loop optimal policy can be determined by solving an optimal control problem [10,17]. Typically, an objective function representing the total mass of the desired product at the final batch time is maximized subject to constraints imposed by the dynamic model equations and operational limitations. Various computational algorithms have been used for dynamic optimization. Sequential solution methods involve iteration between a dynamic simulation code that integrates the bioreactor model equations given a candidate feeding policy and a nonlinear programming code that processes the dynamic simulation results to determine an improved feeding policy. While they are relatively straightforward to develop, sequential solution methods exhibit slow convergence and occasional failure for large optimization problems.

Simultaneous solution methods in which model integration and operating policy optimization are embedded within a single computational algorithm provide more efficient and robust problem solution [25]. A difficulty is that most nonlinear optimization codes cannot accommodate differential equation constraints. Simultaneous solution methods based on temporal discretization of the dynamic model equations are effective due to their ability to explicitly account for state-dependent constraints and their applicability to large optimal control problems. The dynamic optimization approach has been applied to simulated fed-batch bioreactors [16,26] as well as to experimental systems [27]. A representative problem is the maximization of protein production by manipulating substrate feed flow rates [28].

Numerical solution of the *fed-batch optimization* problem yields an open-loop control policy designed to maximize productivity. In practice, direct implementation of the open-loop policy yields suboptimal performance due to the presence of structural modeling errors and unanticipated disturbances during the batch. A standard approach for handling unmodeled dynamics is to combine a feedback controller with an online state estimator to correct the dynamic model predictions when measurement information becomes available. A unique feature of fed-batch operation is the presence of a final batch time such that

the time horizon for estimation and control becomes shorter as the batch proceeds. Extensions of model predictive control based on the concept of a shrinking horizon address this class of problems by assuming predetermined initial batch conditions and a fixed final batch time. The shrinking horizon control problem is solved from the current time instant to the final batch time by using the most current state estimate to reset the initial conditions of the bioreactor model. To compensate for modeling errors and disturbances, only the first set of calculated substrate feed changes is implemented. Then the optimization problem is resolved at the next time instant over a shorter horizon using the new state estimate. Applications of shrinking horizon control to fed-batch bioreactors have been presented [29–30], and more simulation and experimental studies are expected in the future.

Example 19.4: Open-Loop Optimization of a Fed-Batch Yeast Bioreactor

The dynamic models (Equations 19.4 and 19.5) of combined aerobic and anaerobic yeast growth in a fed-batch bioreactor were used to compute an optimal open-loop operating policy. The initial volume $V(0)$ and glucose concentration $S(0)$, the feed flow rate $F(t)$ and oxygen concentration $O(t)$, and the final batch time t_f were manipulated to maximize ethanol productivity. Upper and lower variable bounds were specified to ensure a physically realistic solution. The bioreactor model equations were discretized in time using Radau collocation on finite elements. The resulting optimization problem was solved through the AMPL interface to the nonlinear program solver CONOPT.

Figure 19.9 shows the glucose and oxygen feeding profiles as well as the state profiles obtained from implementation of the optimal control policy. Initially, the policy produced no glucose feeding and partially aerobic conditions that balanced the cellular growth and ethanol production rates (Figure 19.9, top). The glucose concentration declined until glucose feeding commenced, at which time the glucose concentration remained constant (Figure 19.9, bottom) at a value that maximized the combined aerobic and anaerobic growth rates. The oxygen concentration decreased throughout the batch until completely anaerobic conditions were obtained, representing a switch from biomass production to ethanol production. Glucose feeding was stopped near the final batch time to satisfy the constraint on the maximum liquid volume.

19.10 Perspective

Process control is expected to play an increasingly important role in the biotechnology industry. The development of feedback control systems that exploit advances in online measurement technology to achieve optimal productivity of continuous and fed-batch bioreactors is one of the most important challenges in biochemical manufacturing. This chapter provided an overview of current bioreactor control strategies based on unstructured dynamic models of cell growth and product formation. Despite their widespread acceptance, these models suffer from several fundamental limitations, most notably lumped descriptions of cellular metabolism and the assumption of cellular homogeneity. In addition to extending the applicability of existing methods, future work will focus on the utilization of more detailed dynamic models for bioreactor optimization and control.

Steady-state mass balance equations can be used to describe intracellular reaction pathways under the realistic assumption that intracellular dynamics are much faster than extracellular dynamics. The steady-state intracellular model can be combined with transient mass balance equations for key extracellular substrates and products to generate dynamic predictions of the growth rate and product yields [31]. Such models are currently being utilized to develop optimization and control strategies for fed-batch bioreactors [32]. The recent development of online flow cytometry allows heterogeneities across a cell population to be quantified in real time [33]. Dynamic measurements of DNA and protein content distributions can be used as feedback signals for nonlinear controllers that provide direct regulation of cell population properties [9].

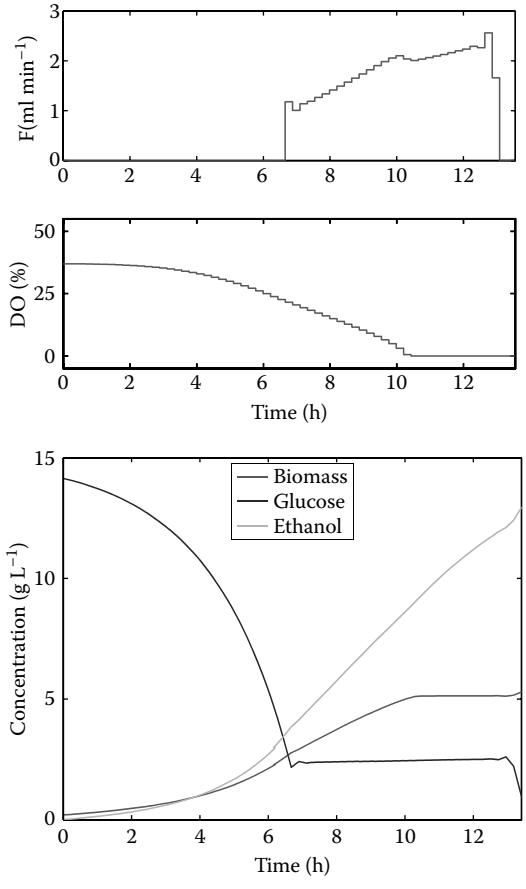

FIGURE 19.9 Open-loop optimization of ethanol productivity in the fed-batch yeast bioreactor model. Optimal glucose and oxygen feeding profiles (top) and glucose, biomass, and ethanol concentration profiles resulting from the optimal policy (bottom).

19.11 Defining Terms

Aerobic operation: Operating mode in which oxygen is continuously supplied to the bioreactor.

Anaerobic operation: Operating mode in which oxygen is not supplied to the bioreactor.

Biomass: The total mass of all cells.

Bioreactor: A vessel in which cells are grown in a controlled environment.

Continuous operating mode: Operating mode in which a liquid stream containing substrates is continuously fed and a liquid stream containing cells, unconsumed substrates, and products is continuously removed at the same rate such that the liquid level in the bioreactor remains constant.

Dilution rate: Ratio of the liquid volume in the bioreactor and the volumetric feed flow rate. This is equal to the inverse of the bioreactor residence time, which is the dominant process time constant.

Fed-batch operating mode: Operating mode in which a liquid stream containing substrates is continuously fed to the bioreactor but no liquid stream is removed from the bioreactor. This mode is characterized by a finite operating time and no steady states.

Fed-batch optimization: The determination of a fed-batch operating policy, most notably the substrate feeding profiles, that optimizes a bioreactor performance measure, such as the productivity.
Input–output linearization: A nonlinear feedback controller design method based on establishing a linear closed-loop relationship between the controlled output and its setpoint.
Product: A biochemical species produced by cells in a bioreactor.
Productivity: The total mass of the desired product produced per unit volume and per unit time.
Specific growth rate: The rate at which biomass increases with time: Typically a function of substrate concentration and possibly product concentrations.
Substrate: A biochemical species necessary for cell growth that is fed to a bioreactor. The growth limiting substrate is the single substrate not supplied in excess.
Unstructured model: A dynamic bioreactor model based on a lumped description of cellular metabolism.
Washout: An undesirable steady state of a continuous bioreactor in which no cells are produced.
Yeast: A type of microbial cell that is commonly used industrially, especially for ethanol production.

19.12 For Further Information

The basics of bioreactor modeling and control are covered in introductory textbooks on biochemical engineering [3] and review articles [18]. The development of more advanced modeling techniques than the unstructured modeling approach described in this chapter is an active area of research [3,8,9,13,31]. The application of advanced model-based control techniques, including feedback linearization [12,14,19,20] and nonlinear adaptive control [11,16,21,23], to bioreactors has received considerable attention. More detailed descriptions and alternative applications of fed-batch bioreactor optimization are available [26–28,32].

Acknowledgment

The efforts of Jared L. Hjersted in generating the fed-batch simulation and optimization results in Figures 19.5 and 19.9 are gratefully acknowledged.

References

1. A. Harrington, Honey, I shrunk the profits, *Fortune*, vol. 147, no. 7, pp. 197–199, 2003.
2. C. Berg, World ethanol production and trade to 2000 and beyond, January 1999. Available at www.distill.com/berg.
3. G.N. Stephanopoulos, A.A. Aristidou, and J. Nielsen, *Metabolic Engineering: Principles and Methodologies*, New York, NY: Academic Press, 1998.
4. M.L. Shuler and F. Kargi, *Bioprocess Engineering: Basic Concepts*, 2nd ed., Upper Saddle River, NJ: Prentice-Hall, 2002.
5. M. Kalyanpur, Downstream processing in the biotechnology industry, *Mol Biotechnol*., vol. 22, no. 1, pp. 87–98, 2002.
6. B. Sonnleitner, Instrumentation of biotechnological processes, *Adv. Biochem. Eng. Biotechnol*., vol. 66, pp. 1–64, 2000.
7. K.C. Schuster, Monitoring the physiological status in bioprocesses at the cellular level, *Adv. Biochem. Eng. Biotechnol.,* vol. 66, pp. 185–208, 2000.
8. A.K. Gombert and J. Nielsen, Mathematical modeling of metabolism, *Curr. Opinion Biotechnol*., vol. 11, no. 2, pp. 180–186, 2000.
9. M.A. Henson, Dynamic modeling of microbial cell populations, *Curr. Opinion Biotechnol*., vol. 14, no. 5, pp. 460–467, 2003.
10. A. Lubbert and S.B. Jorgensen, Bioreactor performance: A more scientific approach for practice, *J. Biotechnol*., vol. 85, no. 2, pp. 187–212, 2001.

11. G. Bastin and D. Dochain, *On-Line Estimation and Adaptive Control of Bioreactors*, Amsterdam: Elsevier, 1990.
12. M.A. Henson and D.E. Seborg, Nonlinear control strategies for continuous fermentors, *Chem. Eng. Sci.*, vol. 47, no. 4, pp. 821–835, 1992.
13. J. Nielsen and J. Villadsen, *Bioreaction Engineering Principles*, New York, NY: Plenum Press, 1994.
14. D. Bonvin, B. Srinivasan, and D. Hunkeler, Batch process control, *Control Systems Magazine*, vol. 26, pp. 54–62, 2006.
15. S. Parekh, V.A. Vinci, and R.J. Strobel, Improvement of microbial strains and fermentation processes, *Appl. Microbiol. Biotechnol.*, vol. 54, no. 3, pp. 287–301, 2000.
16. T. Zhang, M. Guay, and D. Dochain, Adaptive extremum seeking control of continuous stirred-tank reactors, *AIChE J.*, vol. 49, no. 1, pp. 113–123, 2004.
17. K.Y. Rani and V.S.R. Rao, Control of fermenters: A review, *Bioprocess Eng.*, vol. 21, no. 1, pp. 77–88, 1999.
18. A. Isidori, *Nonlinear Control Systems II*, New York, NY: Springer, 1999.
19. T. Proll and N. M. Karim, Nonlinear control of a bioreactor model using exact and I/O linearization, *Int. J. Control*, vol. 60, no.4, pp. 499–519, 1994.
20. J. el Moubaraki, G. Bastin, and J. Levine, Nonlinear control of biotechnological processes with growth-production decoupling, *Math. Biosci.*, vol. 116, no. 1, pp. 21–44, 1993.
21. D. Dochain and M. Perrier, Dynamical modeling, analysis, monitoring and control design for nonlinear bioprocesses, *Adv. Biochem. Eng. Biotechnol.*, vol. 56, pp. 147–197, 1997.
22. M. Farza, M. Nadri, and H. Hammouri, Nonlinear observation of specific growth rate in aerobic fermentation, *Bioprocess. Biosystem Eng.*, vol. 23, no. 4, pp. 359–366, 2000.
23. I.Y. Smets, J.E. Claes, E.J. November, G.P. Bastin, and J.F. van Impe, Optimal adaptive control of (bio)chemical reactors: Past, present and future, *J. Process Control*, vol. 14, no. 7, pp. 795–805, 2004.
24. C. Cannizzaro, S. Valentinotti, and U. von Stockar, Control of yeast fed-batch process through regulation of extracellular ethanol concentration, *Bioprocess. Biosystem Eng.*, vol. 26, no. 6, pp. 377–383, 2004.
25. L.T. Biegler, A.M. Cervantes, and A. Wachter, Advances in simultaneous strategies for dynamic process optimization, *Chem. Eng. Sci.*, vol. 57, no. 4, pp. 575–593, 2002.
26. J.R. Banga, E. Balsa-Canto, C.G. Moles, and A.A. Alonso, Dynamic optimization of bioprocesses: Efficient and robust numerical methods, *J. Biotechnol.*, vol. 117, no. 4, pp. 407–419, 2005.
27. G. Liden, Understanding the bioreactor, *Bioprocess. Biosystem Eng.*, vol. 24, no. 5, pp. 273–279, 2002.
28. D. Levisauskas, V. Galvanauskas, S. Heinrich, K. Wilhelm, N. Volk, and A. Lubbert, Model-based optimization of viral capsid protein production in fed-batch culture of recombinant *Escherichia coli*, *Bioprocess. Biosystem Eng.*, vol. 25, no. 4, pp. 255–262, 2003.
29. B. Frahm, P. Lane, H. Atzert, A. Munack, M. Hoffmann, V.C. Hass, and R. Portner, Adaptive, model-based control by the open-loop-feedback-optimal (OLFO) controller for the effective fed-batch cultivation of hybridoma cells, *Biotechnol. Prog.*, vol. 18, no. 5, pp. 1095–1103, 2002.
30. R. Mahadevan and F.J. Doyle III, On-line optimization of recombinant protein in fed-batch bioreactor, *Biotechnol. Prog.*, vol. 19, no. 2, pp. 639–646, 2003.
31. R. Mahadevan, J.S. Edwards, and F. J. Doyle III, Dynamic flux balance analysis of diauxic growth in *Escherichia coli*, *Biophys. J.*, vol. 83, no. 3, pp. 1331–1340, 2002.
32. J. Hjersted and M. A. Henson, Optimization of fed-batch yeast fermentation using dynamic flux balance models, *Biotechnology Progress*, vol. 22, pp. 1239–1248, 2006.
33. N. R. Abu-Absi, A. Zamamiri, J. Kacmar, S. J. Balogh, and F. Srienc, Automated flow cytometry for acquisition of time-dependent population data, *Cytometry Pt. A*, vol. 51A, no. 2, pp. 87–96, 2003.

20

Robotic Surgery

Rajesh Kumar
Johns Hopkins University

20.1 Introduction

Robots are now widely employed in surgical interventions. While robotic assistance has been investigated in nearly all forms of surgery, complex procedures, in particular those utilizing minimally invasive surgical techniques have seen the widest adoption [1]. Robotic interventions primarily strive to reduce or eliminate human limitations. These limitations include tremor, fatigue, variability, and inability to accurately visualize or target disease discovered in volumetric imaging [computed tomography (CT) and magnetic resonance imaging (MRI) scans] with high accuracy and safety. Robotic devices also often provide improved access, dexterity, and precision.

A surgical robotic system includes robotic manipulation of surgical instruments for improved precision and dexterity, sensing and visualization devices for imaging and targeting, and computing engines for planning, controlling, and monitoring the procedure. Early robotic surgery applications used modified industrial manipulators, such as the modified SCARA manipulator used in the ROBODOC orthopedic system ([2], Integrated Surgical Systems, now Curexo Technology Corp.). The ROBODOC robot drives a pneumatic drill to mill a hip implant cavity in the femur more accurately than manual reaming. Other early medical robots of note include the Computer Motion AESOP laparoscopic camera holder, another modified SCARA design, that aims to replace human camera holding assistants.

With the availability of improved computing and robotics, more advanced robotic systems and a wider range of surgical applications have appeared. As in the evolution of industrial manipulators, teleoperated robotic systems (e.g., da Vinci Robotic Surgery system [3]) remain the most widely used devices in the current generation, although more automated applications [4] are also becoming common. However, unlike industrial applications, surgical robots are required to satisfy much more stringent and application-specific safety, accuracy, and reliability requirements before they can be used on humans.

In addition to orthopedic applications, robotic minimally invasive interventions are being adopted for a range of surgical applications due to patient benefits such as improved recovery times, smaller incisions and reduced blood loss, and potentially better surgical outcomes. Urological, gynecological, and

some cardiac procedures have seen wide adoption of telerobotic surgery. Other common interventions include localized and external beam (e.g., the Accuray GammaKnife system) and localized radiotherapy and other needle-based therapy (e.g., biopsies, ablation and brachytherapy). Natural orifice transluminal endoscopic surgery (NOTES), and similar new minimally invasive techniques are leading to development and adoption of more sophisticated robotic devices. Robots are now also being integrated in nonsurgical medicinal fields such as rehabilitation and assistive technologies. While visual imaging (cameras and endoscopes) remain the most common guidance modality, a range of alternative modalities, for example, ultrasound, fluoroscopy, CT, and MRI scans are located to find the targets for robotic treatment.

In this chapter, we present an introduction to the control aspects of the robotic manipulation used in these applications. This material is not intended to be a survey, and should be accessible to a reader with knowledge of basic robotics. We also do not detail other aspects of *robotic surgery* or the broader *computer-assisted surgery* area. For broader reference, the reader is referred to a large set of review articles and edited texts discussing specific aspects of robotic surgery. Edited texts, including Taylor et al. [5], Faust [6], and Peters et al. [7] provide overview of research in specific areas of the wider computer-assisted surgery or computer-assisted intervention field. The recent *IEEE Robotics and Automation* three part tutorial [4,8,9], and references contained within, also provides a comprehensive overview of all aspects of computer-assisted surgery. For a clinical perspective of robotic devices, see Patel et al. [1], and references contained within.

20.2 Robotic Surgery Systems

Robot mechanism design for surgery is guided by application specific requirements such as accuracy (e.g., microsurgery, or laparoscopy), level of autonomy (computer controlled, teleoperated or directly operated), the type of intervention (e.g., minimally invasive or external therapy), and the type of imaging and sensing employed (MRI or stereo visual endoscopy). Other design considerations include the type, number, dexterity, weight, and actuation of surgical instruments to be carried, as well as stiffness, and transparency of operation. In addition, clinical requirements such as fault tolerance, sterility, and reliability also apply. Readers interested in more detailed mechanism design and additional specific systems are referred to reviews such as Taylor et al. [10] or Camarillo et al. [11] and references contained within.

From a control perspective, most surgical robots can be broadly classified in three categories. The first category of *computer-controlled manipulators* form a part of a surgical intervention suite (also referred to as a surgical CAD/CAM system) that aims to improve access and accuracy. The second category of *telemanipulated manipulators* provide minimally invasive access and improved dexterity, while directly controlled *cooperative manipulators* preserve kinesthetics while reducing or filtering tremor. There are additional classifications within these categories, for example, teleoperation may be classified into direct control, shared control, and supervisory control [12]. Here, we limit ourselves to the broad classification.

20.2.1 Computer Control and CAD/CAM

Targets in bony anatomy can be immobilized, and are easily imaged using commonly available CT and fluoroscopy imaging. Imaging is already used to plan such surgeries even where robotics is not used, and surgical goals for robotic applications are also relatively easily defined in robotic terms. Common surgical tasks in these procedures involve shaping or cutting bones to provide access to the spine or nerves, creating cavities for placement of screws or implants, or shaping bones for reconstruction. Orthopedic surgery is likely the most explored surgical CAD/CAM specialty. Accuracy goals for these tasks emphasize geometric accuracy to prevent damage to nearby nerves or critical structures, or to ensure a proper fit.

The ROBODOC system (Figure 20.1, right) was the first orthopedic robot initially designed for automating a part of a hip replacement surgery [2,8] procedure. Applications have since been extended to include other joint reconstruction. ROBODOC surgery planning system (called the ORTHODOC

FIGURE 20.1 The NEUROMATE neurosurgical robot (left) and the ROBODOC hip replacement system in clinical use (right). (Courtesy of Peter Kazanzides).

workstation) allows a surgeon to graphically position a CAD model of an appropriately sized hip implant with respect to the patient's CT scan. During surgery, this treatment plan is executed by registering the robot coordinate system to the CT coordinate system. Early versions of the software used pins implanted in the bone prior to the CT scan for registration, later versions replaced this with registration using anatomical surfaces. After registration, the robot autonomously machines the bone using a high-speed pneumatic mill to create an accurate cavity in the shape of the desired implant. The remaining portion of the surgery then proceeds as in the nonrobotic technique. Other contemporary robotic systems for orthopedic surgery include the German CASPER system that was also developed for joint replacement applications.

Neurosurgery is another common CAD/CAM application. Figure 20.1, left, shows a NEUROMATE neurosurgical robot. The NEUROMATE procedure planning for positioning a needle or drill guide for cerebral neurosurgical procedures was also performed using CT or MRI. Orthopedic, joint reconstruction surgery, and other neurosurgical procedures have also seen several other similar applications of robotic systems. Ref. [10, Table 1] contains a partial list of surgical CAD/CAM systems.

20.2.2 Teleoperation

Earliest work in surgical teleoperation was aimed at providing assistance in remote or dangerous environments, but current systems perform minimally invasive procedures through small incisions, providing improved visualization and greater dexterity, and reducing the trauma and recovery times. Although long distance surgeries have been attempted, teleoperated surgical systems in common use do not involve large physical separation. The master and slave robots are both placed in the same operating room. The slave robots carry cameras and (often removable) surgical instruments that accurately reproduce scaled motions corresponding to the motion of master manipulators controlled by the surgeon.

The Intuitive Surgical da Vinci Surgical system (Figure 20.2) is currently the only commercially available telerobotic minimally invasive surgery system. Other notables devices include the Computer Motion ZEUS robots, the endoVia Laprotek system, and the several academic efforts (the German Aerospace Center (DLR), the University of Washington, the Johns Hopkins University (JHU), and the University of California, Berkeley).

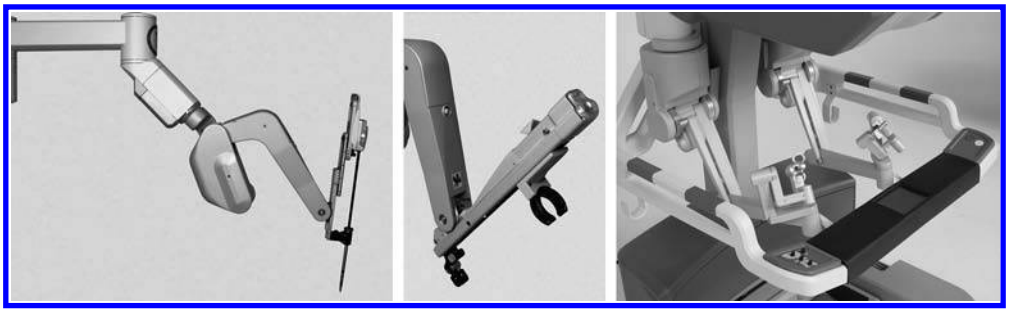

FIGURE 20.2 The da Vinci Surgical System includes multiple instrument (left) manipulators, an endoscopic camera-holding (middle) slave manipulator, and a console containing a stereo video viewer and master manipulators (right). (Copyright Intuitive Surgical, Inc.).

The da Vinci robotic surgery system (now in the third generation) consists of a surgeon's console with a pair of master manipulators (Figure 20.2, right), a set of patient side manipulators (Figure 20.2, left and middle), and a stereo-endoscopic vision equipment cart. The da Vinci system contains four instrument holding slave manipulators, with one dedicated to holding the stereo endoscopic camera. Most recent generation (the da Vinci Si) also allows for up to two surgeon' consoles. The slave manipulators are designed to allow a mechanical constraint at the entry port in the body, and are configurably associated with the master manipulators by using the foot pedals and buttons on the surgeon's console. The scaling of motion between the master manipulators and their corresponding slave motions can also be adjusted using the buttons/touch interface at the surgeon's console.

A variety of 8 mm and 5 mm removable rigid and flexible surgical instruments can be attached to the slave manipulators for specific surgical tasks (e.g., grasping, cutting, suturing, cautery). If including the instrument degrees of freedom, the slave robots can total up to seven degrees of freedom to the tip, allowing greater dexterity than a human wrist. On the other hand, in addition to the substantial system cost, significant annual maintenance expense, and the relatively high cost of the disposable surgical instruments, publications have noted a significant learning curve for clinical proficiency and comparable operating times on the da Vinci systems.

20.2.3 Cooperative Control

Cooperative control envisions the surgeon and the robot sharing the control of the surgical instrument, thus cooperatively controlled robots are best suited for tasks where retention of kinesthetics and ergonomics of operation is a requirement. For example, retinal surgery, and similar microsurgery and micro-neurosurgery are performed under high magnification visualization (e.g., stereo microscope or endoscope) using very light hand-held instruments with visual feedback only. The surgical tasks in these procedures require very high positional accuracy.

Several telesurgical and cooperative systems have been designed specifically for such applications. For example, the cooperatively controlled JHU "Steady Hand" system [13,14] is being developed for retinal surgery and other microsurgical applications. The compact and highly stiff steady hand robot is actuated by highly geared electric actuators (0.002 m/rev for translation, and 50:1 to 200:1 for rotation in one prototype) allowing it to be approximated as a position-controlled device for practical purposes. The robot prototypes [13,15] also include a mechanical pivot constraint similar to the laparoscopic slave robots.

The Steady Hand system is envisioned to be configured around a stereo optical microscope for most applications. The robot may include one or more force sensors in the operating handle that also holds the surgical instrument. The forces sensed from the user (and possibly the environment) are combined [14] to control the motion of the robot using admittance type control. Cooperative control replaces position

scaling advantages of teleoperation with force-scaling. Cooperative control has also been used for normal-scale application of surgical robots, for example, the ACROBOT (http://www.acrobot.co.uk) orthopedic reconstruction robotic system. The ACROBOT combines surgical planning similar to CAD/CAM systems above with cooperative surgical operation.

Taylor et al. [16] describe many other medical robotic devices such as flexible endoluminal snake-like robots, and untethered robots that do not fit the broad categories defined above. For description of control of these specialized devices, the reader is referred to the references contained in [16].

20.3 Computer-Controlled Robots

As discussed above, two initial applications of surgical CAD/CAM were needle/guide positioning in stereotactic neurosurgery, and bone-shaping tasks in orthopedic surgery. Figure 20.3 shows a block level sketch of a surgical CAD/CAM system. Such surgical procedures are planned by detecting abnormalities or disease in preoperative imaging such as CT or MRI. This plan results in a set of task goals for the surgical robot. A process of intraoperative *registration* then colocates these task goals into the robot's workspace at the start of the surgical procedure.

Registration determines the intraoperative spatial relationships between various components of the CAD/CAM system, including those between real-time sensing/imaging, tracked/imaged surgical instruments, the robot, and the anatomy.

Common intraoperative devices for registration include instruments held by the robot, x-ray and fluoroscopy, or optical and electromagnetic trackers, for example, the Optotrak or Polaris systems from Northern Digital, or the Medtronic Axiem EM tracking system. The intraoperative devices are used to collect corresponding anatomical landmarks (a set of points or surfaces etc.) used to establish an optimization problem for computing a coordinate transformation representing the registration.

For most simple tasks, rigid-body transformations are considered adequate. If X_r represents a location in the coordinate system of the robot, and X_{img} the same point in the coordinate system associated with the anatomy in the preoperative imaging, then the process of registration aims to find the relationship:

$$X_r = T_{robot-img} * X_{img} \tag{20.1}$$

where $T_{robot-img}$, a homogeneous transform, is composed of the rotation R and translation P, such that the operation $v' = T * v = R * v + p$ transforms v to v'. Both rigid, and nonrigid methods of registration are common in surgical CAD/CAM systems, although rigid registration (Equation 20.1 is often considered sufficient for orthopedic applications. There is a large body of literature on techniques for registration in surgical CAD/CAM systems, and the reader is referred to [8,16] and references contained within for a more detailed discussion of registration methods.

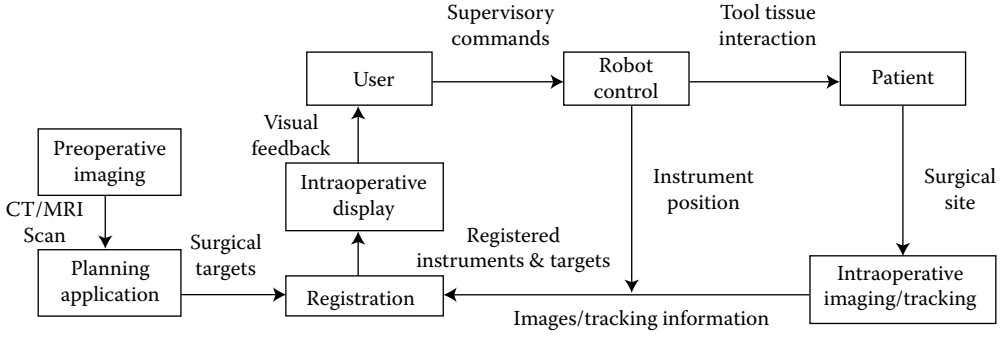

FIGURE 20.3 A block diagram of a surgical CAD/CAM system.

The task goal of the robot is then simply to position its end-effector at the planned task goals transformed using the registration relationships determined above. CAD/CAM robots accomplish their tasks based on geometric relationships computed between targets in patient's anatomy segmented from preoperative images and the robot's coordinate system. Task completion, meeting the task goals provided by the computing engine, is usually subject to geometric precision metrics. For positioning tasks, one such metric may be a RMS error metric.

For additional safety, these automated tasks are still monitored using a range of intra-operative displays intended to provide the surgeon with an updated view of the progress of the task being executed by the robot. In a completely automated surgical CAD/CAM system, the surgeon's role would be limited to planning and supervision for safety. However, currently robots only perform some portions of the procedure that require high precision or safety, and would be difficult for a human to accomplish, leaving the surgeon to complete the remaining procedure.

20.4 Telemanipulation

Telerobotic surgical systems such as the da Vinci surgical system require user input to execute a surgical task. A wide range of feedback options are possible between the master and slave, however, in practice the feedback from the slave is typically limited for reasons of stability. Figure 20.4 outlines a surgical teleoperation system.

Constraints imposed by minimally invasive technique require that teleoperated slave motions pivot about the entry port in the body, while for user convenience the masters typically employ with a spherical wrist. These kinematically dissimilar master and slaves are typically related at the instrument tip and the master wrist handle controlled by the surgeon. If $(T_{slave} = \{R_{slave}, P_{slave}\})$, $(T_{master} = \{R_{master}, P_{master}\})$ represent these coordinate frames, then we have

$$T_{slave} = T_{master} * T_{slave-offset} \tag{20.2}$$

$$T_{master} = T_{slave} * T_{master-offset} \tag{20.3}$$

where T is the rigid homogeneous transformation consisting of a rotation matrix R and a position translation vector P, and the offsets allow for reconfiguration relationships between the master and the slave. For the slave, this can be expanded as,

$$P_{slave} = \frac{1}{m} * P_{master} + P_{slave-offset}, \tag{20.4}$$

$$R_{slave} = R_{master} * R_{slave-offset} \tag{20.5}$$

The scale of operation (m) may allow for master motions to be scaled down to an appropriate slave motion. Such manipulation allows for precise motion of the slave motion while operating the masters in normal human scale operation. Beyond linear scaling described above, nonlinear scaling [12] can

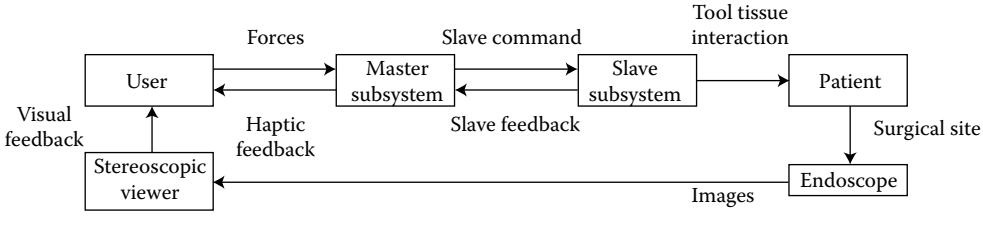

FIGURE 20.4 A block diagram of surgical teleoperation.

allow deformation of the workspace providing accurate manipulation of objects in a much larger total workspace.

It is not typical for orientation to be similarly scaled, but linear and angular velocities are also sometimes connected. Telerobotic systems are operated using visual feedback from a camera imaging the instrument and the work site, and systems such as the da Vinci use a stereo video endoscope for imaging the surgical site, and provide a stereoscopic display to the surgeon for improved depth perception. The slaves typically operate in the camera coordinate system, while the masters operate in a coordinate system measured relative to the user's field of view for natural operation, with the master handles configured to appear at the visualized instrument tips.

Current telerobotic systems limit the feedback that is available from the slave surgical site. The da Vinci surgical system utilizes position–position control that damps feedback to an extent that it can be approximated by an open-loop master position control except for gross slave position errors. The alternative, position–force control may be suitable for surgical tasks if surgical instruments are modified to include instrument tip force sensing. However, telerobotic systems require additional instrumentation to provide force-feedback, an important feedback component that is lost in current implementations.

The overview above does not discuss several important issues, including static compensation, inertia and friction, vibration control, and stability and performance issues associated with master-slave teleoperation. The reader is referred to [12] for a more detailed overview of teleoperation, and robotics texts cited in earlier chapters for details of these finer elements.

20.5 Cooperative Control

As the surgeon and the robot interact with the surgical environment using the same instrument in a cooperative manipulation system, position scaling used in teleoperated systems above is not possible. A cooperative system replaces position scaling with *force-to-motion scaling* to filter unintended forces (e.g., tremor) and provide smoother and more accurate manipulation.

A force sensor integrated in the instrument handle senses user forces, to allow appropriate down scaling of the user input, and a secondary sensor may sense appropriate environment properties (e.g., forces from the surgical instrument, or distance from the tissue surface). If implemented using velocity controlled actuators, an admittance type cooperative controller may aim to drive the surgical instrument with a velocity proportional to the sensed forces.

Assuming only operator applied forces in a noncontact state, and given joint position $x(t)$, the joint velocity $\dot{x}(t)$, desired position $x_d(t)$ and velocities $\dot{x}_d(t)$, we want,

$$x_d(t) = x(t), \quad \dot{x}_d(t) = \dot{x}(t) \tag{20.6}$$

That is, the simplest control law aims to track the user forces $f(t)$, or $\lim_{t\to\infty} \Delta f(t) = 0$, where $\Delta f(t) = f(t) - f_d$, and f_d is the desired force. A control law such as

$$x_d(t) = -k \int \Delta f(t)\, dt \tag{20.7}$$

has been shown to be stable [14] and does not require differentiation of the force signal, or knowledge of the environmental properties. For extremely stiff, and slow moving robots such as the JHU Steady Hand robot, such control provides very precise interactive manipulation.

Figure 20.5 shows a block diagram of cooperative operation. The gain factor (k) depends upon the robot, environment compliance, and force scaling desired. The JHU Steady Hand robot has been shown [14] to be stable for a large range of force scaling gains. These gains can also be dynamically modulated to allow for integration of sensor and image enhanced information overlays and motion constraints.

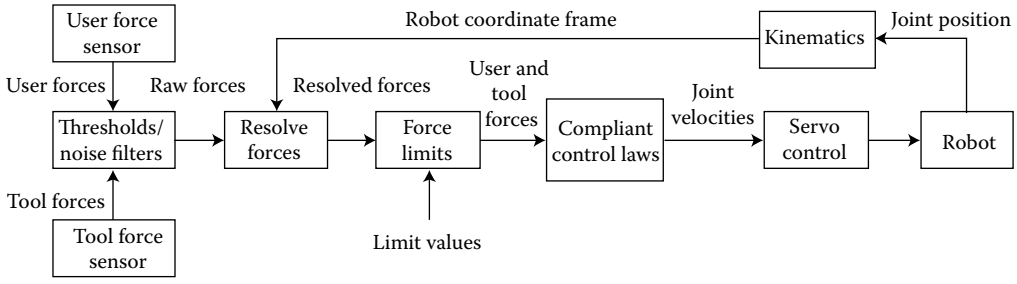

FIGURE 20.5 Block diagram of a directly controlled application.

20.6 NOTES and Flexible Robots

NOTES technology envisions scarless minimally invasive surgery with long flexible instruments being inserted through natural body orifices. As NOTES systems do not employ straight instruments, they are not restricted by the body entry point pivot constraint typical of current laparoscopic minimally invasive robotic systems. NOTES systems may also package multiple instrument carrying manipulators on each delivery mechanism allowing for more instruments to be available at the operation site. Flexible delivery mechanisms may also enable novel surgical approaches because of their ability to provide access around critical structures. Compared to laparoscopic surgeries, NOTES advantages include reduced trauma and faster recovery and improved access. Most NOTES systems currently in development are anticipated to be teleoperated using an architecture similar to the overview in Section 20.4. However, the kinematics, dynamics, and control aspects of the slave manipulators will be specific to the NOTES devices. NOTES disadvantages include the complexity of control and operation, as well as increased difficulty in instrumenting tool–tissue interactions. A discussion of specific approaches is beyond the scope of this overview, and the reader is referred to specific NOTES and endoluminal devices described in Taylor et al. [16].

20.7 Applications

Orthopedic surgeries remain the current leading specialty for robotic surgery CAD/CAM. The ROBODOC system has been used worldwide for over 24,000 joint replacement procedures, and has finally also received approval for human use in the United States in 2008. The ACROBOT, which combines elements of surgical CAD/CAM with cooperative control, is an advanced stage of clinical evaluation. In Neurosurgery, apart from the NEUROMATE, the NeuroArm (MDA Robotics) and several other devices are in development. Other CAD/CAM uses, such as needle holding, for example, the JHU RCM-PAKY (http://urobotics.urology.jhu.edu) robots, have been in clinical use for some time. Other applications of CAD/CAM, for example, brachytherapy have also seen robotic device development. The reader is referred to Fichtinger et al. [4] for details.

Laparoscopic camera holding robots that lead the development of teleoperated robots, such as the AESOP and the more recent Progenics FreeHand robots (http://www.prosurgics.com) are currently in clinical use. Teleoperated systems have been used most frequently in surgery. For example, radical retropubic prostatectomy for treatment of prostate cancer by minimally invasive removal of the prostate has seen the broadest adoption of robotic minimally invasive procedures. Of the approximately 75,000 radical prostatectomies performed in the United States every year for the treatment of prostate cancer, the da Vinci systems now perform a large majority (over 50,000) and have become the dominant treatment for localized prostate cancer, up from approximately 18,000 in 2005, and 8500 procedures performed using

it in 2004. Other complex procedures such as robotic hysterectomies and other gynecological procedures, and some cardiac procedures are also seeing increased rate of acceptance of robotic devices.

While microsurgery cooperative devices have not been used extensively on humans, they are in advanced stage of development for eye surgery, and orthopedic surgery.

20.8 Future

Cost, complexity, and reduced feedback from surgical instruments offset some of the advantages of robotic surgery. An ideal robotic surgery system would restore the haptic feedback available in conventional open surgical techniques while still providing the benefits of minimally or non-invasive technique at a comparable cost. The development of such robotic systems remains a focus of very active research.

As integration of computation increases, functions in the operating room will continue to see increasing automation. Some of these functions, including several tasks currently performed by humans (e.g., nurses and residents), may be transferred to robots. A nurse robot (Penelope, Robotic Systems and Technologies, Inc.) is already in clinical use for instrument inventory management, and its integration with an surgical master–slave system (the da Vinci) has also been explored (as part of the DARPA Trauma-Pod project). Improved clinical technique, such as NOTES, now being developed will also be fueling novel mechanisms that will require improved control algorithms. Intelligent instrumentation, multiuser systems (Figure 20.6), improved human–machine interfaces including haptic feedback from tool–tissue interaction and imaging or information enhanced feedback (*haptic or visual constraints and virtual fixtures*), more sophisticated and fault-tolerant architectures, and robust control even in the presence of communication delays are other active areas of research. Restoring haptic feedback remains a challenge due to the complexity of integrating sensing devices close to the tip of removable and disposable operating instruments for robotic surgery.

It is likely that the progress in these areas will initially be made in the context of systems and applications that are already deployed. Clinical input to the design of these applications has so far been limited to being the end-users of prototypes, and in testing and validation of the clinical techniques associated with the new systems. As experience with these systems increases, improved methods and modes of operation (e.g., automated camera control) will be promoted by the clinical users themselves. Complex and currently

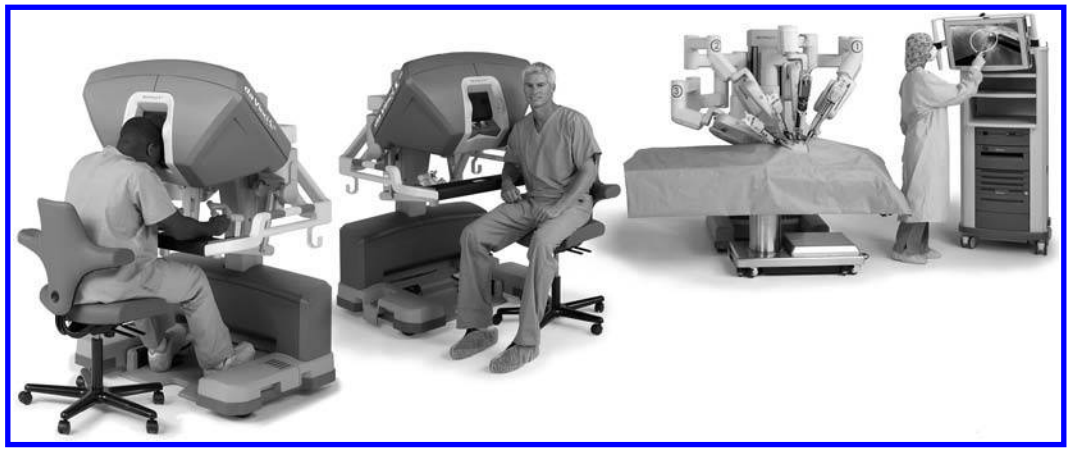

FIGURE 20.6 The da Vinci Si Surgical System introduced in 2009 supports two surgical consoles for multiuser minimally invasive robotic surgery and surgical training. In current operation, instruments are still exclusively controlled by one user but development and validation of appropriate control methods may enable shared control of instruments by more than one user in the future. (Copyright Intuitive Surgical, Inc.).

untreatable procedures, including high-risk cardiac and neurosurgical procedures will continue to retain research focus. Unlike current systems that are moved in and out of conventional operating rooms, future operating rooms are likely to see integrated robotic technology, such as ceiling and wall integrated robotics, that is built into the operating room.

References

1. R. Thaly, K. Shah, and V. R. Patel. Applications of robots in urology. *Journal of Robotic Surgery*, 1:3–17, 2007.
2. R. H. Taylor, B. D. Mittelstadt, H. A. Paul, W. Hanson, P. Kazanzides, J. F. Zuhars, B. Williamson, B. L. Musits, E. Glassman, and W. L. Bargar. An image-directed robotic system for precise orthopaedic surgery. *IEEE Transactions on Robotics and Automation*, 10(3), 1994.
3. G. Guthart and J. Salisbury. The intuitive telesurgery system: Overview and application. In *IEEE International Conference on Robotics and Automation, ICRA 2000*, April 24-28, San Francisco, CA, USA, pp. 618–621, 2000.
4. G. Fichtinger, P. Kazanzides, A. Okamura, G. Hager, L. Whitcomb, and R. Taylor. Surgical and interventional robotics: Part II—Surgical cad-cam systems. *IEEE Robotics and Automation Magazine*, 15(3):94–102, 2008.
5. R. H. Taylor, S. Lavallee, G. Burdea, and R. Mosges. *Computer-Integrated Surgery Technology and Clinical Applications*. MIT Press, Cambridge, MA, 1995.
6. R. A. Faust, ed. *Robotics in Surgery: History, Current and Future Applications*. Nova Science Publishers, Inc., Hauppauge, NY, 2006.
7. T. Peters and K. Cleary, Eds. *Image-Guided Interventions: Technology and Applications*. Springer Science + Business Media LLC, New York, NY, 2008.
8. P. Kazanzides, G. Fichtinger, G. D. Hager, A. M. Okamura, L. L. Whitcomb, and R. H. Taylor. Surgical and interventional robotics: Part I—Core concepts, technology, and design. *IEEE Robotics and Automation Magazine*, 15(2):122–130, 2008.
9. G. Hager, A. Okamura, P. Kazanzides, L. Whitcomb, G. Fichtinger, and R. Taylor. Surgical and interventional robotics: Part III—Surgical assistance systems. *IEEE Robotics and Automation Magazine*, 15(4):84–93, 2008.
10. R. H. Taylor and D. Stoianovici. Medical robotics in computer-integrated surgery. *IEEE Transactions on Robotics and Automation*, 19(5):765–781, 2003.
11. D. B. Camarillo, T. M. Krummel, and J. K. Salisbury. Robotic technology in surgery: Past, present, and future. *The American Journal of Surgery*, 188(4A-Suppl.):2–15, 2004.
12. G. Niemeyer, C. Preusche, and G. Hirzinger. Telerobotics. In Siciliano, B. and Khatib, O. (eds.), *Springer Handbook of Robotics*, pp. 741–757. Springer-Verlag, Berlin/Heidelberg, 2008.
13. R. Taylor, P. Jensen, W. Whitcomb, A. Barnes, D. Kumar, R. Stoianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi. A steady-hand robotic system for microsurgical augmentation. *International Journal of Robotics Research*, 18(12):1201–1210, 1999.
14. R. Kumar, P. Berkelman, P. Gupta, A. Barnes, P. S. Jensen, L. L. Whitcomb, and R. H. Taylor. Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation. In *IEEE International Conference on Robotics and Automation, ICRA 2000*, April 24-28, San Francisco, CA, USA, pp. 610–617, 2000.
15. B. Mitchell, J. Koo, I. Iordachita, P. Kazanzides, A. Kapoor, J. Handa, G. Hager, and R. Taylor. Development and application of a new steady-hand manipulator for retinal surgery. In *IEEE International Conference on Robotics and Automation, ICRA 2007*, April 10–14, Rome, Italy, pp. 623–629, 2007.
16. R. Taylor, A. Menciassi, G. Fichtinger, and P. Dario. Medical robotics and computer-integrated surgery, In: Siciliano, B. and Khatib, O. (eds.), *Springer Handbook of Robotics*, pp. 1199–1222. Springer-Verlag, Berlin/Heidelberg, 2008.

21

Stochastic Gene Expression: Modeling, Analysis, and Identification*

Mustafa Khammash
University of California, Santa Barbara

Brian Munsky
Los Alamos National Lab

21.1 Introduction

In living cells, many key events such as gene expression and protein–protein interactions follow from elementary reactions between the cellular consituents at the molecular level (e.g., genes, RNAs, proteins). There is considerable inherent randomness in the order and timing of these reactions. This randomness can be attributed to the random collisions among cellular constituents whose motion is induced by thermal energy and follows specific statistical distributions. The result is fluctuations in the molecular copy numbers of reaction products both among similar cells and within a single cell over time. These fluctuations (commonly referred to as cellular noise) can propagate downstream, impacting events and processes in accordance to the dynamics of the network interconnection.

Cellular noise has been measured experimentally and classified based on its source [1,2]: intrinsic noise refers to noise originating within the boundaries of the process under consideration and is due to the

* This chapter is an expanded version of a conference paper that appeared in the Proceedings of IFAC 2009 SYSID.

inherent discrete nature of the chemical process of gene expression, whereas extrinsic noise has origins that are more global and affects all processes in the cell under consideration in a similar way (e.g., fluctuations in regulatory protein copy numbers, RNA polymerase numbers, cell-cycle). Noise, both intrinsic and extrinsic, plays a critical role in biological processes. In [3,4] it was proposed that lysis–lysogeny fate decisions for phage λ are determined by a noise driven stochastic switch, implying that the fate of a given cell is determinable only in a probabilistic sense. Another stochastic switch which governs the piliation of *E. coli* has been modeled in [5]. Aside from endogenous switches, bistable genetic switches have been constructed and tested [6,7]. Depending on their parameters, such switches can be quite susceptible to noise. In [8], the first synthetic oscillator was reported. This novel circuit, called the repressilator, consists of three genes, each having a product that represses the next gene, thereby creating a feedback loop of three genes. The role of noise in the operation of the repressilator was recently studied in [9]. Yet another curious effect of noise can be seen in the fluctuation enhanced sensitivity of intracellular regulation termed "stochastic focusing" and reported in [10]. In gene expression, noise-induced fluctuations in gene products have been studied in [11–20]. Many of these studies look at the propagation of noise in gene networks and the impact (and sometimes limitations) of various types of feedback in suppressing such fluctuations.

In this article, we give an overview of the methods used for modeling and analysis of fluctuations in gene networks. We also demonstrate that these fluctuations can be used in identifying model parameters that may be difficult to measure. The presentation follows that in [21,22].

21.1.1 Deterministic versus Stochastic Modeling

The most common approach for modeling chemical reactions relies on the law of mass action to derive a set of differential equations that characterize the evolution of reacting species concentrations over time. As an example, consider the reaction $A + B \xrightarrow{k} C$. A deterministic formulation of chemical kinetics would yield the following description $\frac{d[C]}{dt} = k[A] \cdot [B]$ where $[\cdot]$ denotes the concentration, which is considered to be a continuous variable. In contrast, a discrete stochastic formulation of the same reaction describes the *probability* that at a given time, t, the number of molecules of species A and B take certain integer values. In this way, populations of the species within the network of interest are treated as random variables. In this stochastic description, reactions take place randomly according to certain probabilities determined by several factors including reaction rates and species populations. For example, given certain integer populations of A and B, say N_A and N_B, at time t, the probability that one of the above reactions takes place within the interval $[t, t + dt)$ is proportional to $\frac{N_A \cdot N_B}{\Omega} dt$, where Ω is the volume of the space containing the A and B molecules. In this mesoscopic stochastic formulation of chemical kinetics, molecular species are characterized by their probability density function, which quantifies the amount of fluctuations around a certain mean value. As we show below, in the limit of an infinite number of molecules and infinite volume (the thermodynamic limit), fluctuations become negligible and the mesoscopic description converges to the macroscopic description obtained from mass-action kinetics. In typical cellular environments where small volumes and molecule copy numbers are the rule, mesoscopic stochastic descriptions offer more accurate representations of chemical reactions and their fluctuations. Such fluctuations need to be accounted for as they can generate distinct phenomena that simply cannot be captured by deterministic descriptions. In fact, in certain examples (see e.g., *stochastic focusing* in Figure 21.1) the deterministic model fails to even capture the stochastic mean, underscoring the need for stochastic models.

21.2 Stochastic Chemical Kinetics

In this section, we provide a more detailed description of the stochastic framework for modeling chemical reactions. In the stochastic formulation of chemical kinetics we shall consider a chemically reacting system of volume Ω containing N molecular species $S_1, \ldots, S_N$ which react via M known reaction channels

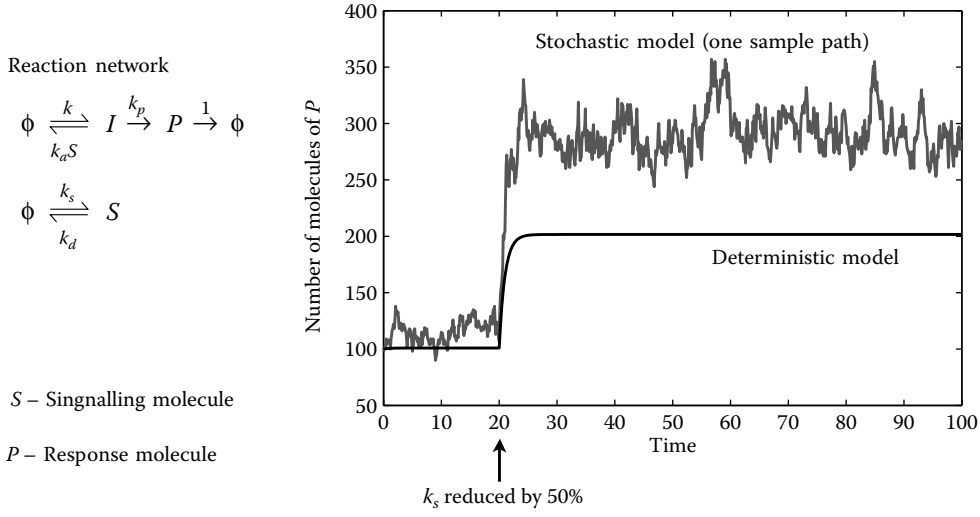

FIGURE 21.1 The reaction system shown on the left represents a signaling species S and its response P. I is an intermediate species. When the system is modeled deterministically, the concentration of P fails to capture the stochastic mean of the same species computed from a stochastic model, as shown in the simulations shown in the figure. This example system and the stochastic focusing phenomenon are described in Ref. 10. (Adapted from J. Paulsson, O. Berg, and M. Ehrenberg. *Proceedings of the National Academy of Sciences*, 97:7148–7153, 2000.)

$R_1, \ldots, R_M$. We shall make the key assumption that the entire reaction system is well-stirred and is in thermal equilibrium. While this assumption does not always hold in examples of biological networks, spatial models of stochastic chemical kinetics can be formulated. In the well-mixed case that we focus on here, the reaction volume is at a constant temperature T and the molecules move due to the thermal energy. The velocity of a molecule in each of the three spacial directions is independent from the other two and is determined according to a Boltzman distribution:

$$f_{v_x}(v) = f_{v_y}(v) = f_{v_z}(v) = \sqrt{\frac{m}{2\pi k_B T}}\, e^{-\frac{m}{2k_B T}v^2}$$

where m is its mass, v its velocity, and k_B is Boltzman's constant. Let $X(t) = (X_1(t) \ldots X_N(t))^T$ be the state vector, where $X_i(t)$ is a random variable that describes the number of molecules of species S_i in the system at time t. We consider elementary reactions, which may be either mono-molecular: $S_i \rightarrow$ Products, or bi-molecular: $S_i + S_j \rightarrow$ Products. More complex reactions can be achieved by introducing intermediate species that interact through a sequence of elementary reactions. In this formulation, each reaction channel R_k defines a transition from some state $\mathbf{X} = \mathbf{x}_i$ to some other state $\mathbf{X} = \mathbf{x}_i + \mathbf{s}_k$, which reflects the change in the state after the reaction has taken place. $\mathbf{s}_k$ is known as the *stoichiometric vector*, and the set of all M reactions give rise to the *stoichiometry matrix* defined as

$$\mathbf{S} = [\mathbf{s}_1 \ldots \mathbf{s}_M].$$

Associated with each reaction R_k is a *propensity function*, $w_k(\mathbf{x})$, which captures the rate of the reaction k. Specifically, $w_k(\mathbf{x})dt$ is the probability that, given the system is in state $\mathbf{x}$ at time t, the kth reaction will take place in the time interval $[t, t + dt)$. The propensity function for various reaction types is given in Table 21.1.

If we denote by k, k', and k'' the reaction rate constants from deterministic mass-action kinetics for the first, second, and third reaction types, it can be shown that $c = k$, $c' = k'/\Omega$, and $c'' = 2k''/\Omega$.

TABLE 21.1 Propensity Function for the Various Elementary Reactions

Reaction type	Propensity function
$S_i \rightarrow$ Products	$c\mathbf{x}_i$
$S_i + S_j \rightarrow$ Products $(i \neq j)$	$c'\mathbf{x}_i\mathbf{x}_j$
$S_i + S_i \rightarrow$ Products	$c''\mathbf{x}_i(\mathbf{x}_i - 1)/2$

21.2.1 Sample Path Representation and Connection with Deterministic Models

A sample path representation of the stochastic process $X(t)$ can be given in terms of independent Poisson processes $Y_k(\lambda)$ with parameter λ. In particular, it can be shown [23] that

$$\mathbf{X}(t) = \mathbf{X}(0) + \sum_k \mathbf{s}_k Y_k \left(\int_0^t w_k(\mathbf{X}(s))\, ds \right).$$

Hence, the Markov process $\mathbf{X}(t)$ can be represented as a random time-change of other Markov processes. When the integral is approximated by a finite sum, the result is an approximate method for generating sample paths, which is commonly referred to as tau leaping [24]. The sample path representation shown here is of theoretical interest as well. Together with the Law of Large numbers, it is used to establish a connection between deterministic and stochastic representations of the same chemical system.

In a deterministic representation based on conventional mass-action kinetics, the solution of the deterministic reaction rate equations describes the trajectories of the concentrations of species $S_1, \ldots, S_N$. Let these concentrations be denoted by $\mathbf{\Phi}(t) = [\Phi_1(t), \ldots, \Phi_N(t)]^T$. Accordingly, $\Phi(\cdot)$ satisfies the mass-action ordinary differential equation (ODE):

$$\dot{\mathbf{\Phi}} = \mathbf{S}f(\mathbf{\Phi}(t)), \quad \mathbf{\Phi}(0) = \mathbf{\Phi}_0.$$

For a meaningful comparison with the stochastic solution, we shall compare the function $\mathbf{\Phi}(t)$ with the volume-normalized stochastic process $\mathbf{X}^\Omega(t) := \mathbf{X}(t)/\Omega$. A natural question is: how does $\mathbf{X}^\Omega(t)$ relate to $\mathbf{\Phi}(t)$? The answer is given by the following fact, which is a consequence of the Law of Large numbers [23]:

Fact 21.1:

Let $\mathbf{\Phi}(t)$ be the deterministic solution to the reaction rate equations

$$\frac{d\mathbf{\Phi}}{dt} = \mathbf{S}f(\mathbf{\Phi}), \quad \mathbf{\Phi}(0) = \mathbf{\Phi}_0.$$

Let $\mathbf{X}^\Omega(t)$ be the stochastic representation of the same chemical systems with $\mathbf{X}^\Omega(0) = \mathbf{\Phi}_0$. Then for every $t \geq 0$:

$$\lim_{\Omega \to \infty} \sup_{s \leq t} \left| \mathbf{X}^\Omega(s) - \mathbf{\Phi}(s) \right| = 0 \quad \text{almost surely.}$$

To illustrate the convergence of the stochastic system to the deterministic description, we consider a simple one species problem with the following nonlinear reaction description:

Reaction	Stoichiometry	Deterministic Description	Stochastic Description
$R_1:$	$\emptyset \rightarrow S,$	$f_1(\phi) = 20 + 40\frac{\phi}{40^{10}+\phi^{10}},$	$w_1(X) = \Omega\left(20 + 40\frac{X/\Omega}{40^{10}+(X/\Omega)^{10}}\right)$
$R_2:$	$S \rightarrow \emptyset,$	$f_2(\phi) = \phi,$	$w_2(X) = \Omega\left(X/\Omega\right).$

From Figure 21.2a, which illustrates the production and degradation terms of the reaction rate equation, one can see that the deterministic model has three equilibrium points where these terms are equal. Figure 21.2a shows the deterministic (smooth) and stochastic (jagged) trajectories of the system from two different initial conditions: $\phi(0) = X(0)/\Omega = 0$ and $\phi(0) = X(0)/\Omega = 100$ and three different volumes $\Omega = \{1, 3, 10\}$. From the plot, it is clear that as the volume increases, the difference between the stochastic and deterministic process shrinks. This is the case for almost every possible initial condition, but with one obvious exception. If the initial condition were chosen to correspond to the unstable equilibrium, then the deterministic process would remain at equilibrium, but the noise-driven stochastic process would not. Of course, this unsteady equilibrium corresponds to a single point of zero measure, thus illustrating the nature of the "almost sure" convergence.

Hence, in the thermodynamic limit, the stochastic description converges to the deterministic one. While this result establishes a fundamental connection which ties together two descriptions at two scales,

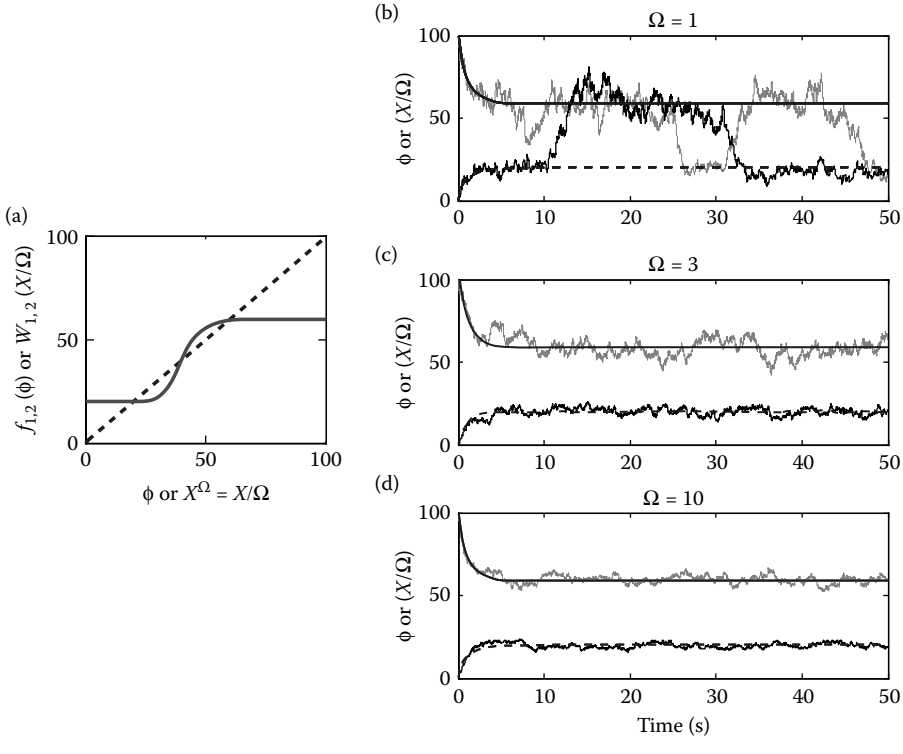

FIGURE 21.2 Convergence of the stochastic and deterministic descriptions with volume scaling. (a) Reaction rates for the production (solid) and degradation (dashed) events. (b–d) Trajectories of the deterministic (smooth) and stochastic representations (jagged) assuming equivalent initial conditions for different volumes: (b) $\Omega = 1$, (c) $\Omega = 3$, (d) $\Omega = 10$.

in practice the large volume assumption cannot be justified as the cell volume is fixed, and stochastic descriptions could differ appreciably from their large volume limit.

21.2.2 The Forward Kolmogorov Equation

The chemical master equation (CME), or the forward Kolmogorov equation, describes the time-evolution of the probability that the chemical reaction system is in any given state, say $\mathbf{X}(t) = \mathbf{x}$. The CME can be derived from the Markov property of chemical reactions. Let $P(\mathbf{x}, t)$, denote the probability that the system is in state $\mathbf{x}$ at time t. We can express $P(\mathbf{x}, t + dt)$ as follows:

$$P(\mathbf{x}, t + dt) = P(\mathbf{x}, t)(1 - \sum_k w_k(\mathbf{x})\, dt) + \sum_k P(\mathbf{x} - \mathbf{s}_k, t) w_k(\mathbf{x} - \mathbf{s}_k)\, dt + \mathcal{O}(dt^2).$$

The first term on the right-hand side is the probability that the system is already in state $\mathbf{x}$ at time t and no reactions occur to change that in the next dt. In the second term on the right-hand side, the kth term in the summation is the probability that the system at time t is an R_k reaction away from being at state $\mathbf{x}$, and that an R_k reaction takes place in the next dt.

Moving $P(\mathbf{x}, t)$ to the left-hand side, dividing by dt, and taking the limit as dt goes to zero we get the CME:

$$\frac{dP(\mathbf{x}, t)}{dt} = \sum_{k=1}^{M} [w_k(\mathbf{x} - \mathbf{s}_k) P(\mathbf{x} - \mathbf{s}_k, t) - w_k(\mathbf{x}) P(\mathbf{x}, t)]$$

21.3 Stochastic Analysis Tools

Stochastic analysis tools may be broadly divided into four categories. The first consists of kinetic Monte Carlo methods, which compute sample paths whose statistics are used to extract information about the system. The second class of methods consists of approximations of the stochastic process $\mathbf{X}(t)$ by solutions of certain stochastic differential equations (SDE). The third type of methods seek to compute the trajectories of various moments of $\mathbf{X}(t)$, while the fourth type is concerned with computing the evolution of probability densities of the stochastic process $\mathbf{X}(t)$.

21.3.1 Kinetic Monte Carlo Simulations

Because the CME is often infinite dimensional, the majority of analyses at the mesoscopic scale have been conducted using kinetic Monte Carlo algorithms. The most widely used of these algorithms is Gillespie's stochastic simulation algorithm (SSA) [25] and its variants. These are described next.

21.3.1.1 The Gillespie Algorithm

Each step of Gillespie's SSA begins at a time t and at a state $\mathbf{X}(t) = \mathbf{x}$ and is comprised of three substeps: (1) generate the time until the next reaction; (2) determine which reaction occurs at that time; and (3) update the time and state to reflect the previous two choices. The SSA approach is exact in the sense that it results in a random variable with a probability distribution exactly equal to the solution of the corresponding CME. However, each run of the SSA provides only a single trajectory. Numerous trajectories are generated which are then used to compute statistics of interest.

We now describe these steps in more detail. To each of the reactions $\{R_1, \ldots, R_M\}$ we associate a random variable $\mathcal{T}_i$, which describes the time for the next firing of reaction R_i. A key fact is that $\mathcal{T}_i$ is exponentially distributed with parameter w_i. From these, we can define two additional random variables,

one continuous and the other discrete:

$$\mathcal{T} = \min_i\{\mathcal{T}_i\} \qquad \text{(Time to the next reaction)}$$
$$\mathcal{R} = \arg\min_i\{\mathcal{T}_i\} \qquad \text{(Index of the next reaction)}$$

It can be shown that: (a) $\mathcal{T}$ is exponentially distributed with parameter: $\sum_i w_i$; and (b) $\mathcal{R}$ has the discrete distribution: $P(\mathcal{R} = k) = \dfrac{w_k}{\sum_i w_i}$. With this in mind, we are ready to give the steps in Gillespie's SSA.

Gillespie's SSA:

- *Step 0:* Initialize time t and state population $\mathbf{x}$.
- *Step 1:* Draw a sample τ from the distribution of $\mathcal{T}$ (see Figure 21.3).
- *Step 2:* Draw a sample μ from the distribution of $\mathcal{R}$ (see Figure 21.3).
- *Step 3:* Update time: $t \leftarrow t + \tau$. Update the state: $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{s}_\mu$.

21.3.2 Stochastic Differential Equation Approximations

There are several SDE approximations of the stochastic process $\mathbf{X}(t)$. One of these is the so-called *chemical Langevin equation*, also called the *diffusion approximation* [26,27]. We will not discuss this here, but instead examine another SDE approximation that relates to SDEs that arise naturally in systems and control settings.

The van Kampen's approximation or linear noise approximation (LNA) (see [28–30]) is essentially an approximation to the process $\mathbf{X}(t)$ that takes advantage of the fact that in the large volume limit ($\Omega \to \infty$), the process $X^\Omega(t) := \mathbf{X}(t)/\Omega$ converges to the solution $\mathbf{\Phi}(t)$ of the deterministic reaction rate equation: $\dot{\mathbf{\Phi}}(t) = f(\mathbf{\Phi})$. Defining a scaled "error" process $\mathbf{V}^\Omega(t) := \sqrt{\Omega}\big(\mathbf{X}^\Omega(t) - \mathbf{\Phi}(t)\big)$ and using the Central limit

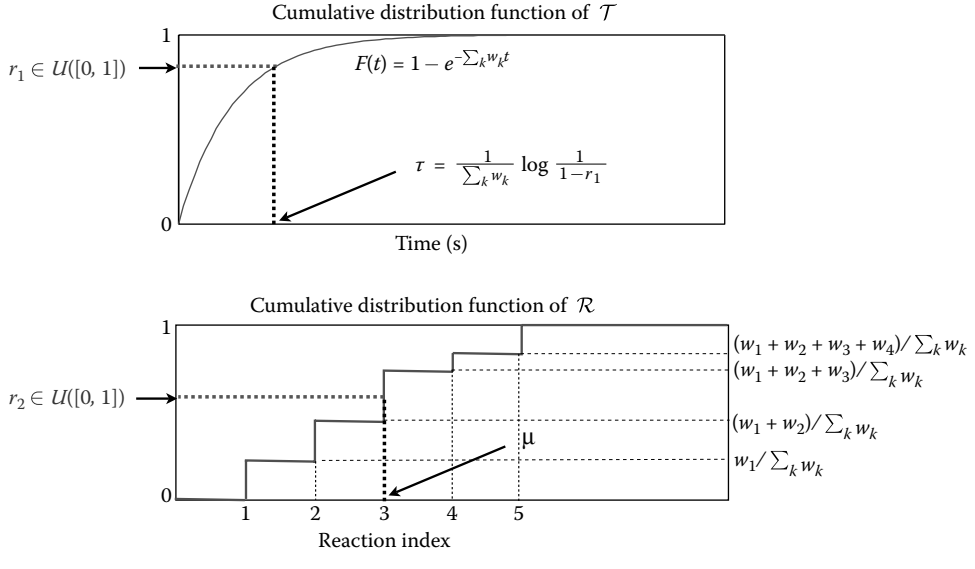

FIGURE 21.3 Cumulative distribution of the two random variables $\mathcal{T}$ and $\mathcal{R}$. A sample of $\mathcal{T}$ is drawn by first drawing a uniformly distributed random number r_1 and then finding its inverse image under F, the cumulative distribution of $\mathcal{T}$. A similar procedure can be used to draw a sample from the distribution of $\mathcal{R}$.

theorem, it can be shown that $V^{\Omega}(t)$ converges in distribution to the solution $V(t)$ to the following linear SDE:

$$d\mathbf{V}(t) = \mathbf{J}_f(\mathbf{\Phi})\mathbf{V}(t)\,dt + \sum_{k=1}^{M} \mathbf{s}_k \sqrt{w_k(\mathbf{\Phi})}\,d\mathbf{B}_k(t),$$

where $\mathbf{J}_f$ denotes the Jacobian of $f(\cdot)$ and $\mathbf{B}_k$ is standard Brownian motion [23]. Hence, the LNA results in a state $\mathbf{X}(t) \approx \Omega\mathbf{\Phi}(t) + \sqrt{\Omega}\mathbf{V}(t)$, which can be viewed as the sum of a deterministic term given by the solution to the deterministic reaction rate equation, and a zero mean stochastic term given by the solution to a linear SDE. While the LNA is reasonable for systems with sufficiently large numbers of molecules (and volume), examples show that it can yield poor results when this assumption is violated, for example, when the system of interest contains species with very small molecular counts, or where the reaction propensity functions are strongly nonlinear over the dominant support region of the probability density function.

21.3.3 Statistical Moments

When studying stochastic fluctuations that arise in gene networks, one is often interested in computing moments and variances of noisy expression signals. The evolution of moment dynamics can be described using the CME. To compute the first moment $\mathbb{E}[X_i]$, we multiply the CME by x_i and then sum of all $(x_1, \ldots, x_N) \in \mathbb{N}^N$ to get

$$\frac{d\mathbb{E}[X_i]}{dt} = \sum_{k=1}^{M} s_{ik}\mathbb{E}[w_k(X)]$$

Similarly, to get the second moments $\mathbb{E}[X_iX_j]$, we multiply the CME by x_ix_j and sum over all $(x_1, \ldots, x_N) \in \mathbb{N}^N$, which gives

$$\frac{d\mathbb{E}[X_iX_j]}{dt} = \sum_{k=1}^{M} s_{ik}\mathbb{E}[X_jw_k(X)] + \mathbb{E}[X_iw_k(X)]s_{jk} + s_{ik}s_{jk}\mathbb{E}[w_k(X)]$$

These last two equations can be expressed more compactly in matrix form. Defining $\mathbf{w}(x) = [w_1(x), \ldots, w_M(x)]^T$, the moment dynamics become:

$$\frac{d\mathbb{E}[\mathbf{X}]}{dt} = \mathbf{S}\mathbb{E}[\mathbf{w}(\mathbf{X})]$$

$$\frac{d\mathbb{E}[\mathbf{X}\mathbf{X}^T]}{dt} = \mathbf{S}\mathbb{E}[\mathbf{w}(\mathbf{X})\mathbf{X}^T] + \mathbb{E}[\mathbf{w}(\mathbf{X})\mathbf{X}^T]^T\mathbf{S}^T + \mathbf{S}\{diag\mathbb{E}[\mathbf{w}(\mathbf{X})]\}\mathbf{S}^T$$

In general, this set of equations cannot be solved explicitly. This is because the moment equations will not always be closed: depending on the form of the propensity vector $w(\cdot)$, the dynamics of the first moment $\mathbb{E}(\mathbf{X})$ may depend on the second moments $\mathbb{E}(\mathbf{X}\mathbf{X}^T)$, the second moment dynamics may in turn depend on the third moments, and so on, resulting in an infinite system of ODE's. However, when the propensity function is affine, that is, $\mathbf{w}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{w}_0$, where $\mathbf{W}$ is $N \times N$ and $\mathbf{w}_0$ is $N \times 1$, then $\mathbb{E}[\mathbf{w}(\mathbf{X})] = \mathbf{W}\mathbb{E}[\mathbf{X}] + \mathbf{w}_0$, and $\mathbb{E}[\mathbf{w}(\mathbf{X})\mathbf{X}^T] = \mathbf{W}\mathbb{E}[\mathbf{X}\mathbf{X}^T] + \mathbf{w}_0\mathbb{E}[\mathbf{X}^T]$. This gives us the following moment equations:

$$\frac{d}{dt}\mathbb{E}[\mathbf{X}] = \mathbf{S}\mathbf{W}\mathbb{E}[\mathbf{X}] + \mathbf{S}\mathbf{w}_0$$

$$\frac{d}{dt}\mathbb{E}[\mathbf{X}\mathbf{X}^T] = \mathbf{S}\mathbf{W}\mathbb{E}[\mathbf{X}\mathbf{X}^T] + \mathbb{E}[\mathbf{X}\mathbf{X}^T]\mathbf{W}^T\mathbf{S}^T + \mathbf{S}\,diag(\mathbf{W}\mathbb{E}[\mathbf{X}] + \mathbf{w}_0)\mathbf{S}^T + \mathbf{S}\mathbf{w}_0\mathbb{E}[\mathbf{X}^T] + \mathbb{E}[\mathbf{X}]\mathbf{w}_0^T\mathbf{S}^T$$

Clearly, this is a closed system of linear ODEs that can be solved easily for the first and second moments.

Defining the covariance matrix $\mathbf{\Sigma} = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}(\mathbf{X})]^T]$, we can also compute covariance equations:

$$\frac{d}{dt}\mathbf{\Sigma} = \mathbf{S}\mathbf{W}\mathbf{\Sigma} + \mathbf{\Sigma}\mathbf{W}^T\mathbf{S}^T + \mathbf{S}\, diag(\mathbf{W}\mathbb{E}[\mathbf{X}] + \mathbf{w}_0)\mathbf{S}^T$$

The steady-state moments and covariances can be obtained by solving linear algebraic equations. Let $\bar{\mathbf{X}} = \lim_{t\to\infty} \mathbb{E}[\mathbf{X}(t)]$ and $\bar{\mathbf{\Sigma}} = \lim_{t\to\infty} \mathbf{\Sigma}(t)$. Then

$$\mathbf{S}\mathbf{W}\bar{\mathbf{X}} = -\mathbf{S}\mathbf{w}_0$$
$$\mathbf{S}\mathbf{W}\bar{\mathbf{\Sigma}} + \bar{\mathbf{\Sigma}}\mathbf{W}^T\mathbf{S}^T + \mathbf{S}\, diag(\mathbf{W}\bar{\mathbf{X}} + \mathbf{w}_0)\mathbf{S}^T = 0$$

The latter is an algebraic Lyapunov equation that can be solved efficiently.

21.3.3.1 Moment Closures

An important property of the Markov processes that describe chemical reactions is that when one constructs a vector μ with all the first- and second-order statistical uncentered moments of the process' state $\mathbf{X}$, this vector evolves according to a *linear* equation of the form

$$\dot{\mu} = \mathbf{A}\mu + \mathbf{B}\bar{\mu}. \tag{21.1}$$

Unfortunately, as pointed out earlier, Equation 21.1 is not in general a closed system because the vector $\bar{\mu}$ may contain moments of order larger than two, whose evolution is not provided by Equation 21.1. In fact, this will always be the case when bi-molecular reactions are involved. A technique that can be used to overcome this difficulty consists of approximating the *open linear* system (Equation 21.1) by a *closed nonlinear* system

$$\dot{\nu} = \mathbf{A}\nu + \mathbf{B}\varphi(\nu), \tag{21.2}$$

where ν is an approximation to the solution μ to Equation 21.1 and $\varphi(\cdot)$ is a *moment closure function* that attempts to approximate the moments in $\bar{\mu}$ based on the values of the moments in μ. The construction of $\varphi(\cdot)$ often relies on postulating a given type for the distribution of X and then expressing the higher-order moments in $\bar{\mu}$ by a nonlinear function $\varphi(\mu)$ of the first- and second-order moments in μ. Authors construct moment closure functions $\varphi(\cdot)$ based on different assumed distributions for $\mathbf{X}$, which include normal [31–33], lognormal [34,35], Poisson, and binomial [36] distributions. Here we discuss only the normal and lognormal moment closure method.

1. *Normal Distribution:* Assuming that the populations of each species follow a multivariate normal distribution leads to the equation:

$$\mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])(X_k - \mathbb{E}[X_k])] = 0$$

   from which an expression for the third order moment $\mathbb{E}[X_iX_jX_k]$ in terms of lower-order moments can be obtained. When substituted in the moment (Equations 21.1), a closed-system results. This is referred to as the mass-fluctuation kinetics in [33]. As long as the reaction rates are at most second order, only the expressions for the third moments will be necessary—all of which can be determined as above. For third or higher-order propensity functions, the resulting higher order moments can also be easily expressed in terms of the first two, using moment and generating functions as described in the example section below.
2. *Lognormal Distribution:* Based on a lognormal distribution for $\mathbf{X}$, one obtains the following equation:

$$\mathbb{E}[X_iX_jX_k] = \frac{\mathbb{E}[X_iX_j]\mathbb{E}[X_jX_k]\mathbb{E}[X_iX_k]}{\mathbb{E}[X_i]\mathbb{E}[X_j]\mathbb{E}[X_k]}.$$

As before, this leads to a closed-system when substituted in the moment (Equation 21.1), provided that the reactions in the system are at most bimolecular. In [37] it was shown that this moment closure results without any *a priori* assumptions on the shape of the distribution for $\mathbf{X}$ by matching all (or a large number of) the time derivatives of the exact solution for Equation 21.1 with the corresponding time derivatives of the approximate solution for Equation 21.2, for a given set of initial conditions. However, for systems with third or higher order terms in the reaction rates, it is more difficult to find expressions for the higher moments necessary to close the system.

When the population standard deviations are not much smaller than the means, choosing $\varphi(\cdot)$ based on a normal distribution assumption often leads to less accurate approximations. Furthermore, normal distributions of $\mathbf{X}$ allows for negative values of $\mathbf{X}$, which clearly does not reflect the positive nature of the populations represented by $\mathbf{X}(t)$. In these cases, a lognormal or other positive distribution closure may be preferred, but at the cost of more complicated closure expressions for the higher-order moments.

21.3.4 Density Computations

Another approach to analyze models described by the CME aims to compute the probability density functions for the random variable $\mathbf{X}$. This is achieved by approximate solutions of the CME, using a new analytical approach called the finite state projection (FSP) [38–41]. The FSP approach relies on a projection that preserves an important subset of the state space (e.g., that supporting the bulk of the probability distribution), while projecting the remaining large or infinite states onto a single "absorbing" state (see Figure 21.4.)

Probabilities for the resulting finite-state Markov chain can be computed exactly, and can be shown to give a lower bound for the corresponding probability for the original full system. The FSP algorithm provides a means of systematically choosing a projection of the CME, which satisfies any prespecified accuracy requirement. The basic idea of the FSP is as follows. In matrix form, the CME may be written as $\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t)$, where $\mathbf{P}(t)$ is the (infinite) vector of probabilities corresponding to each possible state in the configuration space. The generator matrix $\mathbf{A}$ embodies the propensity functions for transitions from one configuration to another and is defined by the reactions and the enumeration of the configuration space. A projection can now be made to achieve an arbitrarily accurate approximation as outlined next: Given an index set of the form $J = \{j_1, j_2, j_3, \ldots\}$ and a vector $\mathbf{v}$, let $\mathbf{v}_J$ denote the subvector of $\mathbf{v}$ chosen according to J, and for any matrix $\mathbf{A}$, let $\mathbf{A}_J$ denote the submatrix of $\mathbf{A}$ whose rows and columns have been chosen according to J. With this notation, we can restate the result from [38,41]: *consider any distribution which evolves according to the linear ODE* $\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t)$. *Let* $\mathbf{A}_J$ *be a principle submatrix of* $\mathbf{A}$ *and* $\mathbf{P}_J$ *be a subvector of* $\mathbf{P}$, *both corresponding to the indexes in J. If for a given* $\varepsilon > 0$ and $t_f \geq 0$ *we have that* $\mathbf{1}^T \exp(A_J t_f)\mathbf{P}_J(0) \geq 1 - \varepsilon$, *then*

$$\| \exp(\mathbf{A}_J t_f)\mathbf{P}_J(0) - \mathbf{P}_J(t_f)\|_1 \leq \varepsilon,$$

which provides a bound on the error between the exact solution $\mathbf{P}_J$ to the (infinite) CME and the matrix exponential of the (finite) reduced system with generator $\mathbf{A}_J$. This result is the basis for an algorithm to compute the probability density function with guaranteed accuracy. The FSP approach and various improvements on the main algorithm can be found in [40,41].

21.4 Parameter Identification

Microscopy techniques and fluorescence activated cell sorting (FACS) technology enable single cell measurement of cellular species to be carried out for large numbers of cells. This raises the prospect of using statistical quantities such as moments and variances, measured at different instants in time, to identify model parameters. Here we demonstrate these ideas through a simple description of gene transcription

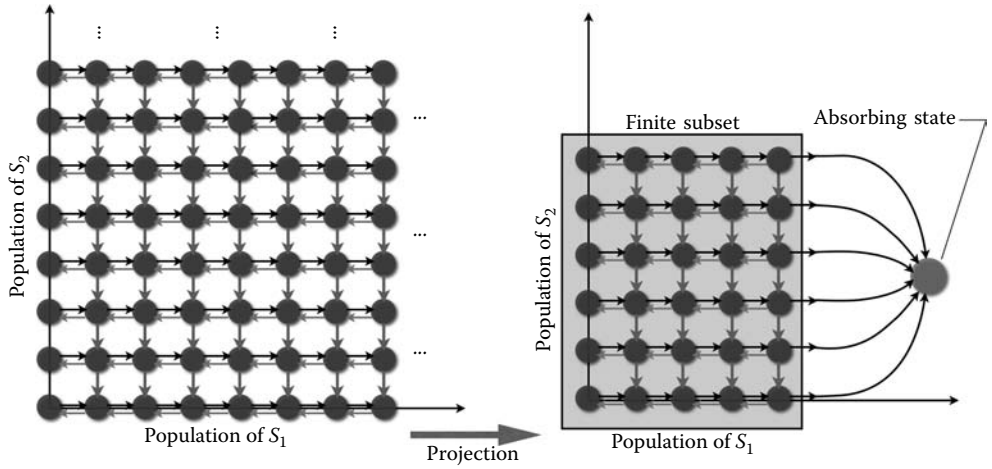

FIGURE 21.4 The finite state projection. Left panel shows the state space for a system with two species. Arrows indicate possible transitions within states. The corresponding process is a continuous-time discrete state Markov process whose state space is typically very large or infinite. Right panel shows the projected system for a specific projection region (gray box). The projected system is obtained as follows: transitions within the projection region are kept unchanged. Transitions that emanate from states within the region and end at states outside (in the original system) are routed to a single absorbing state in the projected system. Transitions into the projection region are deleted. As a result, the projected system is a finite state Markov process, and the probability of each state can be computed exactly.

and translation. Let x denote the population of mRNA molecules, and let y denote the population of proteins in a cell. The system population is assumed to change only through four reactions:

$$\begin{aligned} \emptyset &\rightarrow mRNA \\ mRNA &\rightarrow \emptyset \\ mRNA &\rightarrow mRNA + protein \\ protein &\rightarrow \emptyset \end{aligned}$$

for which the propensity functions, $w_i(x, y)$, are

$$w_1(x,y) = k_1 + k_{21}y; \quad w_2(x,y) = \gamma_1 x;$$

$$w_3(x,y) = k_2 x; \quad w_2(x,y) = \gamma_2 y.$$

Here, the terms k_i and γ_i are production and degradation rates, respectively, and k_{21} corresponds to a feedback effect that the protein is assumed to have on the transcription process. In positive feedback, $k_{21} > 0$, the protein increases transcription; in negative feedback, $k_{21} < 0$, the protein inhibits transcription.

The various components of the first two moments, $\mathbf{v}(t) := \left[\mathbb{E}\{x\} \quad \mathbb{E}\{x^2\} \quad \mathbb{E}\{y\} \quad \mathbb{E}\{y^2\} \quad \mathbb{E}\{xy\}\right]^T$, evolve according to the linear time invariant system:

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \mathbb{E}\{x\} \\ \mathbb{E}\{x^2\} \\ \mathbb{E}\{y\} \\ \mathbb{E}\{y^2\} \\ \mathbb{E}\{xy\} \end{bmatrix} &= \begin{bmatrix} -\gamma_1 & 0 & k_{21} & 0 & 0 \\ \gamma_1 + 2k_1 & -2\gamma_1 & k_{21} & 0 & 2k_{21} \\ k_2 & 0 & -\gamma_2 & 0 & 0 \\ k_2 & 0 & \gamma_2 & -2\gamma_2 & 2k_2 \\ 0 & k_2 & k_1 & k_{21} & -\gamma_1 - \gamma_2 \end{bmatrix} \begin{bmatrix} \mathbb{E}\{x\} \\ \mathbb{E}\{x^2\} \\ \mathbb{E}\{y\} \\ \mathbb{E}\{y^2\} \\ \mathbb{E}\{xy\} \end{bmatrix} + \begin{bmatrix} k_1 \\ k_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ &= \mathbf{A}\mathbf{v} + \mathbf{b} \end{aligned} \tag{21.3}$$

Now that we have expressions for the dynamics of the first two moments, they can be used to identify the various parameters: $[k_1, \gamma_1, k_2, \gamma_2, k_{21}]$ from properly chosen data sets. We will next show how this can be done for transcription parameters k_1 and γ_1. For a discussion on identification of the full set, we refer the reader to [22,41,42].

21.4.1 Identifying Transcription Parameters

We begin by considering a simpler birth–death process of mRNA transcripts, whose populations are denoted by x. The moment equation for this system is:

$$\frac{d}{dt}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -\gamma & 0 \\ \gamma + 2k & -2\gamma \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} k \\ k \end{bmatrix},$$

where we have dropped the subscripts on k_1 and γ_1. By applying the nonlinear transformation:

$$\begin{bmatrix} \mu \\ \sigma^2 - \mu \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 - v_1^2 - v_1 \end{bmatrix},$$

where μ and σ^2 refer to the mean and variance of x, respectively, we arrive at the transformed set of equations:

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} \mu \\ \sigma^2 - \mu \end{bmatrix} &= \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 - 2v_1\dot{v}_1 - \dot{v}_1 \end{bmatrix} \\ &= \begin{bmatrix} -\gamma_1 v_1 + k \\ (\gamma_1 + 2k)v_1 - 2\gamma v_2 + k - (2v_1 + 1)(-\gamma v_1 + k) \end{bmatrix} \\ &= \begin{bmatrix} -\gamma & 0 \\ 0 & -2\gamma \end{bmatrix}\begin{bmatrix} \mu \\ \sigma^2 - \mu \end{bmatrix} + \begin{bmatrix} k \\ 0 \end{bmatrix}. \end{aligned} \tag{21.4}$$

Suppose that μ and σ^2 are known at two instances in time, t_0 and $t_1 = t_0 + \tau$, and denote their values at time t_i as μ_i and σ_i^2, respectively. The relationship between (μ_0, σ_0^2) and (μ_1, σ_1^2) is governed by the solution of 21.4, which can be written:

$$\begin{bmatrix} \mu_1 \\ \sigma_1^2 - \mu_1 \end{bmatrix} = \begin{bmatrix} \exp(-\gamma\tau)\mu_0 \\ \exp(-2\gamma\tau)(\sigma_0^2 - \mu_0) \end{bmatrix} + \begin{bmatrix} \frac{k}{\gamma}(1 - \exp(-\gamma\tau)) \\ 0 \end{bmatrix} \tag{21.5}$$

In this expression there are two unknown parameters, γ and k, that we wish to identify from the data $\{\mu_0, \sigma_0^2, \mu_1, \sigma_1^2\}$. If $\mu_0 = \sigma_0^2$, the second equation is trivial, and we are left with only one equation whose solution could be any pair:

$$\left(\gamma,\ k = \gamma\frac{\mu_1 - \exp(-\gamma\tau)\mu_0}{1 - \exp(-\gamma\tau)}\right).$$

If for the first measurement $\mu_0 \neq \sigma_0^2$ and for the second measurement $\mu_1 \neq \sigma_1^2$, then we can solve for:

$$\gamma = -\frac{1}{2t}\log\left(\frac{\sigma_1^2 - \mu_1}{\sigma_0^2 - \mu_0}\right)$$

$$k = \gamma\frac{\mu_1 - \exp(-\gamma t)\mu_0}{1 - \exp(-\gamma\tau)}.$$

Note that if μ_1 and σ_1^2 are very close, the sensitivity of γ to small errors in this difference becomes very large. From Equation 21.5, one can see that as τ becomes very large, $(\sigma_1^2 - \mu_1)$ approaches zero, and *steady-state measurements do not suffice to uniquely identify both parameters.*

21.5 Examples

To illustrate the above methods, we consider the synthetic self-regulated genetic system as illustrated in Figure. 21.5. The *lac* operon controls the production of the LacI protein, which in turn tetramerizes and represses its own production. The *lac* operon is assumed to be present in only a single copy within each cell and is assumed to have two possible state: g_{ON} and g_{OFF}, which are characterized by whether or not a LacI tertramer, LacI_4, is bound to the operon. In all, the model is described with seven reactions:

Reaction#	Reaction Description	Propensity Function
$R1:$	$4\text{LacI} \rightarrow \text{LacI}_4$	$w_1 = k_1 \binom{[\text{LacI}]}{4}$
$R2:$	$\text{LacI}_4 \rightarrow 4\text{LacI}$	$w_2 = k_2[\text{LacI}_4]$
$R3:$	$g_{ON} + \text{LacI}_4 \rightarrow g_{OFF}$	$w_3 = k_3[g_{ON}][\text{LacI}_4]$
$R4:$	$g_{OFF} \rightarrow \text{LacI}_4 + g_{ON}$	$w_4 = k_4[g_{OFF}]$
$R5:$	$g_{ON} \rightarrow g_{ON} + \text{LacI}$	$w_5 = k_5[g_{ON}]$
$R6:$	$\text{LacI} \rightarrow \phi$	$w_6 = k_6[\text{LacI}]$
$R7:$	$\text{LacI}_4 \rightarrow \phi$	$w_7 = k_7[\text{LacI}_4]$

(21.6)

The first of these reactions corresponds to the combination of four individual monomers to form a tetramer—the rate of this reaction depends upon the total number of possible combinations of four different molecules, which is given by the binomial

$$\binom{[\text{LacI}]}{4} = [\text{LacI}] \cdot ([\text{LacI}] - 1) \cdot ([\text{LacI}] - 2) \cdot ([\text{LacI}] - 3)/24,$$

and the second reaction corresponds to the reverse of the tetramerization event. The next two reactions characterize the ON-to-OFF and OFF-to-ON switches that occur when a tetramer binds to or unbinds

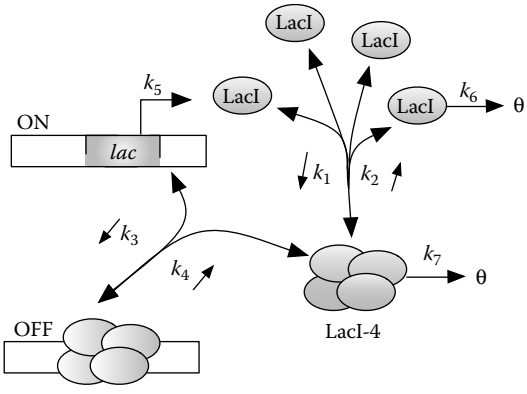

FIGURE 21.5 Schematic representation of a synthetic self-regulated genetic network. In the model, four LacI monomers (represented as ovals) can bind reversibly to form tetramers (represented as clusters of four ovals). The *lac* operon has two states: OFF when LacI tetramers are bound to the gene and blocking the transcription start site, and ON when LacI tetramers are not bound to the gene. Both LacI monomers and tetramers can degrade. See also reactions listed in Equation 21.6.

from the operon, respectively. When the gene is in the ON state, the fifth reaction can occur and LacI monomers are created with an exponentially distributed waiting times. Finally, reactions R6 and R7 correspond to the usual linear decay of the monomers and tetramers, respectively.

For the analysis of this process, we first define the stoichiometry and reaction rate vectors for the process as:

$$\mathbf{S} = \begin{bmatrix} -4 & 4 & 0 & 0 & 1 & -1 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}, \quad \text{and} \tag{21.7}$$

$$\mathbf{w}(\mathbf{x}) = \begin{bmatrix} k_1 \binom{x_1}{4} \\ k_2 x_2 \\ w_3 = k_3 x_3 x_2 \\ w_4 = k_4 x_4 \\ w_5 = k_5 x_3 \\ w_6 = k_6 x_1 \\ w_7 = k_7 x_2 \end{bmatrix}. \tag{21.8}$$

In what follows, we will take many different approaches to analyzing this system. In order to compare each method, we make the assumption that the volume is unity $\Omega = 1$, such that we can avoid parameter scaling issues when moving between reaction rate equations and the stochastic description. We consider the following parameter set for the reaction rates:

$$k_1 = 1/30\ \mathrm{N}^{-4}\,\mathrm{s}^{-1} \quad k_2 = 0.002\ \mathrm{N}^{-1}\,\mathrm{s}^{-1} \quad k_3 = 0.01\ \mathrm{N}^{-2}\,\mathrm{s}^{-1}$$
$$k_4 = 0.2\ \mathrm{N}^{-1}\,\mathrm{s}^{-1} \quad k_5 = 20\ \mathrm{N}^{-1}\,\mathrm{s}^{-1} \quad k_6 = 0.1\ \mathrm{N}^{-1}\,\mathrm{s}^{-1}$$
$$k_7 = 0.1\ \mathrm{N}^{-1}\,\mathrm{s}^{-1},$$

and we assume that the process begins with the gene in the active state and no LacI is present in the system:

$$\mathbf{x}(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

21.5.1 Deterministic (Reaction Rate) Analysis

As a first analysis, let us consider the deterministic reaction rate equations that are described by these four interacting chemical species and their seven reactions. For this case, one can write the reaction rate equations as:

$$\dot{\mathbf{x}}(t) = \mathbf{S}\mathbf{w}(\mathbf{x}(t))$$

or in the usual notation of ODEs

$$\dot{x}_1 = -(4/24)k_1 x_1(x_1 - 1)(x_1 - 2)(x_1 - 3) + 4k_2 x_2 + k_5 x_3 - k_6 x_1,$$

$$\dot{x}_2 = (4/24)k_1 x_1(x_1 - 1)(x_1 - 2)(x_1 - 3) - k_2 x_2 - k_3 x_2 x_3 - k_7 x_2,$$

$$\dot{x}_3 = -k_3 x_2 x_3 + k_4 x_4,$$

$$\dot{x}_4 = k_3 x_2 x_3 - k_4 x_4.$$

We note that the first reaction only makes sense when the $x_1 \geq 4$ corresponding to when there are at least four molecules of the monomer present and able to combine. In the case where there are fewer than four molecules, we must use a different set of equations:

$$\dot{x}_1 = 4k_2x_2 + k_5x_3 - k_6x_1,$$
$$\dot{x}_2 = -k_2x_2 - k_3x_2x_3 - k_7x_2,$$
$$\dot{x}_3 = -k_3x_2x_3 + k_4x_4,$$
$$\dot{x}_4 = k_3x_2x_3 - k_4x_4.$$

These equations have been integrated over time and the responses of the dynamical process are shown in the solid gray lines of Figure 21.7. We note that were one to use the LNA, the computed mean value for the process would be exactly the same as the solutions shown with the solid gray line.

21.5.2 Stochastic Simulations

The reactions listed above can also be simulated using Gillespie's SSA [25]. Two such simulations shown in Figure 21.6 illustrate the large amount of stochastic variability inherent in the model. By simulating the system 5000 times, one can collect the statistics of these variations and record them as functions of time. The dynamics of the mean levels of each species is shown by the solid, but somewhat jagged, black lines in Figure 21.7. Furthermore, one can collect statistics on the number of monomers and tetramers at different points in time and plot the resulting histograms to show their marginal distributions as illustrated in Figures 21.8 and 21.9. From these plots, it is noticeable that the deterministic reaction rate equations and the mean of the stochastic process are not equivalent for this process. This discrepancy arises from the nonlinearity of the propensity functions for the the first and third reactions.

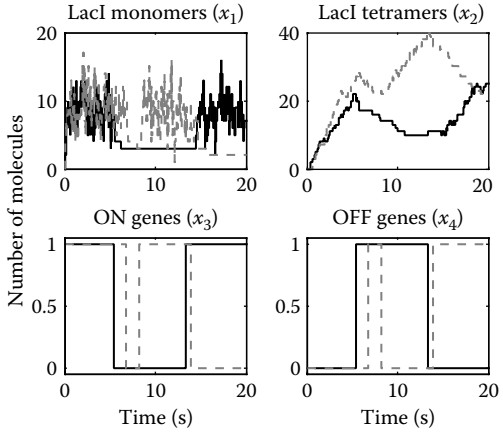

FIGURE 21.6 Results of two stochastic simulations (solid black, dashed gray) of the self-repressing LacI synthetic gene regulatory network. The top left panel corresponds to the populations of LacI monomers; the top right panel corresponds to the population of LacI tetramers; the bottom left corresponds to the population of ON genes; and the bottom right panel corresponds to the population of OFF genes.

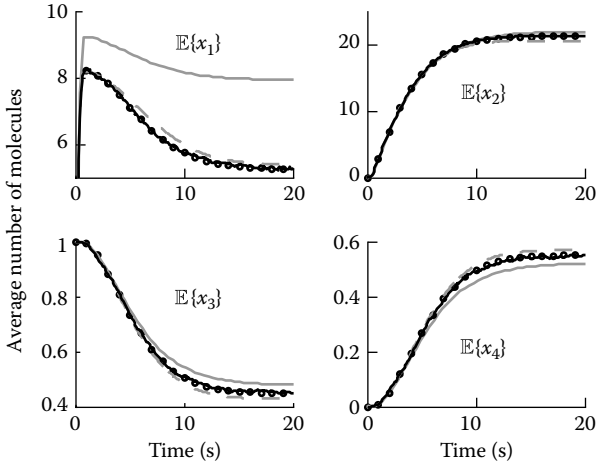

FIGURE 21.7 Dynamics of the mean values of **x** as found using different solution schemes. The solid gray lines correspond to the solution of the deterministic reaction rate equations. The dashed gray lines correspond to the solution using moment closure based upon the assumption of a multivariated Gaussian distribution. The jagged black lines correspond to the solution of 5000 stochastic simulations. The dotted lines correspond to the solution with the FSP approach.

21.5.3 Normal Moment Closures

Above we have derived the differential equation for the mean of the process to be:

$$\frac{d}{dt}\mathbb{E}(\mathbf{X}) = \mathbf{S}\mathbb{E}\{\mathbf{w}(\mathbf{X})\} \tag{21.9}$$

In the case of linear propensity functions, then the average propensity function is simply the propensity function of the average population, and we could substitute:

$$\mathbf{S}\mathbb{E}\{\mathbf{w}(\mathbf{X})\} = \mathbf{S}\mathbf{w}(\mathbb{E}\{\mathbf{X}\}), \quad \text{for affine linear } \mathbf{w}(\mathbf{X}).$$

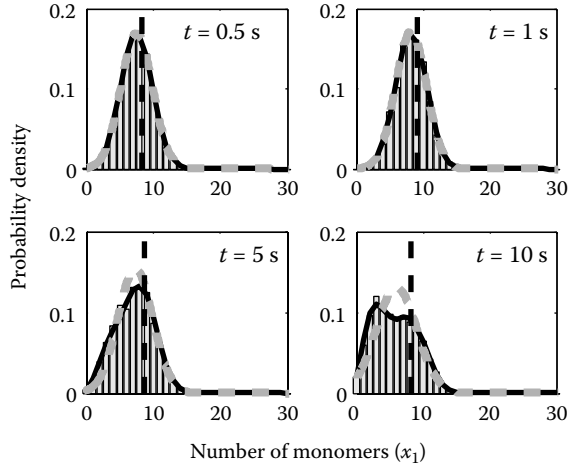

FIGURE 21.8 Probability distribution for the number of LacI monomers (x_1) at different points in time. The grey histograms have been found using 5000 stochastic simulations. The solid black lines correspond to the FSP solution. The dashed black lines show the prediction of the mean using the deterministic reaction rate equations, and the dashed gray lines show the results of the moment closure approach with an assumption of a normal distribution.

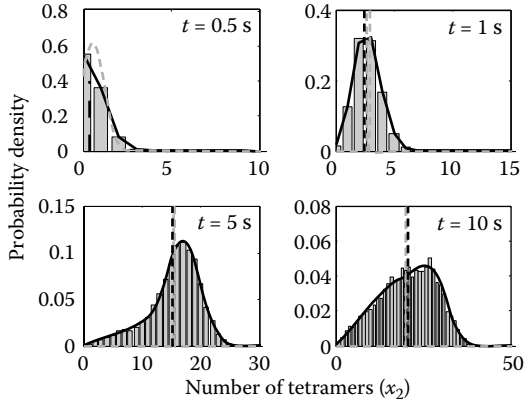

FIGURE 21.9 Probability distribution for the number of LacI tetramers (x_2) at different points in time. The grey histograms have been found using 5000 stochastic simulations. The solid black lines correspond to the FSP solution. The dashed black lines show the prediction of the mean using the deterministic reaction rate equations, and the dashed gray line shows the results of the moment closure approach with an assumption of a normal distribution.

However, when the propensity function are nonlinear, this substitution is incorrect, and in our case we have:

$$\mathbb{E}\{\mathbf{w}(\mathbf{X})\} = \mathbb{E}\left\{\begin{bmatrix} k_1\binom{x_1}{4} \\ k_2x_2 \\ k_3x_3x_2 \\ k_4x_4 \\ k_5x_3 \\ k_6x_1 \\ k_7x_2 \end{bmatrix}\right\} = \begin{bmatrix} k_1/24\left(\mathbb{E}\{x_1^4\} - 6\mathbb{E}\{x_1^3\} + 11\mathbb{E}\{x_1^2\} - 6\mathbb{E}\{x_1\}\right) \\ k_2\mathbb{E}\{x_2\} \\ k_3\mathbb{E}\{x_3x_2\} \\ k_4\mathbb{E}\{x_4\} \\ k_5\mathbb{E}\{x_3\} \\ k_6\mathbb{E}\{x_1\} \\ k_7\mathbb{E}\{x_2\} \end{bmatrix}.$$

Thus, we find that the expected values depend upon higher order moments, and the equations are not closed to a finite set. Similarly, the ODEs that describe the evolution of the second moments are given by:

$$\frac{d}{dt}\mathbb{E}\{\mathbf{X}\mathbf{X}^T\} = \mathbf{S}\mathbb{E}\{\mathbf{w}(\mathbf{X})\mathbf{X}^T\} + \mathbb{E}\{\mathbf{w}(\mathbf{X})\mathbf{X}^T\}^T\mathbf{S}^T + \mathbf{S}\left\{\text{diag}(\mathbb{E}\{\mathbf{w}(\mathbf{X})\})\right\}\mathbf{S}^T, \tag{21.10}$$

where the matrix $\mathbf{w}(\mathbf{x})\mathbf{x}^T$ is

$$\mathbf{w}(\mathbf{X})\mathbf{X}^T = \begin{bmatrix} k_1\binom{x_1}{4}x_1 & k_1\binom{x_1}{4}x_2 & k_1\binom{x_1}{4}x_3 & k_1\binom{x_1}{4}x_4 \\ k_2x_1x_2 & k_2x_2^2 & k_2x_2x_3 & k_2x_2x_4 \\ k_3x_1x_2x_3 & k_3x_2^2x_3 & k_3x_2x_3^2 & k_3x_2x_3x_4 \\ k_4x_1x_4 & k_4x_2x_4 & k_4x_3x_4 & k_4x_4^2 \\ k_5x_1x_3 & k_5x_2x_3 & k_5x_3^2 & k_5x_3x_4 \\ k_6x_1^2 & k_6x_1x_2 & k_6x_1x_3 & k_6x_1x_4 \\ k_7x_1x_2 & k_7x_2^2 & k_7x_2x_3 & k_7x_2x_4 \end{bmatrix},$$

In this case we see that the second moment also depends upon higher order moments. In particular the second moment of x_1 now depends upon the fifth uncentered moment of x_1. This relationship will continue for every higher moment such that the nth moment will always depend upon the $(n+3)$th order moment for this system.

If we make the assumption that the joint distribution of all species are given by a multivariate normal distribution, then we can use this relationship to close the moment equations. Perhaps the easiest way to

find these relationships is to use the moment generating function (MGF) approach. We define the MGF as:

$$M_{\mathbf{x}}(\mathbf{t}) = \exp\left(\mu^T \mathbf{t} + 1/2\mathbf{t}^T \Sigma \mathbf{t}\right),$$

where the vectors are defined as:

$$\begin{aligned}
\mu &= \mathbb{E}\{\mathbf{x}\},\\
\Sigma &= \mathbb{E}\{(\mathbf{x}-\mu)(\mathbf{X}-\mu)^T\}\\
&= \mathbb{E}\{\mathbf{X}\mathbf{X}^T\} - \mathbb{E}\{\mathbf{X}\}\mathbb{E}\{\mathbf{x}^T\}, \text{ and}\\
\mathbf{t} &= [t_1, t_2, t_3, t_4]^T.
\end{aligned}$$

With this definition, one can write any uncentered moment in terms of μ and Σ as follows:

$$\mathbb{E}\{x_1^{n_1} \ldots x_4^{n_4}\} = \frac{d^{n_1+\cdots+n_4}}{dx_1^{n_1} \ldots dx_4^{n_4}} M_x(\mathbf{t})\bigg|_{\mathbf{t}=\mathbf{0}}.$$

For example, the fifth uncentered moment of x_1 is given by:

$$\begin{aligned}
\mathbb{E}\{x_1^5\} &= \frac{d^5}{dx_1^5} M_x(\mathbf{t})\bigg|_{\mathbf{t}=\mathbf{0}}\\
&= 15\mathbb{E}\{x_1\}\mathbb{E}\{x_1^2\}^2 - 20\mathbb{E}\{x_1\}^3\mathbb{E}\{x_1^2\} + 6\mathbb{E}\{x_1\}^5.
\end{aligned}$$

Such an expression can be found for each moment of order three or higher in Equations 21.9 and 21.10. As a result the approximated distribution is fully described in terms of the first and second moments, which are our new set of 14 dynamic variables:

$$\begin{aligned}
&\mathbb{E}\{x_1\}, \mathbb{E}\{x_2\}, \mathbb{E}\{x_3\}, \mathbb{E}\{x_4\},\\
&\mathbb{E}\{x_1^2\}\mathbb{E}\{x_1x_2\}, \mathbb{E}\{x_1x_3\}, \mathbb{E}\{x_1x_4\},\\
&\mathbb{E}\{x_2^2\}, \mathbb{E}\{x_2x_3\}, \mathbb{E}\{x_2x_4\},\\
&\mathbb{E}\{x_3^2\}, \mathbb{E}\{x_3x_4\}, \mathbb{E}\{x_4^2\}.
\end{aligned} \tag{21.11}$$

We note that because there is only a single gene then x_3 and x_4 are mutually exclusive and take values of either zero or one. As a result, we can specify algebraic constraints on the last three of the moments listed in Equation 21.11 as:

$$\begin{aligned}
\mathbb{E}\{x_3^2\} &= \mathbb{E}\{x_3\},\\
\mathbb{E}\{x_4^2\} &= \mathbb{E}\{x_4\},\\
\mathbb{E}\{x_3x_4\} &= 0
\end{aligned}$$

and thus we are left with only 11 ODEs.

We have solved the nonlinear ODE's resulting from the moment closure, and the results for the mean values of each species are represented by the gray dashed lines in Figure 21.7. From the figure, we see that for this case, the use of the coupled first and second moments results in a much better approximation of the of the mean behavior than did the deterministic reaction rate equation (compare solid and dashed gray lines in Figure 21.7).

By including some description of the second uncentered moment of the process, the moment closure does a much better job of capturing the mean behavior of the process as can be seen by Figure 21.7. Furthermore, closer examination reveals that the second moment for the population of monomers is also well captured by this approximation as is seen in Figure 21.8. However, it is clear that the actual

distributions are not Gaussian, and truncating away the higher-order moments has introduced significant errors. This can be seen first in the monomer distributions at $t = 10$ s, where the actual distribution appears to be almost bimodal. An even worse approximation is obtained for the tetramer distribution as is shown in Figure 21.9, where the solution of the moment closure equations actually produces a physically unrealizable result of negative variance for the tetramer distribution. This failure is not unexpected due to the fact that the dynamics of the tetramer population depend strongly on the approximated high-order moments of the monomer population.

21.5.4 FSP Analysis

In general the master equation can be written in the form $\mathbf{P}(t) = \mathbf{A}\mathbf{P}(t)$, where the infinitesimal generator $\mathbf{A}$ is defined as:

$$A_{i_2 i_1} = \begin{cases} -\sum_{k=1}^{M} w_k(\mathbf{x}_{i_1}) & \text{for } i_1 = i_2 \\ w_k(\mathbf{x}_{i_1}) & \text{for } \mathbf{x}_{i_2} = \mathbf{x}_{i_1} + \mathbf{s}_k \\ 0 & \text{otherwise} \end{cases}$$

However, in order for this notation to make sense, one first has to define the enumeration of all the possible states $\{\mathbf{x}\}$. Based upon a few runs of the SSA, we can restrict our attention to a finite region of the state space ($x_1 \leq N_1 = 30$ and $x_2 \leq N_2 = 55$), then we can use the following scheme:

$$i(\mathbf{x}) = x_4(N_1 + 1)(N_2 + 1) + x_1(N_2 + 1) + x_2 + 1.$$

Note that we can make this enumeration depend only on x_1, x_2 and x_4 due to the fact that x_3 and x_4 are mutually exclusive and $x_3 = 1 - x_4$.

The FSP analysis has been conducted, and the black dotted lines in Figure 21.7 show the mean value of each of the four species as functions of time. With the chosen projection, the total one norm error in the computed probability distribution is guaranteed to be 4.8×10^{-5} or less at every instant in time. As such, the FSP solution makes a good basis to compare the other solution schemes. With the FSP solution we can also determine not just the mean but the entire probability distribution at each time point, and the marginal distributions of the monomers (x_1) and the tetramers (x_2) are shown at times t = {0.5, 1, 5, 10}s in Figures 21.8 and 21.9.

Acknowledgments

The authors acknowledge support by the National Science Foundation under grants ECCS-0835847 and ECCS-0802008, the Institute for Collaborative Biotechnologies through Grant DAAD19-03-D-0004 from the US Army Research Office, and Los Alamos LDRD funding.

References

1. M. Elowitz, A. Levine, E. Siggia, and P. Swain. Stochastic gene expression in a single cell. *Nature*, 297(5584):1183–1186, 2002.
2. P. Swain, M. Elowitz, and E. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proceedings of the National Academy of Sciences*, USA, 99(20):12795–12800, 2002.
3. H. H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences USA*, 94(3):814–819, 1997.
4. H. H. McAdams and A. Arkin. It's a noisy business! genetic regulation at the nanomolar scale. *Trends in Genetics*, 15(2):65–69, February 1999.
5. B. Munsky, Hernday, D. Low, and Khammash. Stochastic modeling of the pap pili epigenetic switch. In *Foundations of Systems Biology in Engineering*, August 2005.
6. T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403:339–342, 2000.

7. J. Hasty, D. McMillen, and J. J. Collins. Engineered gene circuits. *Nature*, 420(6912):224–230, 2002.
8. M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.
9. M. Yoda, T. Ushikubo, W. Inoue, and M. Sasai. Roles of noise in single and coupled multiple genetic oscillators. *Journal of Chemical Physics*, 126:115101, 2007.
10. J. Paulsson, O. Berg, and M. Ehrenberg. Stochastic focusing: Fluctuation-enhanced sensitivity of intracellular regulation. *Proceedings of the National Academy of Sciences*, 97:7148–7153, 2000.
11. M. Thattai and A. Van Oudenaarden. Intrinsic noise in gene regulatory networks. *Proceedings of the National Academy of Sciences*, 98:8614–8619, 2001.
12. M. Thattai and A. Van Oudenaarden. Attenuation of noise in ultrasensitive signaling cascades. *Biophysics Journal*, 82:2943–2950, 2002.
13. N. Rosenfeld, M. Elowitz, and U. Alon. Negative autoregulation speeds the response times of transcription networks. *Journal of Molecular Biology*, 323:785–793, 2002.
14. F. J. Isaacs, J. Hasty, C. R. Cantor, and J. J. Collins. Prediction and measurement of an autoregulatory genetic module. *Proceedings of the National Academy of Sciences, USA*, 100:7714–7719, 2003.
15. P. S. Swain. Efficient attenuation of stochasticity in gene expression through post-transcriptional control. *Journal of Molecular Biology*, 344:965–976, 2004.
16. H. El-Samad and M. Khammash. Stochastic stability and its applications to the study of gene regulatory networks. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, 3: 3001–3006, December 2004.
17. J. M. Pedraza and A. van Oudenaarden. Noise propoagation in gene networks. *Science*, 307(5717):1965 – 1969, March 2005.
18. J. Paulsson. Summing up the noise in gene networks. *Nature*, 427(6973):415–418, 2004.
19. M. Khammash and H. El-Samad. Stochastic modeling and analysis of genetic networks. In *Proceedings of the 44th IEEE Conference on Decision and Control and 2005 European Control Conference*, pp. 2320–2325, 2005.
20. H. El-Samad and M. Khammash. Regulated degradation is a mechanism for suppressing stochastic fluctuations in gene regulatory networks. *Biophysical Journal*, 90:3749–3761, 2006.
21. M. Khammash. *Control Theory in Systems Biology*, Chapter 2. MIT Press, Cambridge, MA, 2009.
22. B. Munsky and M. Khammash. Using noise transmission properties to identify stochastic gene regulatory networks. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 768–773, December 2008.
23. S. N. Ethier and T. G. Kurtz. *Markov Processes Characterization and Convergence*. Wiley Series in Probability and Statistics, 1986.
24. D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics*, 115:1716–1733, 2001.
25. D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.
26. T. G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and their Applications*, 6:223–240, 1978.
27. D. T. Gillespie. The chemical Langevin and Fokker–Planck equations for the reversible isomerization reaction. *Journal of Physical Chemistry*, 106:5063–5071, 2002.
28. N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier Science, North Holland, 2007.
29. J. Elf and M. Ehrenberg. Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Research*, 13:2475–2484, 2003.
30. R. Tomioka, H. Kimura, T. J. Koboyashi, and K. Aihara. Multivariate analysis of noise in genetic regulatory networks. *Journal of Theoretical Biology*, 229(3):501–521, 2004.
31. P. Whittle. On the use of the normal approximation in the treatment of stochastic processes. *Journal of Royal Statistical Society, Series B*, 19:268–281, 1957.
32. I. Nasell. Moment closure and the stochastic logistic model. *Theoretical Population Biology*, 63:159–168, 2003.
33. C. A. Gomez-Uribe and G. C. Verghese. Mass fluctuation kinetics: Capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. *Journal of Chemical Physics*, 126(2):024109–024109–12, 2007.
34. M. J. Keeling. Multiplicative moments and measures of persistence in ecology. *Journal of Theoretical Biology*, 205:269–281, 2000.
35. A. Singh and J. P. Hespanha. A derivative matching approach to moment closure for the stochastic logistic model. *Bulletin of Mathematical Biology*, 69:1909–1025, 2007.

36. I. Nasell. An extension of the moment closure method. *Theoretical Population Biology*, 64:233–239, 2003.
37. J. P. Hespanha. Polynomial stochastic hybrid systems. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, Vol. 3414, pp. 322–338. Springer-Verlag, Berlin, March 2005.
38. B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *Journal of Chemical Physics*, 124:044104, 2006.
39. S. Peles, B. Munsky, and M. Khammash. Reduction and solution of the chemical master equation using time scale separation and finite state projection. *Journal of Chemical Physics*, 20:204104, November 2006.
40. B. Munsky and M. Khammash. The finite state projection approach for the analysis of stochastic noise in gene networks. *IEEE Transactions on Automatic Control*, 53:201–214, January 2008.
41. B. Munsky. The finite state projection approach for the solution of the master equation and its application to stochastic gene regulatory networks. PhD thesis, University of California, Santa Barbara, 2008.
42. B. Munsky, B. Trinh, and M. Khammash. Listening to the noise: Random fluctuations reveal gene network parameters. *Molecular Systems Biology*, 5(318), 2009.

22

Modeling the Human Body as a Dynamical System: Applications to Drug Discovery and Development

M. Vidyasagar
The University of Texas at Dallas

22.1 Introduction

The objective of this article is to introduce the reader to some aspects of drug discovery where system theory can potentially play a useful role. Specifically, attention is focused on modeling the human body as a dynamical system, so that the action of a drug (both beneficial as well as unwanted) can possibly be predicted in a systematic fashion. We begin by describing the current serious, almost crisis-like, situation in drug discovery. Then we describe two out of the many success stories of physiological modeling, namely the glucose–insulin control system in diabetics, and the control of infection among HIV patients. Then we conclude with a brief description of how probabilistic methods can be used to model/predict toxicity. The overall message is that, given the current state of knowledge of human physiology, it is quite reasonable for control and system theorists to aspire to play a significant role in drug discovery and development.

22.2 The Crisis in Drug Discovery

In the electronics industry, the famous "Moore's law" states that the cost of computing goes down by a factor of 2, while the speed of computing goes up by a factor of 2, over every 18-month period. This remarkable rate of growth has been maintained at a more or less constant rate for nearly three decades.

TABLE 22.1 Cost of Discovering a Drug

Year	Cost (in million dollars)
1975	138
1987	318
2001	802
2006	1318

Indeed, the pervasiveness of computation (in all its forms) in day-to-day life is directly attributable to Moore's law.

The contrast with the situation in the pharmaceutical industry could not be more stark. Table 22.1 shows how the cost of discovering a new drug has gone up over the years [1]. It can be seen that the cost of discovering a new drug has gone up roughly 10 times over a 30-year period, with the maximum rise during the past decade. Indeed, the rate of increase in the cost of discovering a new drug is far higher than the rate of overall inflation.

If one were to consider the cost of research and development, a similar picture emerges. Table 22.2 shows the R&D spending (in billion dollars) by members of PhRMA (Pharmaceutical Research and Manufacturers of America) and the total industry (including Europe).

Figure 22.1 displays the relentlessly rising trend in R&D expenditure during the decade 1996–2006.

And yet, despite this nearly 20-fold increase in R&D expenditure, there has not been a commensurate increase in the number of new drugs approved. Figure 22.2 illustrates the situation in a very dramatic fashion.

Specifically, though R&D expenditures have grown steadily at around 6% annually over the past decade or more, the number of new drugs approved *has actually declined* over the same period. The time taken to bring a new drug to market has increased by roughly 50% over the same period. Given that a new drug molecule enjoys patent protection for 20 years, the increase in the time taken to market a new drug results in a corresponding *decrease* in the amount of time that the inventor has to profit from the patent. Paradoxically, it can be seen from the same figure that the *total sales* of the pharmaceutical industry have outstripped even the galloping increases in R&D expenditure. This is reflected in the ever-increasing fraction of a nation's GDP that is devoted to health care. The figures for the United States are nothing short of dramatic. Table 22.3 shows the GDP of the United States, the national health expenditure, and the fraction of the GDP that went into health care, from 1960 until 2008, the last year for which figures are available.

It is sobering to realize that the national health expenditure in 2008 is roughly equal to the entire US GDP in 1980. The same Web site projects a continued increase in the proportion of GDP that goes for health expenditure. Thus, even though the pharmaceutical industry has not been very successful in

TABLE 22.2 R&D Spending (in billion dollars)

Year	PhRMA Members	Total Industry
1980	2.0	N/A
1990	8.4	N/A
2000	26.0	N/A
2004	37.0	47.6
2005	39.9	51.8
2006	47.9	63.2

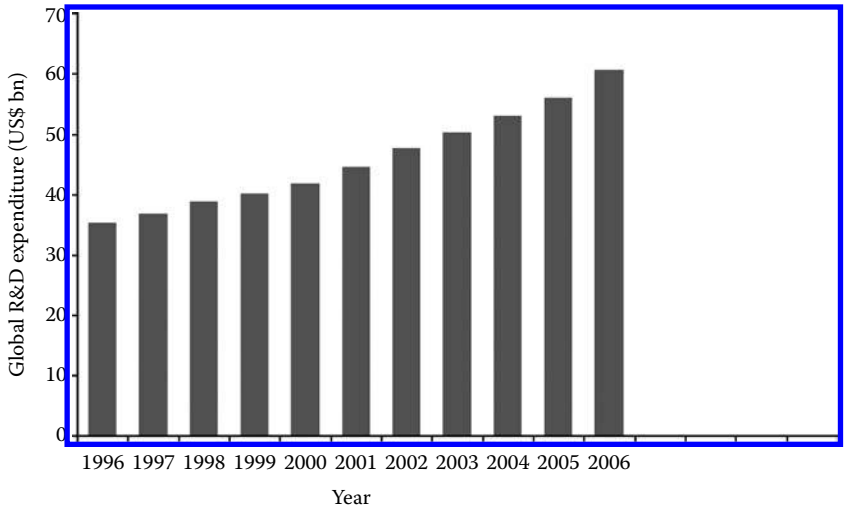

FIGURE 22.1 R&D expenditures, 1996–2006. (From CMR International Performance Metrics Program © Thomson Reuters, 2008.)

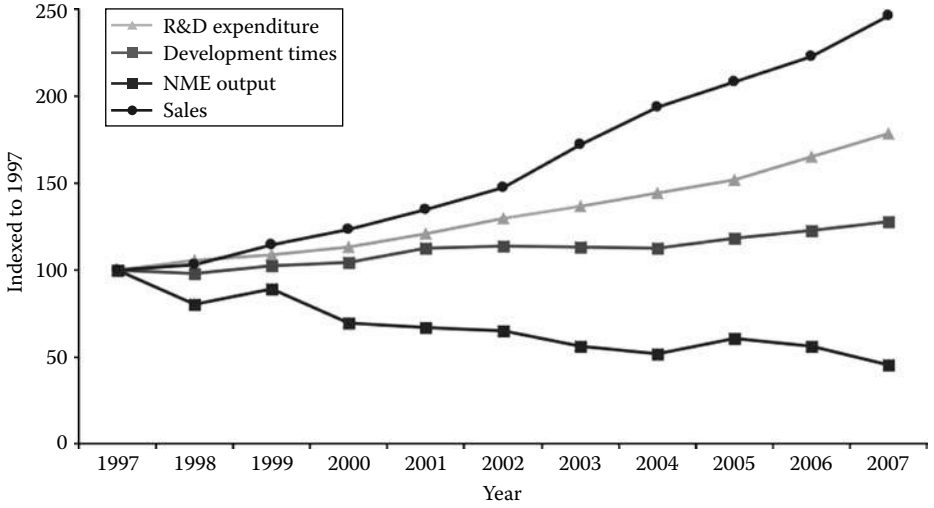

FIGURE 22.2 Global R&D expenditure, development times, global pharmaceutical sales and new molecular entity output 1997–2007. (From CMR International Performance Metrics Program © Thomson Reuters & IMS Health, 2008.)

discovering new drugs, it has been quite successful in passing on its expenses to society at large. It is a moot point as to how long such a situation can continue unchecked.

22.3 Systematic Approaches to Drug Discovery

By and large, the reason for the failure of the pharmaceutical industry to discover new drugs is the reliance on an outdated methodology. The basics of the drug discovery process are by now well-established. The human body has around 100,000 proteins, produced by around 30,000 genes. Thus the same gene can

TABLE 22.3 Fraction of US GDP Going toward National Health Expenditure

Year	NHE $	US GDP $	NHE/GDP %
1960	27.5	526	5.2
1970	74.9	1038	7.2
1980	253.4	2788	9.1
1990	714.2	5801	12.3
2000	1352.9	9952	13.6
2005	1982.5	12,638	15.7
2006	2112.5	13,399	15.8
2007	2239.7	14,078	15.9
2008	2388.7	14,441	16.5

Source: Data from Centers for Medicare & Medicaid Services, available at http://www.cms.hhs.gov/NationalHealthExpendData/downloads/tables.pdf

Note: NHE stands for "National Health Expenditure." All figures are in billion dollars.

produce a different protein if excited differently. An externally introduced protein, say from a virus or from a pollutant, can combine with one or more of these proteins and have a deleterious effect. Thus the objective of a drug is to "bind" with the target protein, and either retard the action of the protein (if it is harmful) or promote it (if it is beneficial).

Until quite recently, new drugs were attempted to be discovered predominantly by trial and error. Once a drug target was identified, it was then hit with a huge number of compounds from a "compound library" (numbering sometimes in the millions) to see which if any of them would interact with the target protein. Among the small molecules that seem to bind to the target protein, some are chosen for further testing and fine-tuning. If they show any promise at all, then they are referred to as "drugs under development." However, even at this stage, success is not guaranteed. The 2009 Industry Profile of PhRMA shows that in 1999 1800 compounds were deemed to have been "under development"; yet in 2008, only 31 compounds were approved, a ratio of just about 1.5%. This enormous "attrition rate" of 98.5% of drugs under development has been fairly constant over the years, unfortunately. This statement applies primarily to the so-called "small molecule" drugs and not to biologics, but small molecule drugs are still a major part of the drug discovery landscape.

22.4 Two Success Stories

The main reason for the enormous attrition rate mentioned above is that by and large there is not enough first-principles modeling of what causes a therapeutic condition, and how a drug is supposed to work. In the absence of such modeling, trial and error is the only option. However, there have indeed been some success stories based on physiological modeling. Among the many success stories that can be selected, two are highlighted here, namely diabetes and HIV.

22.4.1 Diabetes

There are two types of diabetes, namely Type 1 and Type 2. Of these, Type 2 is by far more common, accounting for more than 90% of patients. Type 1 diabetes is caused at a very young age and its cause is not properly understood. Type 2 diabetes is caused by an insufficient production of insulin by the pancreas, or in some cases, by the body becoming resistant to the insulin that it produces (in adequate quantity). Recall that insulin is the enzyme that regulates the level of blood glucose within acceptable limits. Several decades ago, patients used to administer themselves insulin several times a day, to synchronize with meals

(which would cause blood glucose levels to rise). Subsequently several experiments were conducted with "automatically" detecting the level of blood glucose and causing insulin to be released from an external store. This task is made very tricky by the fact that insulin is highly unstable.

At the present state of knowledge, there are by now hundreds of papers that address various aspects of the problem, including

- A dynamical model of the glucose control system, and specifically the role of insulin in regulating the level of glucose.
- Various methods of detecting when glucose levels are too low (e.g., if too much insulin has been introduced), or too high (e.g., after a meal).
- Methods of delivering insulin through methods other than injection, such as nasal inhalation, subcutaneous release, and so on.
- Model-based predictive control of the control system.

The reader is referred to just a couple of illustrative examples, such as [2,3]. The chapter [4] contains a wealth of references on detecting when a meal has been consumed, which is an important part of glucose monitoring and regulation. Model-predictive control (MPC) is very popular as a means of glucose regulation, because MPC makes minimal assumptions about the unknown system, and the time constants of the blood glucose system are sufficiently slow that MPC can work very effectively. There are even attempts now to build an "artificial pancreas" that consists of a neat package that encapsulates sensors, actuators and control.

22.4.2 HIV Treatment

Now let us come to HIV, whose success story is more recent and more rapid when compared to diabetes. The discussion below follows [5]. The reason for choosing this particular chapter is that it illustrates some of the special features of biological systems, as explained in due course.

HIV is caused by an external virus (the HIV virus) that causes CD4+ T-cells in the body to become infected. A model for the dynamics of uninfected CD4+ T-cells (T), infected CD4+ T-cells (T^*) and virions (V) is given below:

$$\dot{T} = s - \delta T - \beta TV \left(+ rT \frac{V}{K+V} \right), \dot{T}^* = \beta TV - \mu T^*, \dot{V} = kT^* - cV.$$

The various terms in the above equation are explained as follows: T represents the density of uninfected cD4+ T-cells, T^* represents the density of infected CD4+ cells, and V represents the density of the virions (virus products). The premise is that, in the absence of other factors, each type of cell exhibits exponential decay with its own time constant. In the presence of the virus, the rate of infection of previously uninfected cells is proportional to the *product* of the densities of uninfected cells and virions, the logic being that the product is proportional to the probability that an uninfected cell will come into contact with a virion. Thus, in the equation for $\dot{T}$, the following terms are present: (1) s corresponds to a steady accretion of the density of T-cells in the absence of any other influence; (2) $-\delta T$ corresponds to the exponential decay of T-cells; (3) βTV, corresponds to the rate at which healthy (or uninfected) T-cells get infected; and (4) the term $rTV/(K+V)$, which is an example of Michaelis–Menten kinetics which we will discuss separately. Basically, this term appears to be linear in V when $V \ll K$ and "saturates" at rT when $V \gg K$. The term $rTV/(K+V)$ is called the "proliferation term"; it should be mentioned that not all models of HIV use it. In the equation for $\dot{T}^*$, there are just two terms. First, the infection of healthy T-cells adds to the ranks of the infected, and this is the βTV term, and second, even infected T-cells decay exponentially, in this case with the time constant μ. Finally, in the equation for the density of virions, the infected cells in turn produce virions leading to the kT^* term, and virions also decay at an exponential rate with time constant c.

Note that there is no "treatment" in the above model. HIV is usually treated with reverse transcriptase inhibitors (RTI) and protease inhibitors (PI). In this case the virions do not all replicate as infectious

virions. Let V_1 denote the infectious virions and V_2 the noninfectious virions. The model is now

$$\dot{T} = s - \delta T - (1 - \eta_{RTI})\beta T V_1 \left(+ rT \frac{V}{K + V} \right),$$

$$\dot{T}^* = (1 - \eta_{RTI})\beta T V_1 - \mu T^*,$$

$$\dot{V}_1 = (1 - \eta_{PI})kT^* - c_1 V_1, \dot{V}_2 = \eta_{PI} kT^* - c_2 V_2,$$

where η_{RTI}, η_{PI} denote the effectiveness of the treatment. Thus, with treatment, out of the original term kT^*, a fraction η_{PI} replicate as noninfectious virions, while the others replicate as infectious virions. Similarly, when an infectious virion comes into contact with an uninfected T-cell, the T-cells get infected only $1 - \eta_{RTI}$ of the time.

Note that in the above system description there is no "control" term; instead, the "control signal" (the treatment) affects the *parameters of the system*, in this case the constants η_{RTI} and η_{PI}. This is one of the characteristics of physiological systems in the context of drug design, namely that the "control" (namely the drug) acts indirectly by affecting some of the parameters in the physiological model. Another characteristic of such physiological models is that the right side of the differential equations consists of terms that are either linear or bilinear.* The linear terms usually represent decay with a fixed time constant, while the bilinear terms represent interactions whose rate is proportional to the *product* of the two densities. Thus one will never see a quadratic term on the right side. One feels that there ought to be a theory of such systems, but thus far not enough attention has been paid to this class of systems.

Now that a model has been presented, it is natural to ask whether the model is "identifiable," that is, whether the various parameters can be identified on the basis of experiment. In [5], it is shown through rigorous analysis that the system is indeed identifiable. Thus by sampling the T cells of individual patients, the parameters in the model can be "identified" using standard system identification methods. Finally, the values of identified parameters can be used to "fine-tune" the therapeutic regime. Such approaches are actually being used in practice.

22.5 Some Considerations in Modeling the Human Body as a Dynamical System

Physiological modeling of the effect of drugs or external stimulants on the human body can be broadly categorized under the two headings of pharmacokinetics (PK) and pharmacodynamics (PD). By convention, pharmacokinetics is defined as "what the body does to the drug" while pharmacodynamics is defined as "what the drug does to the body." When a drug is introduced into the body, one studies the so-called ADME, which is an acronym for absorption, distribution, metabolism and excretion. It is sad but true that an overwhelming majority of the drug introduced into a body, sometimes as much as 99%, simply passes through without doing anything, or else breaks down before reaching the targeted organ. The phrase "bioavailability" refers to the fraction of the administered drug that reaches the site of physiological activity in an unchanged condition.

Ingestion of a drug is like an impulse to the body, concentrated in both space and time. If one wishes to study the temporal distribution of the drug in the target organ, then the model would be a set of ordinary differential equations. Sometimes, one is interested in more detailed information of how the drug is distributed within the target organ. In such a case one would use a model consisting of a set of partial differential equations.

* Perhaps one can coin the barbaric phrase "bi-affine" to describe such systems.

22.5.1 The Reaction Diffusion Model

One of the first models of how a drug is dispersed in the body is called the "reaction diffusion equation." It was first formulated by Alan Turing, who was said to be rather more famous for some other contributions! The reaction diffusion equation assumes the form

$$\frac{\partial \mathbf{u}}{\partial t} = K\nabla^2 \mathbf{u} + \mathbf{f}(\mathbf{u}),$$

where $\mathbf{u}(x, t)$ denotes a vector of variables of interest (varying over both space and time), K is a diagonal matrix of diffusion coefficients, and $\mathbf{f}(\mathbf{u})$ denotes the interactions among variables. Many biological phenomena can be described by this equation via suitable choices of K and $\mathbf{f}(\mathbf{u})$. Due to its nonlinear nature, "closed-form" solutions are not possible; but qualitative analysis is possible.

22.5.2 Compartmental Models

To analyze ADME properties, compartmental models are widely used. These models consist of viewing the human body as a sequence of organs, where the drug diffuses from one organ to the next. For the most part, within the human body the flow of drugs is unidirectional, though there may be exceptions. In compartmental models, one can study either the absolute amount of a drug within a compartment, or its concentration. If V_i is the volume of the ith organ, and $x_i(t)$ is the amount of drug in the ith organ, then the concentration of the drug in the ith organ is obviously $x_i(t)/V_i$. Mass balance (in its pure form) would require that the net change in $x_i(t)$ across all organs must add up to zero at each time interval. However, in reality drugs can also degrade, meaning that they get transformed into some other by-product. Hence, mass balance in its pure form need not always hold in such models.

A typical compartmental model has the following form. Let $c_i(t)$ denote the concentration of the drug in compartment i at time t, and let λ_{ij} denote the rate of diffusion from compartment i to compartment j. Then the typical compartmental model is

$$\dot{c}_i(t) = -\sum_{j=1}^{n} \lambda_{ij} c_j(t) + u_i(t), \ \forall i.$$

The coefficient λ_{ij} is nonzero if and only if compartments i and j are connected; moreover, in general $\lambda_{ij} \neq \lambda_{ji}$. Usually, only one compartment has an external input, and very few compartments can be accessed externally to measure $c_i(\cdot)$; thus the various coefficients have to be inferred. However, based on physiology, it is obviously very easy to determine the pairs (i, j) for which the diffusion coefficient λ_{ij} is nonzero. Often one uses animal studies to determine typical values of the diffusion coefficients, and then scales up to humans after adjusting for the volumes of the organs, differences in weight, and so on.

A recent development is the picturesquely named "PBPK" or "physiologically-based pharmacokinetics." In this approach, one tries to identify how the characteristics of one compartment (organ) differ from those of another in terms of its functionality and other characteristics.

The above compartmental model consists of linear dynamics and is therefore, easy to analyze. Moreover, various diffusion constants can be estimated from observations. However, this model is generally viewed as being unrealistic, because it permits arbitrarily large rates of diffusion from one compartment to another. To overcome this problem, the generally accepted approach is to use Michaelis–Menten kinetics. In this model, one assumes that the rate of change of concentration "saturates," as shown below:

$$\dot{c}_i(t) = \frac{V_{\max} \delta_i(t)}{K_i + \delta_i(t)},$$

where

$$\delta_i(t) = -\sum_{j=1}^{n} \lambda_{ij} c_j(t) + u_i(t), \forall i.$$

In the above equation, if $\delta_i(t) \ll K_i$, then one can neglect $\delta_i(t)$ in comparison to K_i, and the rate of change of concentration $\dot{c}_i(t)$ is linearly proportional to $\delta_i(t)$. As $\delta_i(t)$ increases, the resulting dynamics become nonlinear. Finally, if $\delta_i(t) \gg K_i$, then one can neglect K_i with respect to $\delta_i(t)$, and $\dot{c}_i(t)$ saturates as $V_{\max}$.

In Michaelis–Menten kinetics, the underlying premise is that the rate of change of concentration is limited by the volume of the relevant organ. In particular, if a (physically) small organ is adjacent to a much larger organ, the concentration in the smaller organ cannot increase faster than a certain rate. Taking this argument further, it is also possible to discuss a more refined model known as target mediated drug disposition. This model is relevant when the concentration of the drug is limited not only by the volume of the target organ, but also by its binding affinity to the target protein(s). See for instance [6] for a discussion.

There are several excellent texts in both traditional PK/PD as well as PBPK, and we cite only [7] as a sample. It is a collection of articles and is therefore a good starting point. There are also several more traditional (though somewhat dated) texts on the topic. The style of exposition in the book by Macheras [8] is close to that found in a typical controls or system theory text.

22.6 Conclusions

There are several outstanding issues in modeling the human body that are not touched upon in this chapter. One of the most challenging, and at the same time most promising task, is using physiological modeling coupled with statistical analysis to predict "adverse events," that is, the likelihood that a particular drug may induce toxic side effects in an unacceptably high fraction of the population. The phrase "unacceptably high" could refer to as little as 0.5%, and often adverse events lie outside the realm of the observable. For instance, in animal studies or in Phase I human trials, a drug may be administered to only 10–20 subjects. Thus, at best one can only speak about the 95th percentile of the data, and yet one is obliged to make educated guesses about the 99.5th percentile! This calls for novel approaches that combine pure probability theory for predicting extreme events (such as large deviation theory) with machine learning techniques for estimating the probability distribution of the unknown parameter set. As our understanding of human physiology improves, it is reasonable to assume that drug discovery will make greater and greater use of system-theoretic approaches and methods.

References

1. PhRMA (Pharmaceutical Research and Manufacturers of America), Industry Profile, 2009.
2. R. Hovorka, J. Kremen, J. Blaha, M. Matias, K. Anderlova, L. Bosanska, T. Roubicek, et al., Blood glucose control by a model predictive control algorithm with variable sampling rate versus a routine glucose management protocol in cardiac surgery patients: A randomized controlled trial, *The Journal of Clinical Endocrinology and Metabolism*, 92(8), 2960–2964, 2007.
3. M. W. Percival, E. Dassau, H. Zisser, L. Jovanovic, and F. J. Doyle III*, practical approach to design and implementation of a control algorithm in an artificial pancreatic beta cell, *Industrial & Engineering Chemistry Research*, 48, 6059–6067, 2009.
4. E. Dassau, B. W. Bequette, B. A. Buckingham, and F. J. Doyle III, Detection of a meal using continuous glucose monitoring: Implications for an artificial β-cell, *Diabetes Care*, 31, 295–300, 2008.
5. D. A. Ouattara, M.-J. Mhawej, and C. H. Moog, Clinical tests of therapeutical failures based on mathematical modeling of the HIV infection, *IEEE Transactions on Automatic Control and Ciruits and Systems*, (Joint Special Issue on Systems Biology), AC-53, 23–241, January 2008.

6. X. Yan, D. E. Mager, and W. Krzyzanski, Selection between Michaelis–Menten and target mediated drug disposition pharmacokinetic models, *Journal of Pharmacokinetics and Pharmacodynamics*, 37(1), 25–47, February 2010.
7. M. B. Reddy (Ed.), *Physiologically Based Pharmacokinetic Modeling*, Wiley-Interscience, Hoboken, NJ, 2005.
8. P. Macheras, *Modeling in Biopharmaceuticals, Pharmacokinetics, and Pharmacodynamics: Homogeneous and Heterogeneous Approaches*, Springer-Verlag, New York, 2006.

V

Electronics

23

Control of Brushless DC Motors

Farhad Aghili
Canadian Space Agency

23.1 Optimal Torque Control of Brushless DC Motors Through Quadratic Programming

23.1.1 Introduction

Accurate and ripple-free torque control of electric motors is essential to precision motion control, with a huge range of applications: from silicon wafer manufacturers, medical, robotics and automation industries

to the military. Permanent-magnet synchronous motors, also known as brushless DC (BLDC) motors, are commonly used as the drives of servo systems. BLDC motors are composed of a rotor containing a series of permanent magnets and the armature which remains static while the electric power is distributed by an electronically controlled commutation system, instead of a mechanical commutator using brushes. BLDC motors offer several advantages over brushed DC motors making them suitable for use as servomotors. These include higher efficiency and longer lifetime because of the absence of electrical and friction losses as well as erosion due to brushes, and reduction of electromagnetic interference and noise because of elimination of ionizing sparks from the brushes. Moreover, BLDC motors can deliver more power per mass compared to their cousin DC motors because the stator windings attached to the BLDC motor's housing can be cooled effectively through conduction whereas DC motors with windings on the rotor dissipate heat mainly by convention, typically by a cooling fan inside the motor. This also means that BLDC motors can be completely sealed off and protected from dirt, oil, grease, and other types of foreign matter.

BLDC motors achieve commutation electronically by incorporating a feedback from the rotor-position into a control system instead of a mechanical commutator found in brushed DC motors. Encoders are usually utilized to measure the rotor's position. However, some designs use Hall effect sensors or measure the back EMF in the undriven coils to extract the rotor position information. The position sensor, however, is not a burden if the motor is the actuator of a motion control system as the very position sensor, which provides feedback for the motion controller can be also used for the electronically controlled commutator. The controller takes the rotor position information and control input signals to excite the stator coils of the motor in a specific order in order to rotate the magnetic field generated by the coils to be followed along by the rotor. A comprehensive description of the basic structure of electric machines including BLDC motors and the drive, conventional controller, circuitry, and power electronics can be found in [1–3].

Suppressing the torque ripple of the motor drive of a servo system can significantly improve system performance by reducing speed fluctuations [4,5]. In general, electric motors generate torque-ripple due to the distortion of the distribution of flux linkage and/or the variation of magnetic reluctance due to saliency. The control problem then is: how to modulate the excitation currents as a function of motor angular position such that the instantaneous torque generated by the motor is equal to the command torque at every angular position. Control approaches for generating accurate torque with electric motors and their underlying models have been studied by several researchers [4–11]. The traditional trend in the control of BLDC motors is to transform the excitation currents via the so-called d–q transformation [2]. Since this transformation linearizes only an ideal motor with a perfectly sinusoidally distributed magneto-motive force, another torque set point is cascaded to cancel torque ripples [12]. Murai et al. [6] proposed a heuristic commutation scheme for nonsinusoidal flux distribution. Le-Huy et al. [13] reduced the torque-ripple harmonics for brushless motors by using several current waveforms. Ha et al. [14] completely characterized, in an explicit form, the class of feedback controllers that produce ripple-free torque in brushless motors. Optimal torque control was addressed in [4,10]. Control strategies based on feedback linearization [15], and Hamiltonian and energy-based realization [11] have been proposed in the past.

The control problem is radically simplified when the motor's phase currents are considered as the inputs as opposed to the armature voltages. Then the control problem is reduced to the torque control of motors, which is a nonlinear mapping from desired torques and positions to phase currents, and the control of the multi-body dynamics of the manipulator that traditionally relies on torque control inputs. Control approaches for accurate torque production in direct drive systems and their underlying models have been studied by several researchers [8]. A free function can be used to achieve other control objective, such as minimization of power dissipation, but phase current saturation was not considered [10]. Optimal torque control taking current limitation into account was addressed in [4].

Sections 23.2 through 23.4 present the design of a ripple-free torque controller that minimizes copper losses and maximizes the torque capability of the motor under current limitation. In conventional commutation approaches with fixed current-angle waveforms, the maximum torque is reached when at least one phase current saturates. In our scheme, through the application of constrained optimization, the

waveforms vary to compensate saturated (current limited) phases by boosting currents of the unsaturated phases. This tends to increase the maximum torque capability of the motor while operating in linear magnetic regime. In this scheme the motor torque capability is increased because the motor can produce more torque until current of all phases saturate.

23.2 Optimal Phase Current

23.2.1 Motor Model

We assume that there is negligible cross-coupling between the phase torques and there is no reluctance torque. In addition, we assume that the phase currents can be controlled accurately and instantaneously so that the phase currents can be treated as the control inputs. Then, the torque developed by a single-phase is a function of the phase current i_k and the (angular motor) position θ

$$\tau_k(i_k, \theta) = i_k\, y_k(\theta) \quad k = 1, \ldots, p \tag{23.1}$$

where $y_k(\theta)$ is the position nonlinearity associated with the kth phase, or torque shape function. For brevity, we shall omit the argument θ in the sequel. The motor torque τ is the superposition of all phase torque contributions

$$\tau = \sum_{k=1}^{p} i_k(\theta, \tau_d)\, y_k(\theta). \tag{23.2}$$

The torque control problem is to solve the above equation in terms of current, $i_k(\theta, \tau_d)$, as a function of motor position, given a desired motor torque τ_d. Given a scalar torque set point, Equation 23.2 permits infinitely many (position dependent) phase current wave forms. Since the continuous mechanical power output of electrical motors is limited primarily by heat generated from internal copper losses, it makes sense to use the freedom in the phase current solutions to minimize power losses

$$P_{\text{loss}} \propto i^T i, \tag{23.3}$$

where $i = \text{col}\left(i_1, \ldots, i_p\right)$ is the vector of phase currents. Current saturation is the other limitation which should be considered. Let $i_{\max} > 0$ be the maximum equivalent phase current corresponding to a linear phase current–torque relationship, that is, Equation 23.1 is valid, or to current limit of the servo-amplifier. Then the phase currents must satisfy

$$-i_{\max} \le i_j \le i_{\max} \quad \forall\, j = 1, \ldots, p. \tag{23.4}$$

23.2.2 Quadratic Programming

In order to derive the optimal phase currents $i_k^*(\theta, \tau_d)$ which generate the desired torque (Equation 23.2) and minimize the power losses (Equation 23.3) subject to the constraints (Equation 23.4) we need the torque functions $i_k(\theta)$. Let the functions be represented in a discrete manner at a finite number of motor positions. Then, the values of the functions at any specific position θ, that is, $\{y_1(\theta), \ldots, y_p(\theta)\}$, can be interpolated. Hereafter, we drop the argument θ for simplicity. Now, by setting $\tau = \tau_d$ in Equation 23.2, the problem of finding optimal phase currents that minimize power losses subject to the constraints is formulated by the quadratic programming problem,

$$\min \quad i^T i \tag{23.5a}$$

$$\text{subject to:} \quad h(i) = y^T i - \tau_d = 0 \tag{23.5b}$$

$$g_1(i) = |i_1| - i_{\max} \le 0 \tag{23.5c}$$

$$\vdots$$

$$g_p(i) = |i_p| - i_{\max} \le 0$$

Since all the functions are convex, any local minimum is a global minimum as well. Now, we seek the minimum point $i^* = \text{col}\left(i_1^*, i_2^*, \ldots, i_p^*\right)$ satisfying the equality and inequality constraints. Before we pay attention to the general solution, it is beneficial to exclude the trivial solution, $i_k^* = 0$. If the kth torque shape function is zero, that phase contributes no torque regardless of its current. Hence,

$$y_k = 0 \implies i_k^* = 0 \quad \forall\, k = 1, \ldots, p \tag{23.6}$$

immediately specifies the optimal phase currents at the crossing point. By excluding the trivial solution, we deal with a smaller set of variables and number of equations in our optimization programming. Therefore, we have to find the optimal solution corresponding to the nonzero part. Hereafter, without loss of generality, we assume that all torque shape functions are nonzero.

Now, by defining the function

$$\mathcal{L}(i) = f(i) + \lambda h(i) + \mu^T g(i), \tag{23.7}$$

where $f(i) = i^T i$, $g(i) = \text{col}\left(g_1(i), g_2(i), \ldots, g_p(i)\right) \in \mathbb{R}^p$, $\lambda \in \mathbb{R}$, and $\mu = \text{col}\left(\mu_1, \mu_2, \ldots, \mu_p\right) \in \mathbb{R}^p$. Let i^* provide a local minimum of $f(i)$ satisfying the equality and inequality constraints Equations 23.5b and 23.5c. Assume that vectors $(\partial g_k/\partial i)^T|_{i=i^*} \forall\, k = 1, \ldots, p$ are linearly independent. Then according to the Kuhn–Tucker theorem [16], there exist $\mu_k \geq 0\ \forall\, k = 1, \ldots, p$ such that

$$\left(\frac{\partial \mathcal{L}}{\partial i}\right)_{i=i^*} = 0 \tag{23.8a}$$

$$\mu^T g(i^*) = 0. \tag{23.8b}$$

Let $\text{sgn}(\cdot)$ represent the sgn function, where

$$\text{sgn}(x) = \frac{d}{dx}|x|.$$

Then $(\partial g_j/\partial i)^T = \text{diag}\big(\text{sgn}(i_1^*), \text{sgn}(i_2^*), \ldots, \text{sgn}(i_p^*)\big)$ is a diagonal matrix whose columns are linearly independent. The only pitfall is $i_k^* = 0$, where the sign function is indefinite. We assume that the optimal solutions, i_k^* are nonzero because $y_k \neq 0$. This assumption will be relaxed later. Substituting $f(i)$, $h(i)$ and $g(i)$ into Equation 23.8 yields

$$2i^* + \lambda y + \mu^T \text{sgn}(i^*) = 0 \tag{23.9}$$

$$\mu_k(|i_k^*| - i_{\max}) = 0 \qquad k = 1, \ldots, p \tag{23.10}$$

Equations 23.9 and 23.10 together with Equation 23.5b constitute a set of $2p+1$ nonlinear equations with $2p+1$ unknowns i^*, λ, and μ to be solved in the following. Since $\mu^T g(i^*) = 0$ while $\mu \geq 0$ and $g(i^*) \leq 0$, we can say that $\mu_k = 0$ for $|i_k| < i_{\max}$, and that $\mu_k \geq 0$ for $|i_k| = i_{\max}$. Therefore, Equation 23.9 can be written in the following compact form:

$$T(i_k^*) = -0.5\lambda y_k \quad \forall\, k = 1, \ldots, p. \tag{23.11}$$

The mapping $T : \mathcal{D} \mapsto \mathbb{R}$, and $\mathcal{D}(x) = \{x \in \mathbb{R} : |x| \leq i_{\max}\}$, is defined by

$$T(x) = \begin{cases} x & |x| < i_{\max} \\ x + 0.5\text{sgn}(x)\mu & |x| = i_{\max} \end{cases} \tag{23.12}$$

where μ_k is any positive number. It is apparent that the mapping is invertible on $\mathcal{D}$, that is there exists a function $T^{-1}(x)$ such that $T^{-1}(T(x)) = x\ \forall\, x \in \mathcal{D}$. In other words, the variable i_k^* in Equation 23.11 can

be determined uniquely if the right-hand-side (RHS) of the equation is given. The inverse of the mapping is the saturation function, that is, $T^{-1}(\cdot) \equiv \text{sat}(\cdot)$, defined by

$$\text{sat}(x) = \begin{cases} x & |x| \le i_{\max} \\ \text{sgn}(x) i_{\max} & \text{otherwise.} \end{cases} \tag{23.13}$$

Now, Equation 23.11 can be rewritten as

$$i_k^* = \text{sat}(-0.5\lambda y_k) \quad \forall\, k = 1, \ldots, p. \tag{23.14}$$

The above equation implies that $i_k^* \neq 0$ as $y_k \neq 0$, which relaxes the assumption we made earlier. The second result is that the larger the magnitude of the torque shape function $|y_k|$, the larger the magnitude of the optimal current i_k^*. If the phases are labeled in descending order,

$$|y_1| \ge |y_2| \ge \cdots \ge |y_p| \implies |i_1^*| \ge |i_2^*| \ge \cdots \ge |i_p^*|, \tag{23.15}$$

the optimal phase currents from i_1^* to i_p^* must be saturated consecutively. We use this fact to calculate the optimal phase currents consecutively in the same order, starting with i_1^*. In case saturation of a phase occurs, Equation 23.14 implies that only knowing the sign of λ is enough to calculate the associate phase current. One can infer from Equations 23.11 and 23.5b that

$$\text{sgn}(\tau_d) = \text{sgn}(-\lambda). \tag{23.16}$$

Therefore, if i_1^* saturates, then

$$i_1^* = \text{sgn}(-y_1\lambda) i_{\max} = \text{sgn}(i_1 \tau_d) i_{\max}. \tag{23.17}$$

If i_1^* does not saturate, that is, $|i_1^*| < i_{\max}$, then neither does $\{i_2, \ldots, i_p\}$, see Equation 23.15. Let $\lambda^{(1)}$ represents the Lagrangian multiplier when i_1^* does not saturate, then the Lagrangian multiplier can be calculated by substituting phase currents from $i_k^* = -0.5\lambda y_k$ into Equation 23.5b

$$\lambda^{(1)} = \frac{-2\tau_d}{\sum_{k=1}^{p} y_k^2} \tag{23.18}$$

which, in turn, can be substituted in Equation 23.14 to obtain the optimal phase current

$$i_1^* = \text{sat}\left(\frac{y_1 \tau_d}{\sum_{k=1}^{p} y_k^2}\right). \tag{23.19}$$

Since the denominator in Equation 23.19 is always positive, by virtue of Equation 23.17, one can infer that Equation 23.17 provides the optimal solution for the saturation case as well. Analogously, i_2^* can be calculated if $y_1 i_1^* - \tau_d$ is treated as the known parameter in Equation 23.5b. In general, the ith phase current can be calculated by induction as follows: since up to $(i-1)$th phase currents have been already solved, we have

$$y_i i_i^* + \cdots + y_p i_p^* = \tau_d - (y_1 i_1^* + \cdots + y_{i-1} i_{i-1}^*), \tag{23.20}$$

where the value of the RHS of the above equation is known. The Lagrangian multiplier associated with the case of unsaturated i_i^* can be found from Equations 23.14 and 23.20 as

$$\lambda^{(i)} = \frac{-2(\tau_d - \sum_{k=1}^{i-1} y_k i_k^*)}{\sum_{k=i}^{p} y_k^2}. \tag{23.21}$$

Finally, substituting Equation 23.21 in Equation 23.14, gives the optimal phase currents, which produce the desired torque precisely, while minimizing power losses, subject to the constraints of current

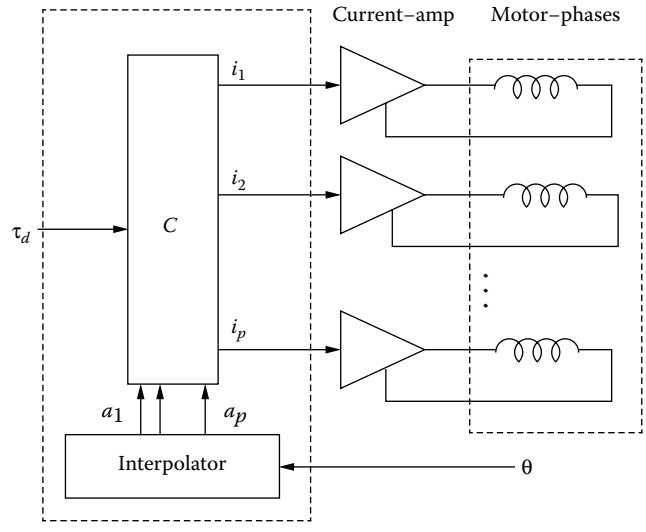

FIGURE 23.1 The torque controller and power electronics.

saturation,

$$i_1^* = \text{sat}\left(\frac{y_1 \tau_d}{\|y\|^2}\right) \tag{23.22}$$

$$i_i^* = \text{sat}\left(\frac{y_i \tau_d - y_i \sum_{k=1}^{i-1} y_k i_k^*}{\sum_{k=i}^{n} y_k^2}\right), \quad i = 2, \ldots, p. \tag{23.23}$$

23.2.3 Implementation of the Torque Control Algorithm

Assume that vector col $\big(y_k(\theta_1), y_k(\theta_2), \ldots, y_k(\theta_n)\big) \in \mathbb{R}^n$ represents the discrete torque shape functions corresponding to n measurements of the phase torque (with unit current excitation) and positions. Then at any given position θ, the corresponding shape function $y_k(\theta)$ can be calculated via interpolation. Figure 23.1 illustrates the control system architecture. The torque control algorithm is implemented as follows:

1. Interpolate the torque functions y_k for the current motor position.
2. Set $i_k^* = 0$ for $y_k = 0$ (or for sufficiently small $|y_k|$).
3. Pick the set of nonzero shape functions and sort them such that $|y_1| \geq |y_2| \geq \cdots \geq |y_p|$ and calculate the optimal currents from Equations 23.19 and 23.23. Go to step (1).

23.3 Maximum Attainable Torque

The control algorithm presented in previous section permits torque among phases when some phases saturate. How much torque is gained by this method? One can show that the optimal solution of phase current without taking the saturation into account can be expressed explicitly in closed form as

$$i_k(\theta, \tau_d) = \frac{y_k \tau_d}{\sum_{m=1}^{p} y_m^2} \qquad \forall\, k = 1, \ldots, p. \tag{23.24}$$

In this case, the maximum torque depends on the saturation of the largest phase torque function. It is a clear from Equation 23.24 that, at any given motor position θ, the phase with the largest torque

shape function $|y_j|$ reaches saturation first. Again, assuming that $|y_1| \geq |y_2| \geq \cdots \geq |y_p|$, then maximum achievable torque can be calculated from Equation 23.24

$$|\tau_d| \leq (|y_1| + |y_2/y_1||y_2| + \cdots + |y_p/y_1||y_p|)i_{\max} = k_1(\theta)i_{\max}. \tag{23.25}$$

On the other hand, the proposed algorithm increases the torque contribution of the unsaturated phases when one phase saturates, until, in the limit, all phases are saturated. Hence, the maximum torque is

$$|\tau_d| \leq (|y_1| + |y_2| + \cdots + |y_p|)i_{\max} = k_2(\theta)i_{\max}. \tag{23.26}$$

Both $k_1(\theta) > 0$ and $k_2(\theta) > 0$ are decisive factors in the torque capability of electric motors. Since $|y_2/y_1| \leq 1, \ldots, |y_p/y_1| \leq 1$, one can conclude from Equations 23.25 and 23.26 that $k_1(\theta) \leq k_2(\theta)$. The values of k_1 and k_2 depend on the torque shape functions, y_k. However, they can be expressed explicitly for an ideal three-phase motor, that is, $n = 3$, where we have a three-shifted sinusoidal torque function as

$$\begin{aligned} y_1(\theta) &= \hat{y}\sin(\theta + \varphi), \\ y_2(\theta) &= \hat{y}\sin\left(\theta + \frac{2\pi}{3} + \varphi\right), \\ y_3(\theta) &= \hat{y}\sin\left(\theta + \frac{4\pi}{3} + \varphi\right), \end{aligned}$$

where φ is an offset angle. In this case, using the properties of triangular functions, one can show that

$$1.5\hat{y} \leq k_1(\theta) \leq \sqrt{3}\hat{y}$$

$$\sqrt{3}\hat{y} \leq k_2(\theta) \leq 2\hat{y}.$$

Therefore, the maximum torque capability is boosted by $2/\sqrt{3}$ (15.5%) when the phase saturation is considered in the phase current shape function.

23.4 Experimental Characteristics

23.4.1 Experimental Setup

Figure 23.2 illustrates the experimental setup. The motor used for the testing is the McGill/MIT synchronous motor [17]. The motor and a hydraulic rack and pinion rotary motor are mounted on the rigid structure of the dynamometer. The hydraulic motor's shaft is connected to that of the direct drive motor via a torque transducer (Himmelstein MCRT 2804TC) by means of two couplings which relieve bending moments or shear forces due to small axes misalignments. The speed of the hydraulic motor is controlled by a pressure compensated flow control valve. The hydraulic pressure is set sufficiently high so that the hydraulic actuator regulates the angular speed regardless of the applied direct-drive motor torque. The motor torque is measured in a quasi-static condition, where the motor velocity is kept sufficiently low 1 deg/s, to ensure that the inertial torque does not interfere with the measurement. An adjustable camera and two limit switches detect the two rotational extremes and activate a solenoid valve through a PLC unit (not shown) to reverse the direction. The position sensor is an optical encoder mounted to the motor shaft. Its mechanical resolution of 4500 lines per revolution is extended 80 times by an electronic interpolator for 0.001° resolution. Three independent current servo amplifiers (Advanced Motion Control 30A20AC) control the motor's phase currents as specified by the processor. The amplifier's rated current and voltage are 15 A and 190 V with a switching rate of 22 kHz.

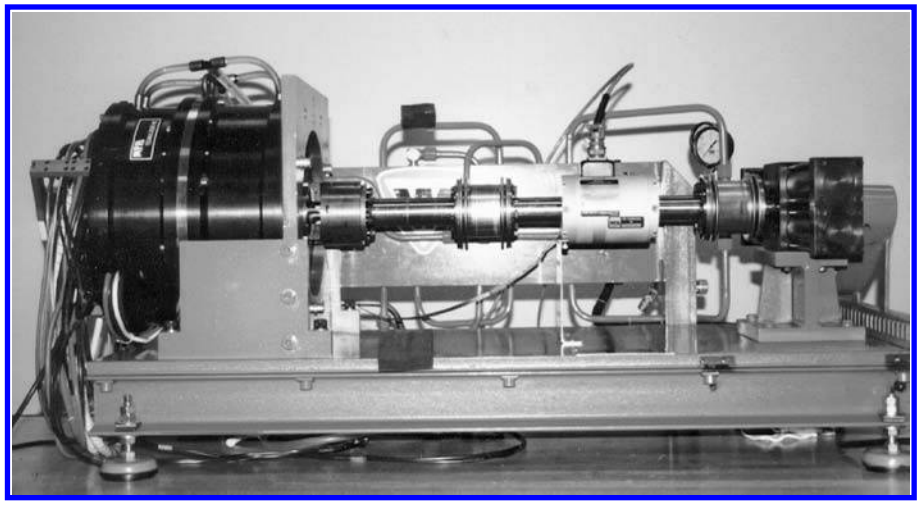

FIGURE 23.2 The motor prototype mounted on the dynamometer.

23.4.2 Friction and Cogging Torques

The torque shape function is measured by making use of the hydraulic dynamometer. To this end, the torque trajectory data versus position was registered during the rotation, while one phase was energized with a constant current. First however, the joint friction and cogging torque are identified and then subtracted from the torque measurement. The cogging torque is attributed to residual magnetization in the stator armatures [8] or to the presence of winding slots in the magnetic material, while the friction torque arises in the motor bearings and consists of viscous and dry friction. Since direct-drive motors operate at relatively low speeds, dry friction, τ_F, dominates. The main practical problem in identifying the phase torque-angle characteristic is that the dry-friction is position dependent.

Let $\tau_M(\theta)$ and $\tau_F(\theta)$ represent the motor torque and the magnitude of the dry-friction. Then

$$\tau_M(\theta) = \tau(\theta) - \tau_F(\theta)\,\mathrm{sgn}(\dot{\theta}). \tag{23.27}$$

If τ_M^+ and τ_M^- represent two sequences of motor torque measurements corresponding to clockwise and counterclockwise rotations, then the magnetic and friction torques can be calculated as

$$\begin{aligned} \tau(\theta) &= \frac{1}{2}[\tau_M^+(\theta) + \tau_M^-(\theta)], \\ \tau_F(\theta) &= \frac{1}{2}[\tau_M^+(\theta) - \tau_M^-(\theta)]. \end{aligned} \tag{23.28}$$

The cogging torque can be measured by setting the phase current to zero. The dry friction, cogging, and the three phases' torque-angle profiles (with friction and cogging torques subtracted) are illustrated in Figure 23.3 where the phase currents are individually set to 8 A. Although our experiments showed that friction torque with ± 1 Nm and cogging torques are relatively low, we compensate both for a more accurate torque generation.

23.4.3 Torque–Current Relationship

The torque–current relationship of the motor prototype is also investigated experimentally. Graphical realization of torque–current over all positions is difficult due to the large number of plots required. This is greatly simplified in the frequency-domain because of the small numbers of harmonics. Since the motor

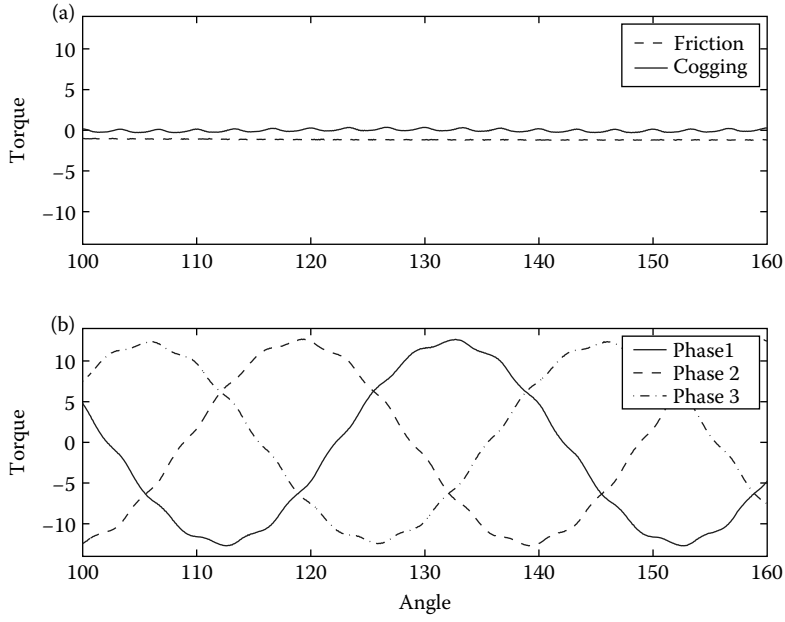

FIGURE 23.3 (a) Friction and cogging torques and (b) the three-phase torque-angle profiles.

has nine pole pairs, the torque trajectory is periodic in position with a fundamental spatial-frequency of 9 cpr (cycles/revolution) and thus the torque pattern repeats every 40°, as shown in Figure 23.3. The discrete Fourier series coefficients of the torque-position function are used to derive the spectrum. The frequency contents are expressed in harmonics of 9 cpr, that is, the spatial frequency of the 11th harmonics is 99 cpr. It turns out that the significant frequency components appear at the 1st, 11th and 13th harmonics.

Similar to the previous experiment, the torque–angle relationship is recorded within almost one rotation while the phase current is kept constant. But the current is incremented at the end of each rotation stroke by 1 A until an ensemble of torque profiles belonging to the span of $[-15, 15]$ A is obtained. The phase torque is a function of the position and the phase current, while the position-dependent part of the function is periodic. Therefore, we assume that the torque function of the kth phase can be expressed in terms of the complex Fourier series as

$$\tau_k(i_k, \theta) = \sum_{n=-\infty}^{\infty} \tau_k^n(i_k) e^{jqn\theta} \quad \forall\, k = 1, \ldots, 3,$$

where q is the number of motor poles, j is the imaginary unit, and $\tau_k^n(i_k)$ in the complex Fourier coefficient of the nth harmonic at contact excitation current i_k. The magnitude of the major torque harmonics of the first phase, $|\tau_1^n| = \mathrm{Re}(\tau_1^n) + \mathrm{Im}(\tau_1^n)$, are plotted versus current in Figure 23.4. Due to the motor phase symmetry, similar results are obtained for the other two phases. It can be concluded from these experimental results that the torque is a linear function of current within the current range for this particular motor. However, we will still be able to demonstrate the capability of the proposed torque controller to compensate for phase current limitations—a similar limitation as saturation—in the next section.

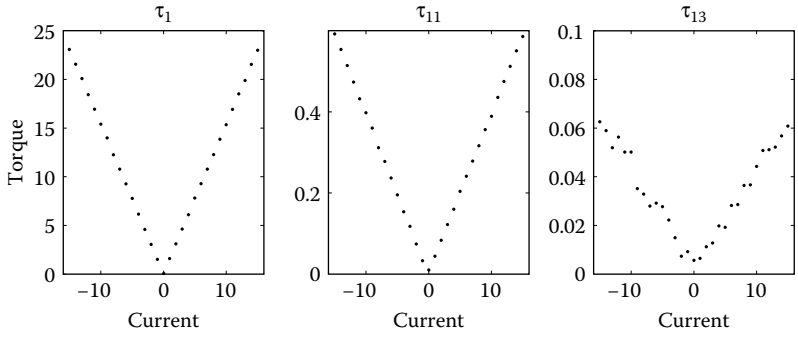

FIGURE 23.4 The magnitude of torque harmonics versus phase current.

23.5 Performance Test

23.5.1 Torque Ripple

The torque controller was tested on the dynamometer. Again, the motor shaft is rotated by the hydraulic actuator while the motor torque is monitored by the torque transducer. Figure 23.5 shows the motor torque versus position when standard sinusoidal commutation and our torque controller are applied. Clearly, a drastic reduction in torque ripple is achieved.

23.5.2 The Effect of Torque Ripple in Motion Control

A motor's torque ripple acts as a perturbation to the control system, degrading the tracking performance, especially at low velocities. We examine the position tracking accuracy of our direct drive system with

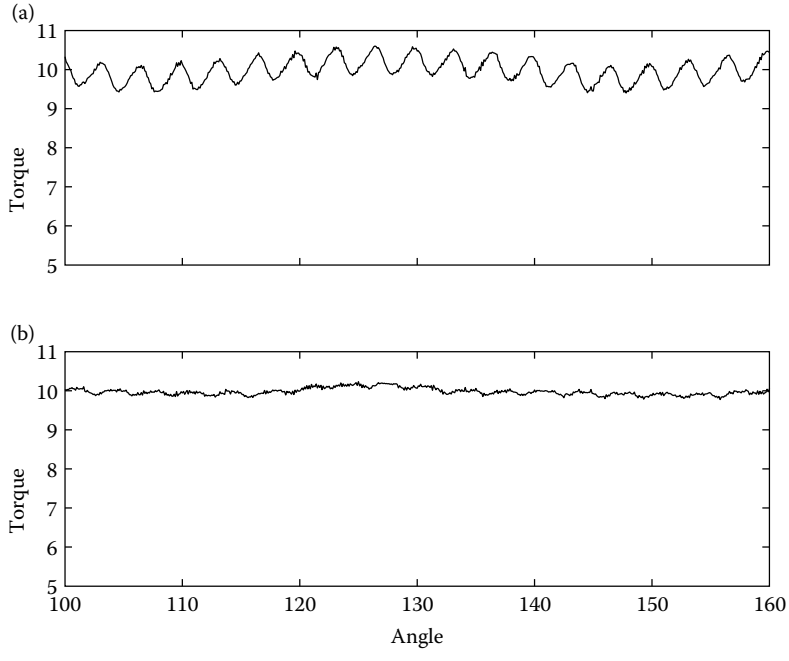

FIGURE 23.5 Motor torque profile (a) with sinusoidal commutation and (b) with the ripple free commutation.

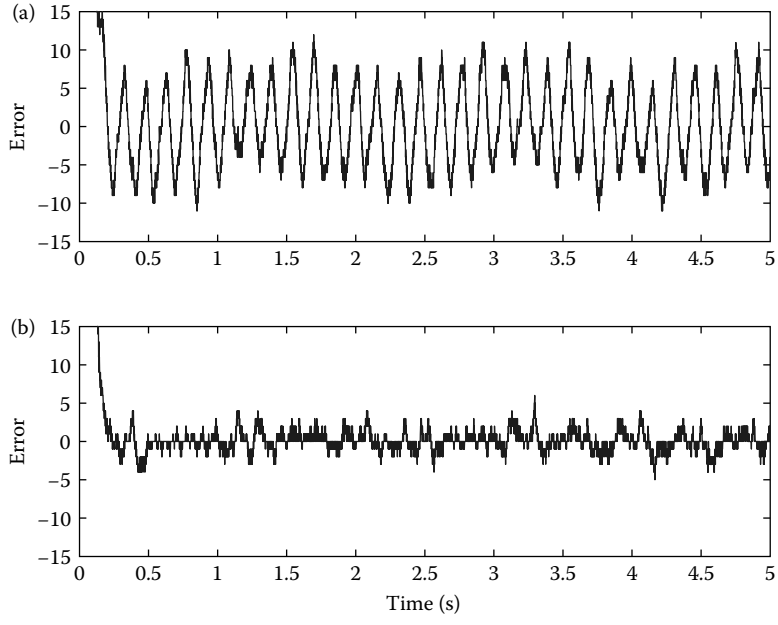

FIGURE 23.6 Position tracking errors to a ramp input (a) with sinusoidal commutation and (b) with the ripple free commutation.

and without torque ripple. To this end, a PID position controller $\tau_d = K_P e + K_I \int e\,\mathrm{d}t + K_D \dot{e}$, where $e = \theta_d - \theta$ (Gains: $K_P = 30$ Nm/deg, $K_I = 200$ Nm/deg · s, $K_D = 0.65$ Nm · s/deg) is implemented, in addition to the torque controller. Figure 23.6 documents the tracking error of the system to a ramp input, that is equivalent to a step input velocity of 20 deg/s, when the sinusoidal commutation (a) and controller (b) are applied. The figure clearly shows that the tracking error is limited by the torque ripple. In the absence of actuator torque ripples, the tracking error is reduced down to about the encoder resolution (0.001 deg).

23.5.3 Torque Saturation

By how much does the proposed controller improve the maximum torque capability of our motor prototype? This is investigated by comparing the maximum torque produced by the motor prototype when our proposed torque controller (Equation 23.23) and the conventional one (Equation 23.24) are applied. Figure 23.7 shows the graphs of the maximum achievable torque with respect to maximum phase current $i_{\max} = 15$ A. The solid line and the dashed line depicted the maximum attainable torque with respect to the proposed controller (Equation 23.23) and conventional controller (Equation 23.24), respectively. As described in Section 23.3 the torque saturation points differs from one position to another. Therefore, only the lowest torque value is available over all motor positions without having saturation induced torque ripple. It is evident from the graphs that the motor torque limits corresponding to the optimal torque controllers (Equations 23.23 and 23.24), that is, with and without taking current saturation into account, are 34 and 41 Nm, respectively—an increase of 20%.

One aspect of our proposed torque control algorithm is the current-position pattern which varies with requested torque. This is demonstrated in Figure 23.8a and b which show the current-position pattern of the motor with respect to the requested torques 10 and 38 Nm, respectively.

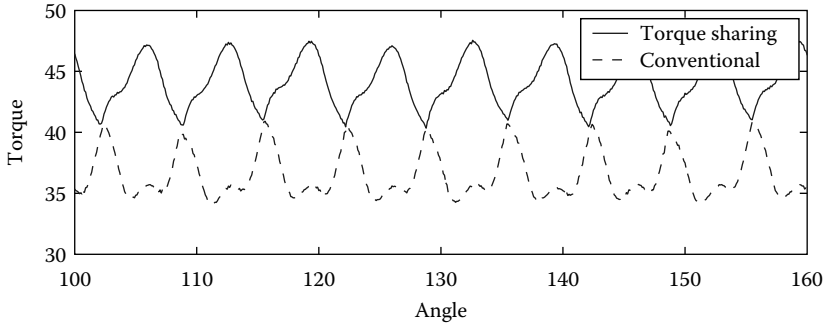

FIGURE 23.7 Maximum admissible torque corresponding to maximum phase current 15 A. Solid: with torque sharing, dashed: without torque sharing.

23.5.4 Two-Phase Commutation

Contrary to the past approaches, the proposed strategy does not rely on any condition for the phase torque-angle waveforms, for example, having balanced phases, which imposes the KCL constraint at the floating neutral node. One interesting aspect of the proposed controller is that it does not rely on any condition on the torque-position pattern of the phases, such as having balanced phases, where $\sum y_k \equiv 0$ [14]. Therefore, the control algorithm can achieve ripple-free torque even if a motor phase fails. As an illustration, Figure 23.9 shows the phase currents of two phases producing the same torque 10 Nm as the three phases in Figure 23.8a. This can be useful in practice when there is a need to continue operating the motor even in the case of a phase failure. However, the price is a higher power consumption—in this particular case from 75 to 128 W.

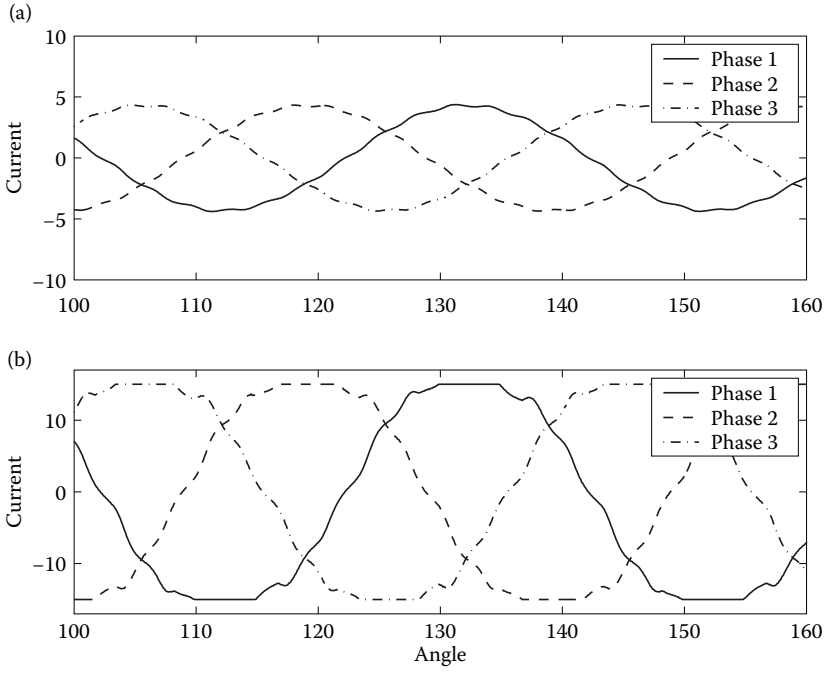

FIGURE 23.8 Phase current with the requested torque 10 Nm (a), and 38 Nm (b), respectively.

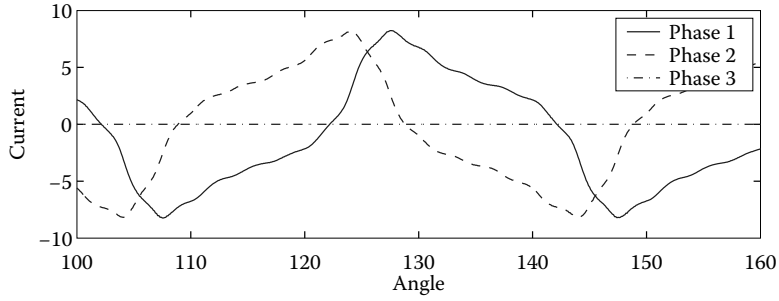

FIGURE 23.9 Phase current profile when the motor operates only with two phases.

23.6 Commutation Law Based on Spatial Frequency Analysis

23.6.1 Introduction

In rotary electric motors, both torque and commutation functions are periodic functions. Moreover, the phase torques are shifted versions of each other. Hence, sinusoidal bases naturally offer a very concise representation of the functions. In contrast, describing the motor wave function by a lookup table may require a vast amount of data. This feature can be well exploited in the spatial frequency analysis to simplify the design of the commutation, which also gives a great deal of insight [10]. In addition, the Fourier coefficients of the torque function can be extracted on-line based on phase voltage measurements only, as will be discussed in Section 23.9.

Section 23.7 presents a model for torque generation of brushless motors and its torque controller based on Fourier coefficients. The commutation law delivers ripple-free torque and simultaneously minimizes copper losses for the case when the motor's servo amplifier dynamics are negligible [10]. However, high motor velocity gives rise to high frequency control signals, and often the dynamics of the current amplifier are no longer negligible. To minimize the deteriorating effect of motor velocity on the generation of torque ripple, the dynamics of the power amplifier is also taken into account in the commutation design as described in Section 23.8.

23.7 Modeling and Control of Motor Torque in Terms of Fourier Series

In rotary electric motors, the torque shape function in the expression of motor torque,

$$\tau = \sum_{k=1}^{p} i_k(\theta, \tau_d)\, y_k(\theta), \tag{23.29}$$

is a *periodic function*. Since successive phase windings are shifted by $2\pi/p$, we have the relationship,

$$y_k(\theta) = y\left(q\theta + \frac{2\pi(k-1)}{p}\right), \quad \forall\, k = 1, \ldots, p \tag{23.30}$$

where q is the number of motor poles. The structure of the electronic commutator is shown in Figure 23.10. The commutator commands the phase currents, i_k^*, through

$$i_k^*(\tau_d, \theta) = \tau_d\, u_k(\theta), \quad \forall\, k = 1, \ldots, p \tag{23.31}$$

where $u_k(\theta)$ is the *commutation shape function* associated with the kth phase. The individual phase control signals can be expressed based on the periodic *commutation function* $u(\theta)$, which is also a periodic

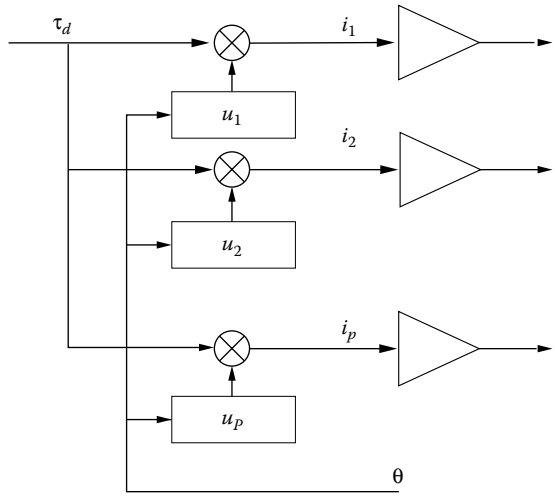

FIGURE 23.10 Electronic commutator.

function, that is,

$$u_k(\theta) = u\left(q\theta + \frac{2\pi(k-1)}{p}\right),$$

Since both $u(\theta)$ and $y(\theta)$ are periodic functions with position periodicity of $2\pi/q$, they can be approximated effectively via the truncated complex *Fourier series*

$$u(\theta) = \sum_{n=-N}^{N} c_n e^{jnq\theta}, \tag{23.32a}$$

$$y(\theta) = \sum_{m=-N}^{N} d_m e^{jmq\theta}, \tag{23.32b}$$

where $j = \sqrt{-1}$ and N can be chosen arbitrary large, but $2N/p$ must be an integer. Since both are real-valued functions, their negative Fourier coefficients are the conjugate of their real ones, $c_{-n} = \bar{c}_n$ and $d_{-n} = \bar{d}_n$. Furthermore, since the magnetic force is a conservative field for linear magnetic systems, the torque associated with the kth phase satisfies:

$$\oint \tau_k(\theta)\, d\theta = 0$$

which implies zero average torque over a period, and thus $c_0 = 0$.

The motor model and its control can be described by the vectors $c, d \in \mathbb{C}^N$ of the Fourier coefficients of $u(\theta)$ and $y(\theta)$, respectively, by

$$c = \text{col}\,(c_1, c_2, \ldots, c_N), \tag{23.33a}$$

$$d = \text{col}\,(d_1, d_2, \ldots, d_N). \tag{23.33b}$$

In the following, we seek a particular vector c for a given torque spectrum vector d so that the motor torque τ becomes ripple free, that is, independent of the motor angle θ. Assuming the current amplifiers deliver the demanded current instantaneously, that is, $i_k = i_k^\star$ for $k = 1, \ldots, p$. In this case,

after substituting Equations 23.30 through 23.32 into Equation 23.29, we arrive at

$$\tau = \tau_d \sum_{k=1}^{p} \sum_{\substack{n=-N \\ n\neq 0}}^{N} \sum_{\substack{m=-N \\ m\neq 0}}^{N} c_n d_m e^{j(n+m)(q\theta + \frac{2\pi(k-1)}{p})}. \tag{23.34}$$

This expression can be simplified by noting that the first summation vanishes when $l = m + n$ is not a multiple of p, that is,

$$\sum_{k=1}^{p} e^{jl\frac{2\pi(k-1)}{p}} = \begin{cases} p & \text{if} \quad l = \pm p, \pm 2p, \pm 3p, \ldots \\ 0 & \text{otherwise.} \end{cases} \tag{23.35}$$

Defining $\varrho := pq$, the torque expression (Equation 23.34) can be written in the following compact form:

$$\tau = \tau_d \, p \sum_{\substack{m=-N \\ m\neq 0}}^{N} \sum_{n=\lfloor(-N+m)/p\rfloor}^{\lfloor(N+m)/p\rfloor} d_m c_n e^{-jqm\theta} e^{j\varrho n\theta} \tag{23.36}$$

The expression of the torque in Equation 23.36 can be divided into two parts: the position-dependent torque, $\tau_{\text{rip}}(\theta, \tau_d)$, and the position-independent torque, $\tau_{\text{lin}}(\tau_d)$. That is,

$$\tau = \tau_{\text{lin}}(\tau_d) + \tau_{\text{rip}}(\theta, \tau_d), \tag{23.37}$$

in which

$$\tau_{\text{lin}}(\tau_d) = \tau_d k_0 \tag{23.38a}$$

$$\tau_{\text{rip}}(\theta, \tau_d) = \tau_d \sum_{l=-2N/p}^{2N/p} k_l \, e^{j\varrho l\theta}, \tag{23.38b}$$

where k_l is the Fourier coefficient of the motor torque, and can be calculated by

$$k_l = \begin{cases} p \sum_{n=1}^{N} c_n \bar{d}_{n-pl} + p \sum_{n=1}^{N-pl} \bar{c}_n d_{n+pl} & \text{if } l < \dfrac{N}{p} \\ p \sum_{n=pl-N}^{pl-1} c_n d_{pl-n} & \text{otherwise.} \end{cases} \tag{23.39}$$

The term k_0 in Equation 23.38a is the constant part of the circular convolution of $u(\theta)$ and $y(\theta)$. This, in turn, is equal to twice the real part of the inner product of the vectors c and d,

$$k_0 = 2p\text{Re}\langle c, d\rangle. \tag{23.40}$$

A ripple-free torque implies that all coefficients k_l except are zero and $k_0 \equiv 1$ so that $\tau \equiv \tau_d$. That is, the spectrum of the current excitation, c, must be calculated so that $k_0 = 1$ and $k_n = 0 \;\forall\; n = 1, \ldots, 2N/p$. This problem has infinitely many solutions. In this case, it is possible to minimize the power dissipation.

The average of dissipated power per unit command torque over one period, assuming constant speed, is

$$P_{\text{loss}} \propto \frac{1}{T}\int_0^T \|i(t)\|^2 \,\mathrm{d}t$$

By changing the integral variable from time t to θ, where $\mathrm{d}\theta = \omega\,\mathrm{d}t$ and $\omega T = 2\pi/q$, we have

$$P_{\text{loss}} \propto \frac{q}{2\pi}\sum_{k=1}^{P}\int_0^{2\pi/q} u_k^2(\theta)\,\mathrm{d}\theta, \tag{23.41}$$

where $\tau_d \equiv 1$. By virtue of Parseval's theorem, the power loss per unit commanded torque, that is, $\tau_d = 1$, is

$$P_{\text{loss}} \propto p\|c\|^2 \tag{23.42}$$

Remark 1

Minimizing power loss is tantamount to minimizing the Euclidean norm of the commutation spectrum vector $\|c\|$.

Consider the spectrum of the excitation current $c \in \mathbb{C}^N$ as the set of unknown variables. Then, according to Equation 23.39 and Remark 1, we must solve

$$\min \quad \|c\|^2 \tag{23.43a}$$

$$\text{subject to:} \quad Ac + B\bar{c} - \varsigma = 0, \tag{23.43b}$$

where $\varsigma \triangleq \text{col}(1, 0, \ldots, 0) \in \mathbb{R}^{2N/p+1}$, and matrices $A, B \in \mathbb{C}^{(2\frac{N}{P}+1)\times N}$ can be constructed from the torque spectrum vector. For example, for a three-phase motor ($p = 3$), the A and B matrices are given as

$$A = \begin{bmatrix} \bar{d}_1 & \bar{d}_2 & \bar{d}_3 & \bar{d}_4 & \bar{d}_5 & \cdots & \bar{d}_{N-1} & \bar{d}_N \\ d_2 & d_1 & 0 & \bar{d}_1 & \bar{d}_2 & \cdots & \bar{d}_{N-4} & \bar{d}_{N-3} \\ d_5 & d_4 & d_3 & d_2 & d_1 & \cdots & \bar{d}_{N-7} & \bar{d}_{N-6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{N-1} & d_{N-2} & d_{N-3} & d_{N-4} & d_{N-5} & \cdots & d_1 & 0 \\ 0 & 0 & d_N & d_{N-1} & d_{N-2} & \cdots & d_4 & d_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & d_N \end{bmatrix} \tag{23.44a}$$

$$B = \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & \cdots & d_N \\ d_4 & d_5 & d_6 & d_7 & d_8 & \cdots & 0 \\ d_7 & d_8 & d_9 & d_{10} & d_{11} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{N-2} & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{23.44b}$$

By separating real and imaginary parts, Equation 23.43a can be rewritten as

$$\underbrace{\begin{bmatrix} \text{Re}(A+B) & -\text{Im}(A-B) \\ \text{Im}(A+B) & \text{Re}(A-B) \end{bmatrix}}_{Q(d)} \begin{bmatrix} \text{Re}(c) \\ \text{Im}(c) \end{bmatrix} = \begin{bmatrix} \varsigma \\ 0 \end{bmatrix} \tag{23.45}$$

In general, for motors with more than two phases ($p > 2$), there are fewer equations than unknowns in Equation 23.45. Therefore, a unique solution is not expected. The *pseudo-inverse* offers the minimum-norm solution, that is, minimum $\|c\|$, which is consistent with the minimum power losses. Thus

$$c = \begin{bmatrix} I_N & jI_N \end{bmatrix} Q^{+} \begin{bmatrix} \varsigma \\ 0 \end{bmatrix}, \tag{23.46}$$

where Q^{+} represents the *pseudo-inverse* of matrix Q and I_N is the $N \times N$ identity matrix.

For convenience we represent the mapping (Equation 23.46) from the spectrum of the phase shape function d to that of the current excitation c in the compact form

$$c = \phi(d).$$

23.8 Modification of Commutation Law at High Velocity

Since motor phase currents are determined based on sinusoidal functions of the motor angle, high motor velocities result in a high drive frequency, which tend to become difficult for the current servo unit to track the reference current input. Therefore, design of ripple-free commutation at high velocities necessitates taking the dynamics of the current drives into account. Such a commutation is useful for velocity regulator applications where the velocity remains in the vicinity of an operating point.

With $h(t)$ defined as the impulse response of the current amplifiers, the actual and dictated phase currents are no longer identical, rather they are related by

$$i_k(t) = \int_0^t i_k^*(\zeta)\, h(t-\zeta)\, \mathrm{d}\zeta \quad \forall\, k = 1, \ldots, p. \tag{23.47}$$

After substituting Equations 23.31 and 23.32 into Equation 23.47, the total motor torque can be expressed as

$$\begin{aligned} \tau(\tau_d, \theta) &= \sum_{k=1}^{p} \left(\left(\sum_{m=-N}^{N} d_m e^{jm(q\theta(t) + 2\pi \frac{k-1}{p})} \right) \int_0^t \tau_d(\zeta) \sum_{n=-N}^{N} c_n e^{jn(q\theta(\zeta) + 2\pi \frac{k-1}{p})}\ h(t-\zeta)\mathrm{d}\zeta \right) \\ &= p \sum_{\substack{n=-N \\ n\neq 0}}^{N} \sum_{l=(n-N)/p}^{(n+N)/p} c_n d_{pl-n}\, e^{jql\theta} \left(\int_0^t \tau_d(\zeta) e^{-jq\omega n(t-\zeta)} h(t-\zeta)\, \mathrm{d}\zeta \right) \end{aligned} \tag{23.48}$$

in which, Equation 23.48 is obtained by using Equation 23.35 and assuming a constant velocity, that is, $\theta(t) - \theta(\zeta) = \omega(t-\zeta)$. The integral term in the RHS of Equation 23.48 can be written as the convolution integral, $\tau_d(t) * e^{-jq\omega nt} h(t)$, where function $e^{-jq\omega nt} h(t)$ can be interpreted as the impulse response of a virtual system associated with the nth harmonics. Then, the corresponding steady-state response to the step torque input response is given by $\tau_d H(jq\omega n)$, where $H(s)$ is the Laplace transform of function $h(t)$, that is, the amplifier's transfer functions. Now, define coefficients

$$c_n' \triangleq c_n H(jq\omega n) \quad \forall\, n = 1, \ldots, N. \tag{23.49}$$

and the corresponding vector $c' = \mathrm{col}\left(c_1', \ldots, c_N'\right)$ is related to vector c by

$$c' = D(\omega)c \quad \text{where} \quad D(\omega) = \mathrm{diag}\big(H(jq\omega), H(j2q\omega), \ldots, H(jNq\omega)\big). \tag{23.50}$$

The angular velocity variable ω in Equations 23.49 and 23.50 should not be confused with the frequency. Since $H(-jq\omega n) = \overline{H(jq\omega n)}$, the new coefficients satisfy

$$c_{-n}' = \overline{c_n'} \quad \forall\, n = 1, \ldots, N$$

Therefore, one can take the c_n''s as the Fourier coefficients of a commutation law that in the presence of actuator dynamics yields the same steady-state torque profile as the pervious case. This means

that all commutations that yield ripple-free torques at constant velocity ω must satisfy the constraint Equation 23.43b with c being replaced by c'. Furthermore, it will be shown in the following analysis that the power dissipation in the presence of amplifier's dynamics is proportional to $\|c'\|^2$. In view of actual excitation currents (Equation 23.47), the average power dissipation is

$$\begin{aligned} P_{\text{loss}} &\propto \sum_{k=1}^{p} \lim_{T\to\infty} \frac{1}{T} \int_0^T \left(\int_0^t \sum_{n=-N}^{N} c_n e^{jq\omega n\xi} h(t-\xi) \mathrm{d}\xi \right)^2 \mathrm{d}t \\ &= \lim_{T\to\infty} \frac{p}{T} \int_0^T \left(c_n e^{jqn\omega t} \int_0^t e^{-jqn\omega\nu} h(\nu) \mathrm{d}\nu \right)^2 \mathrm{d}t \\ &= \lim_{T\to\infty} \frac{p}{T} \int_0^T \left(c'_n e^{jqn\omega t} \right)^2 \mathrm{d}t \end{aligned} \tag{23.51}$$

It follows from a development similar to Equations 23.41 and 23.42 that

$$P_{\text{loss}} \propto p \left\| c' \right\|^2 \tag{23.52}$$

Therefore, the problem of finding c' that minimizes power dissipation and yields ripple-free torque at particular motor velocity ω can be similarly formulated as Equation 23.43 if c is replaced by c'. With c' in hand, the spectrum of actual commutation, c, can be obtained from the linear system (Equation 23.50) through matrix inversion.

23.8.1 Torque Transfer Function

The aim of this section is to derive the torque transfer function in the presence of amplifier dynamics. The position independent part of the generated torque is

$$\tau_{\text{lin}}(\tau_d) = p \sum_{\substack{n=-N \\ n\neq 0}}^{N} \bar{c}_n d_n \int_0^t \tau_d(\zeta) e^{-jq\omega n(t-\zeta)} h(t-\zeta)\, \mathrm{d}\zeta \tag{23.53}$$

$$g(t) = {}^*\tau_d(t) \tag{23.54}$$

where $*$ denotes the convolution integral and $g(t)$ is the impulse function of the system

$$g(t) = 2p \sum_{n=1}^{N} |a_n| \cos(q\omega nt + \angle a_n) h(t), \tag{23.55}$$

with $a_n = c_n \bar{d}_n$. Transforming function (Equation 23.55) into the Laplace domain, the system torque transfer function becomes

$$G(s) = p \sum_{n=1}^{N} a_n H(s + jq\omega n) + \bar{a}_n H(s - jq\omega n)), \tag{23.56}$$

where

$$G(s) = \frac{\tau_{\text{lin}}(s)}{\tau_d(s)}.$$

23.8.2 Simulation

Consider the electric motor described in Section 23.4 driving a mechanical load characterized by inertia 0.05 kg m^2 and viscous friction 5 Nm · s/rad. Also, assume that the motor and its electronically controlled

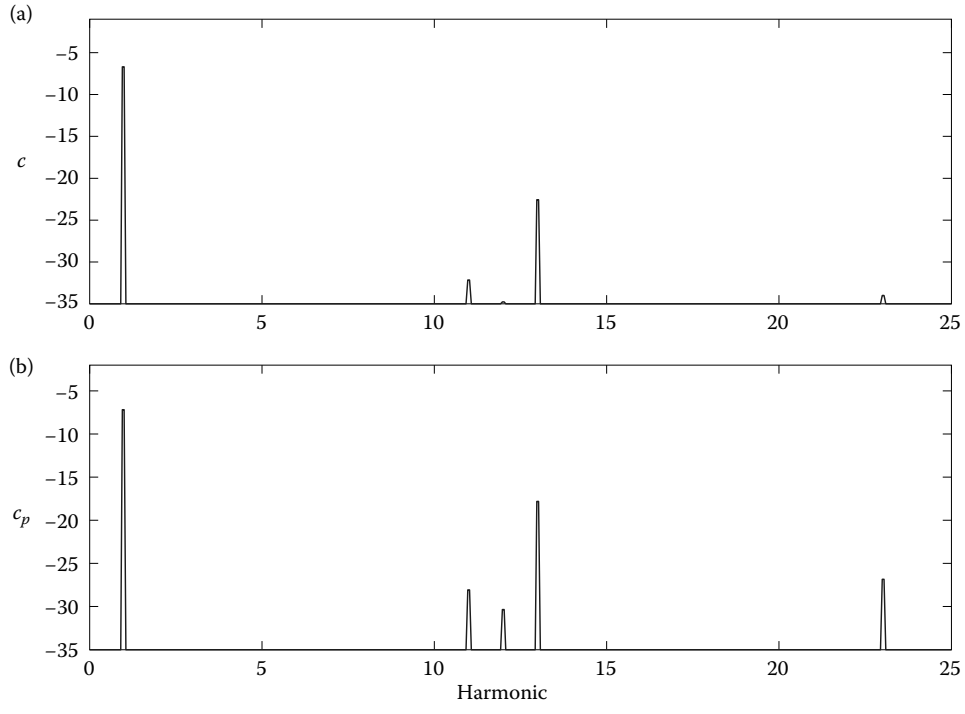

FIGURE 23.11 The spectrums of the first commutation (a) and the second one (b).

commutator is nested inside a PI velocity feedback loop where the controller gains are set to $K_p = 1$ Nm · s/rad and $K_i = 15$ Nm/rad. The transfer function of the current amplifier is assumed to be

$$H(s) = \frac{40s + 400}{s^2 + 40s + 400}$$

The control objective is to regulate the motor velocity at $\omega_d = 1$ rad/s. The spectrum of the commutation function without and with taking the frequency response of the amplifiers are plotted in Figures 23.11a and b, respectively. The step response of the closed-loop velocity controller with the two commutation schemes are illustrated in Figure 23.12. It is apparent from the graphs that velocity fluctuation can be eliminated only when the commutation law is designed with taking the amplifier dynamics into account.

23.9 Adaptive Reshaping the Excitation Currents of Brushless Motors

23.9.1 Introduction

The torque control problem is radically simplified when the motor's excitation currents are considered as inputs. In this case, the problem becomes a nonlinear mapping from desired torques and measured position to phase currents [8,10,13,14]. However, the main shortcoming associated with such an open-loop control is that its ability to cancel the position nonlinearity critically depends on the torque-angle profile of the motor phases. Many researchers have proposed different adaptive mechanisms to tune the motor controller during its operation. Chen [12] developed an adaptive linearization for a smooth motion control. Shouse et al. [18] applied a self-tuning tracking controller for permanent-magnet synchronous motors. In addition, different methods for the estimation of torque or the measurement of motor parameters

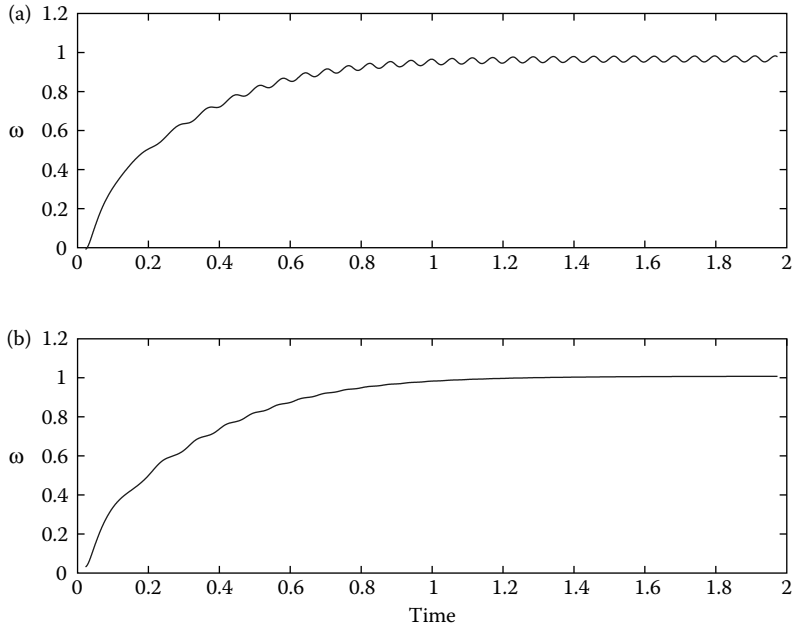

FIGURE 23.12 Step responses of the closed-loop PI velocity controller with the first commutation (a) and with the second one (b).

have been proposed in the literature. Delecluse et al. [19] proposed a measurement method based on the variation of the self and mutual inductance of a type of brushless motor. These parameters were then used to develop a nonlinear control strategy. Other researchers developed dynamical torque controllers by using either a torque observer or a flux observer to compensate torque ripple [7]. These adaptive control schemes are all motion controllers, and they heavily rely on the accurate modeling of the mechanical load attached to the motor.

In the previous sections, we presented a commutation law for precise torque control of brushless motors based on Fourier series analysis. In this approach, it was assumed that the mutual torque dominates the other torque components such as reluctance and detent torques. In the following sections, we present an adaptive self-tuning version of the torque control scheme [20]. An estimator is developed for the Fourier coefficients of motor torque function based on the measurement of phase voltage to reshape the excitation currents. The estimated Fourier coefficients are used by the commutation law, which simultaneously achieves accurate and ripple-free torque control and minimizes copper losses; see Figure 23.13. The stability of the entire control configuration is proved analytically. The results show that the actual torque converges to the command torque provided that the latter signal is bounded, and that the load system attached to the motor is stable. The advantages of this adaptive scheme are twofold: first, the estimator does not rely on the modeling of mechanical load, which may have complex dynamics. Second, it is shown that the motor torque under the proposed adaptive scheme asymptotically approaches the command torque regardless of the input trajectories. These advantages along with the self-tuning capability of the control scheme make the torque controller suitable for servo applications.

23.10 Modeling of Electric Motors in Terms of Inductance Matrix

According to Faraday's and Ohm's laws, the time varying magnetic flux linkage Ψ is related to the terminal voltages v and winding currents i by

$$\frac{d\Psi(i,\theta)}{dt} = -Ri + v, \tag{23.57}$$

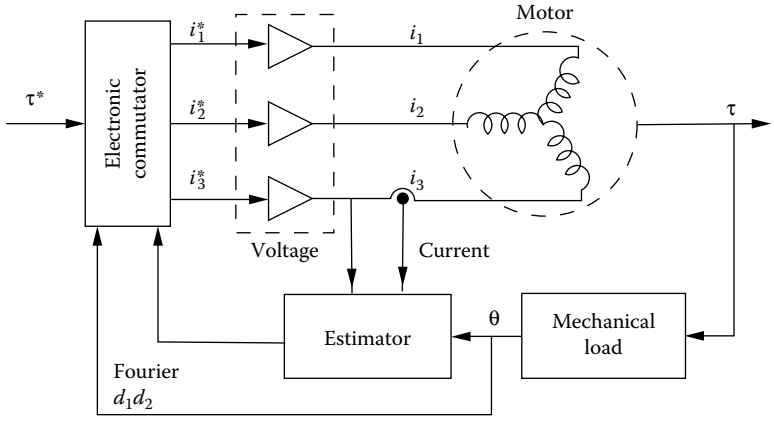

FIGURE 23.13 The architecture of the adaptive self-tuning torque controller.

where θ is the motor angle and $R = \text{diag}\{R_s, \ldots, R_r\}$ denotes the coil resistances. An electromagnetic machine is a device that converts input electric energy W_{ele} into output mechanical energy W_{mech}. This electromechanical energy conversion occurs through the medium of magnetic stored energy $W_{\text{fld}} = \int_0^{\Psi} i^T(\Psi', \theta)\, \mathrm{d}\Psi'$. Now, assuming the magnetic system is lossless, the first law of thermodynamics leads to

$$\mathrm{d}W_{\text{fld}} = \mathrm{d}W_{\text{ele}} - \mathrm{d}W_{\text{mech}} \tag{23.58}$$

According to Faraday's law and the principal of virtual work, we can find the expression for $\mathrm{d}W_{\text{ele}} = i^T v\, \mathrm{d}t = i^T \mathrm{d}\Psi$ and $\mathrm{d}W_{\text{mech}} = \tau\, \mathrm{d}\theta$. Introducing coenergy $W_{\text{co}}(i, \theta) = i^T \Psi - W_{\text{fld}}(\Psi, \theta)$, one can rewrite Equation 23.58 as

$$\mathrm{d}W_{\text{co}}(i, \theta) = \Psi^T \mathrm{d}i + \tau\, \mathrm{d}\theta \tag{23.59}$$

The total differential of $W_{\text{co}}(i, \theta) = \int_0^i \Psi^T(i', \theta)\, \mathrm{d}i'$, which, in general, is a function of the two independent variables i and θ, can be written as

$$\mathrm{d}W_{\text{co}}(i, \theta) = \frac{\partial W_{\text{co}}}{\partial i}\, \mathrm{d}i + \frac{\partial W_{\text{co}}}{\partial \theta}\, \mathrm{d}\theta \tag{23.60}$$

Comparing Equations 23.59 and 23.60 we obtain

$$\tau(i, \theta) = \frac{\partial W_{\text{co}}(i, \theta)}{\partial \theta} = \frac{\partial}{\partial \theta} \int_0^i \Psi^T(i', \theta)\, \mathrm{d}i' \tag{23.61}$$

Equations 23.57 and 23.61 govern the dynamics of most electromagnetic motors.

In general, the magnetic flux $\Psi(i, \theta)$ is a nonlinear function of the currents and the position. But for a linear electromagnetic system, we have

$$\Psi(\theta, i) = L(\theta) i, \tag{23.62}$$

where $L(\theta)$ is an inductance matrix. The form of the inductance matrix depends on the structure of the electric motor, but it is always symmetric and periodic in θ. We assume that the reluctance torque is negligible and that the phases are magnetically decoupled. The inductance matrix, then takes the form

$$L(\theta) = \begin{bmatrix} L_s & L_{sr}(\theta) \\ L_{sr}^T(\theta) & L_r \end{bmatrix}, \tag{23.63}$$

where $L_s = \text{diag}\{L_s'\}$ and $L_r = \text{diag}\{L_r'\}$ are the stator and rotor self-inductance matrices, and $L_{sr}(\theta)$ is the position-dependent mutual-inductance matrix; the subscripts s and r denote the stator and the rotor,

respectively. For a motor with p phases, the vectors of the phase currents and voltages are $i = \text{col}\left(i_s, i_r\right) \in \mathbb{R}^{2p}$ and $v = \text{col}\left(v_s, v_r\right) \in \mathbb{R}^{2p}$. If the magnetic field is produced by permanent magnet, we consider the equivalent rotor current $i_r = \text{constant}$. With these assumptions, the torque equation (Equation 23.61) can be rewritten as

$$\begin{aligned}\tau(i,\theta) &= \frac{1}{2} i^T \left(\frac{\partial L}{\partial \theta}\right) i \\ &= \sum_{k=1}^{p} y_k(\theta) i_{s_k},\end{aligned} \tag{23.64}$$

where $y_k(\theta) = \sum_{j=1}^{p} \frac{\partial}{\partial \theta} L_{sr_{kj}}(\theta) i_{r_j}$ are expressed by the so-called *torque shape function*.

The following section outlines how to design the excitation currents i_{s_k} so as to achieve a desired torque by making use of Equation 23.64 and with the assumption that the torque shape function $y_k(\theta)$ is perfectly known. This assumption will be relaxed in Section 23.11 where the shape function is estimated in real time by making use of the voltage equation (Equation 23.57).

23.11 Adaptive Control

23.11.1 Voltage Dynamic Equation

Substituting Equation 23.62 into 23.57, we arrive at a set of independent differential equations. Without loss of generality we consider only the equation associated with the first phase, that is

$$v_{s_1} = R_s i_{s_1} + L_s' \frac{\mathrm{d}i_{s_1}}{\mathrm{d}t} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{k=1}^{p} L_{sr_{1k}}(\theta) i_{r_k}\right). \tag{23.65}$$

Using the chain rule and noting that $i_r = \text{constant}$, we obtain

$$\begin{aligned}\sum_{k=1}^{p} L_{sr_{1k}}(\theta) i_{r_k} &= \int_0^{\theta} y_1(\xi)\, \mathrm{d}\xi \\ &= \sum_{\substack{n=-N \\ n\neq 0}}^{N} \frac{-j\mathrm{d}_n}{qn} e^{jqn\theta}\end{aligned} \tag{23.66}$$

Using Equation 23.66 in Equation 23.65 and rearranging the resultant equation yields

$$\left(1 + \kappa \frac{\mathrm{d}}{\mathrm{d}t}\right) R_s i_{s_1} = v_{s_1} + \frac{\mathrm{d}}{\mathrm{d}t} \sum_{\substack{n=-N \\ n\neq 0}}^{N} \frac{j\mathrm{d}_n}{qn} e^{jqn\theta(t)}, \tag{23.67}$$

where

$$\kappa = \frac{L_s'}{R_s}$$

is the time constant of the electrical system.

23.11.2 Self-Tuning Control

Defining stable and proper filters $\breve{G}_1(s) = 1/(1+\kappa s)$ and $\breve{G}_2(s) = s/(1+\kappa s)$, we can rewrite Equation 23.67 in terms of filtered signals as

$$R_s i_{s_1} - G_1 * v_{s_1} = G_2 * \left(\sum_{\substack{n=-N \\ n\neq 0}}^{N} \frac{j\mathrm{d}_n}{qn} e^{jqn\theta(t)} \right), \tag{23.68}$$

where $G_1(t)$ and $G_2(t)$ are the *impulse responses* of the corresponding filers, and $*$ stands for the convolution integral. The dynamics equation (Equation 23.68) can be linearly parameterized as

$$G_2 * Y^T \rho = \frac{1}{2} q (R_s i_{s_1} - G_1 * v_{s_1}) := v_F \tag{23.69}$$

where

$$Y(t) := \begin{bmatrix} \sin(q\theta(t)) \\ \frac{1}{2}\sin(2q\theta(t)) \\ \vdots \\ \frac{1}{N}\sin(Nq\theta(t)) \\ \cos(q\theta) \\ \frac{1}{2}\cos(2q\theta(t)) \\ \vdots \\ \frac{1}{N}\cos(qN\theta(t)) \end{bmatrix}, \tag{23.70}$$

and where vector $\rho = \mathrm{col}\left(\mathrm{Re}(d),\ \mathrm{Im}(d)\right)$ contains the parameters of interest, and v_F is the filtered version of the phase voltage.

The control architecture for a three-phase brushless motor is depicted in Figure 23.13. Recall that the electronic commutator modulates the torque commands with the periodic commutation function according to

$$i_k^\star = u\left(q\theta + \frac{2\pi(k-1)}{3}\right)\tau_d \quad \forall\, k = 1, \ldots, 3,$$

where the periodic function $u(\cdot)$ is expressed by a Fourier series as

$$u(\theta) = \sum_{n=1}^{N} c_n e^{jnq\theta}$$

and the Fourier coefficients c_n's are determined based on the Fourier coefficients of the motor shape function, d_n's, through mapping

$$c = \phi(d) \tag{23.71}$$

so that the motor generates torque as requested (as described in the previous sections). Let $\hat{\rho}$ and $\tilde{\rho} = \rho - \hat{\rho}$ denote the estimated parameters and the parameter errors, respectively. Considering i_k as the control

inputs, we propose the following control law:

$$i_k = \tau_d u_k\big(\phi(\hat{\rho}), \theta\big) \quad \forall\, k = 1, \ldots, p, \tag{23.72}$$

in conjunction with the following parameter update law:

$$\dot{\hat{\rho}} = \gamma(G_2 * Y)\sigma, \tag{23.73}$$

where

$$\sigma = \frac{1}{2} q R_s \tau_d u_1\big(\phi(\hat{\rho}), \theta\big) - \frac{1}{2} q G_1 * v_{s_1} - (G_2 * Y^T)\hat{\rho}, \tag{23.74}$$

and $\gamma > 0$ is the estimator gain.

Proposition 1:

The control law (Equation 23.72*) together with the parameter update law (Equation* 23.73*) ensures the following properties:* $\sigma \in \mathcal{L}_2 \cap \mathcal{L}_\infty$ *and* $\tilde{\rho}, \dot{\tilde{\rho}} \in \mathcal{L}_\infty$*, where* $\mathcal{L}_\infty$ *and* $\mathcal{L}_2$ *indicate the spaces of bounded and square integrable signals, respectively.*

Proof. It can be readily inferred from Equations 23.69 and 23.74 that

$$\sigma = (G_2 * Y^T)\tilde{\rho}. \tag{23.75}$$

Choose the following positive function:

$$V = \frac{1}{2}\tilde{\rho}^T \gamma^{-1} \tilde{\rho}. \tag{23.76}$$

Using Equation 23.75 in the time derivative of V along trajectories of Equation 23.73 gives

$$\dot{V} = -\sigma^2.$$

A standard argument proves the proposition. ■

The rest of this section deals with stability analysis of the proposed adaptive law. The final objective is to prove that the motor torque τ tends to the command torque τ_d under the control law. In other words, the torque tracking error $e = \tau - \tau_d$ asymptotically converges to zero. Let us define

$$\tilde{y}(\rho, \hat{\rho}, \theta) \triangleq y(\rho, \theta) - y(\hat{\rho}, \theta)$$

as the difference between actual shape function and the one obtained based on the estimated Fourier coefficients. In the following analysis, we will show that under mild conditions, the error $\tilde{w}$ converges to zero even though $\hat{\rho}$ does not necessarily tend to ρ. This is essential for subsequently proving that e is uniformly stable. Assume that

A1 the motor velocity $\omega = \mathrm{d}\theta/\mathrm{d}t$ is bounded, but not identically zero.
A2 torque set point τ_d is bounded

Corollary 1:

Under the above assumptions, the following hold:

1. *Error $\tilde{w}$ is uniformly stable*
2. *Torque tracking error e asymptotically converges to zero.*

Proof. First, we must show that σ converges to zero. Asymptotic stability of σ can be inferred from Proposition 1, if $\dot{\sigma}$ is shown to be bounded. The time derivative of Equation 23.75 leads to

$$|\dot{\sigma}| \leq a_1 \|Y\|_\infty \, \|\dot{\tilde{\rho}}\|_\infty + a_1 \|\dot{Y}\|_\infty \, \|\tilde{\rho}\|_\infty, \tag{23.77}$$

where

$$a_1 = \max\left(\frac{1}{\kappa^2}, \frac{2}{\kappa} - \frac{1}{\kappa^2}\right)$$

is the *peak gain** of the filter $\breve{G}_2(s)$, that is, $\|\breve{G}_2(s)\|_{\text{pk}-\text{gn}} = \|G_2\|_1 = a_1$. Since $\tilde{\rho}$ and $\dot{\tilde{\rho}}$ are bounded variables, we need only show that Y and $\dot{Y}$ are bounded. Moreover, knowing that sinusoidal functions are bounded by unity, we can obtain conservative bounds for Y and its time derivative as

$$\|Y\| \leq 2\sum_{n=1}^{N} \frac{1}{n}, \quad \text{and} \quad \|\dot{Y}\| \leq 2Nq|\omega|. \tag{23.78}$$

Thus $Y, \dot{Y} \in \mathcal{L}_\infty$, which implies that all terms on the RHS of Equation 23.77 are bounded, and hence $\dot{\sigma} \in \mathcal{L}_\infty$. This result together with Proposition 1 implies that $\sigma \to 0$ as $t \to \infty$. Moreover, it can be inferred from Equation 23.73 that

$$\|\dot{\tilde{\rho}}\| \leq a_2|\sigma|,$$

where $a_2 = a_1\gamma\|Y\|$. Since σ asymptotically converges to zero, so does $\dot{\tilde{\rho}}$. It follows that $\hat{\rho}$ tends to a constant vector, which may not necessarily be ρ. Since $\breve{G}_2(s) = s\breve{G}_1(s)$ we can rewrite Equation 23.75 as

$$\sigma = (G_1 * \dot{Y}^T)\tilde{\rho}. \tag{23.79}$$

Now with $\tilde{y} = y(\tilde{\rho}, \theta)$ in the time derivative of Equation 23.78, we obtain $\dot{Y}\tilde{\rho} = \omega\tilde{y}$. Using this result together with the fact that $\tilde{\rho}$ converges to a constant vector in Equation 23.79 leads to following: $G_1 * \big(\omega \cdot \tilde{y}(\theta)\big) \to \sigma$ as $t \to \infty$. Furthermore, since σ converges to zero, we see that

$$G_1 * \big(\omega\, \tilde{y}(\theta)\big) \to 0 \quad \text{as} \quad t \to \infty, \tag{23.80}$$

which implies that $\omega\tilde{y} \to 0$ according to the Final Value Theorem and to the fact that $\breve{G}_1(0) \neq 0$. Finally, by virtue of Assumption A1, we can conclude $\tilde{y}(\theta) \to 0$ as $t \to \infty$.

Now, we are ready to prove the second argument of the corollary. Suppose the spectrum of the excitation currents are redesigned based on the estimated Fourier coefficients. Then, the mapping from

* The peak gain of a LTI system H is defined as

$$\|H\|_{\text{pk}-\text{gn}} := \sup_{\|w\|_\infty \neq 0} \frac{\|Hw\|_\infty}{\|w\|_\infty},$$

which is equal to $\mathcal{L}_1$ norm of its impulse response, that is, $\|\breve{H}\|_{\text{pk}-\text{gn}} = \|H\|_1$.

the command torque to the actual torque τ can be expressed by

$$\tau = \tau_d \sum_{k=1}^{p} u_k\big(\phi(\hat{\rho}), \theta\big) y_k(\rho, \theta). \tag{23.81}$$

Notice that the summation on the RHS of Equation 23.81 is identically unity if $\rho = \hat{\rho}$, that is,

$$\sum_{k=1}^{p} u_k\big(\phi(\hat{\rho}), \theta\big) \cdot y_k(\hat{\rho}, \theta) = 1 \quad \forall\, \theta \in \mathbb{R}, \forall\, \hat{\rho} \in \mathbb{R}^{2N}. \tag{23.82}$$

From Equations 23.82 and 23.81, we can calculate the torque error as

$$\begin{aligned} e &= \tau_d - \tau_d \sum_{k=1}^{p} u_k\big(\phi(\hat{\rho}), \theta\big) \cdot y_k(\rho, \theta) \\ &= \tau_d \sum_{k=1}^{p} u_k\big(\phi(\hat{\rho}), \theta\big)\big(w_k(\hat{\rho}, \theta) - y_k(\rho, \theta)\big) \\ &= \tau_d \sum_{k=1}^{p} u_k\big(\phi(\hat{\rho}), \theta\big) y_k(\tilde{\rho}, \theta). \end{aligned} \tag{23.83}$$

By virtue of Equations 23.36, 23.46, and 23.83, we can find a conservation bound on the torque error as

$$\begin{aligned} |e| &\le 2pN \|\phi(\hat{\rho})\|_\infty |\tau_d|\, |\tilde{y}| \\ &\le 2N \|Q^+(\hat{\rho})\|\, |\tau_d|\, |\tilde{y}| \end{aligned} \tag{23.84}$$

According to Assumption A2 all terms on the RHS of inequality Equation 23.84 are bounded. Moreover, since $\tilde{w}$ tends to zero, we can conclude that $e \to 0$ as $t \to \infty$. ■

23.11.3 Input/Output Stable Mechanical Loads

The following argument shows that the velocity remains bounded if command torque τ_d retains bounded. Let's M represent the dynamics of the mechanical load system, where the input and output are τ and ω, respectfully, that is, $\omega = M\tau$; typically $\breve{M}(s) = 1/(Js + b)$ where J and b are the inertia of the motor rotor and viscous friction in the bearings, respectively. It is now reasonable to assume that the mechanical load is an *input–output bounded* system, that is

$$|\omega| \le a_3 |\tau|, \tag{23.85}$$

where $a_3 = \|\breve{M}\|_{\text{pk}-\text{gn}}$ is the peak-gain of the mechanical system. It can be inferred from Equation 23.46 that

$$\|\phi(\hat{\rho})\| \le \frac{1}{p\sigma_{\min}(Q(\hat{\rho}))} \tag{23.86}$$

Using inequalities

$$\|u_k(\phi(\hat{\rho}), \theta)\| \le \|\phi(\hat{\rho})\|, \quad \|y_k(\rho, \theta)\| \le \|\rho\| \quad \forall\, k = 1, \ldots, p,$$

and Equation 23.86 in Equation 23.81, we can obtain a conservative bound on the generated torque as

$$|\tau| \le \frac{\|\rho\|}{\sigma_{\min}(Q(\hat{\rho}))}\, |\tau_d| \tag{23.87}$$

It is evident from Equations 23.85 and 23.87 that bounded τ_d means bounded velocity. It should be pointed out that without a mechanical load, there is no guarantee of a bounded velocity, and hence no

stable torque error. However, the mechanical load consists of the external load attached to the motor plus the motor's rotor so that, with no external load the rotor of the motor becomes the only mechanical load. The rotor system is input/output bounded if there is damping, that is, nonzero friction in the motor bearings.

Additionally, assuming that the summation terms $\sum_{k=1}^{p} u_k(\phi(\hat{\rho}), \theta) y_k(\rho, \theta)$ in Equation 23.81 remain nonzero during the adjustment period, we can say that τ and hence ω are not identically zero. Therefore, Assumption A2 automatically satisfies Assumption A1 provided that the mechanical load is an input-to-output bounded system and that τ_d is not identically zero.

23.12 Experiment

In order to evaluate the performance of the self-tuning torque controller, experiments were conducted on a three-phase synchronous motor with 9 pole pairs. The actual values of the winding resistance and self-inductance are measured by a Wheatstone bridge instrument, see Table 23.1.

Although, the Fourier coefficients of the phase shape function are adaptively estimated by the controller, we also measured the actual coefficients for comparison. To this end, as described in our previously related work [4], we characterized the motor phase shape function by using a specially designed dynamometer setup. Since the motor has nine pole pairs, the torque trajectory is periodic in position with a fundamental spatial-frequency of 9 cpr and thus the torque pattern repeats every 40 degrees. The harmonics of the spatial frequency content of the torque shape function are shown in Figure 23.14b—for example, the

TABLE 23.1 Electric Parameters of the Motor

Selfinductance (mH)	Resistance (Ω)	Time Constant
12.5	2.54	0.0049

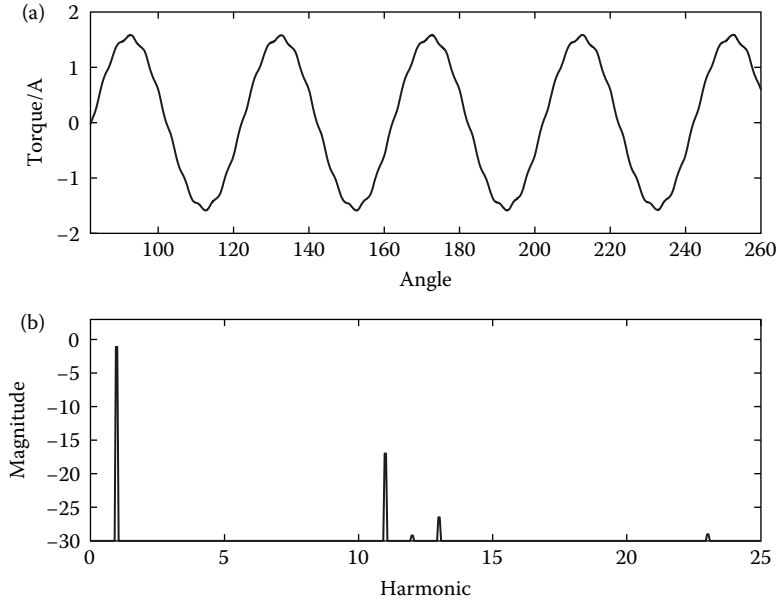

FIGURE 23.14 Torque shape function (a) and its special harmonics (b).

TABLE 23.2 Torque Harmonics of Motor

Harmonics No.	Complex Value (Nm/A)	Magnitude (Nm/A)
1st	$0.270 + 0.720j$	0.769
11th	$0.015 + 0.015j$	0.0212
13th	$0.003 + 0.002j$	0.0036

spatial frequency of the 11th harmonic is 99 cpr. It is clear from the graph that the significant frequency components appear at the 1st, 11th and 13th harmonics; see Table 23.2.

The adaptive self-tuning torque controller was implemented for this motor setup. Unlike our previous experiment, the torque controller does not require *a priori* knowledge of the motor's shape function. The torque controller was nested inside a PID position feedback, which was commanded to follow a sinusoidal reference position trajectory $\theta^\star = 0.44\sin 3\pi t$. To limit the number of the estimated parameters, only the

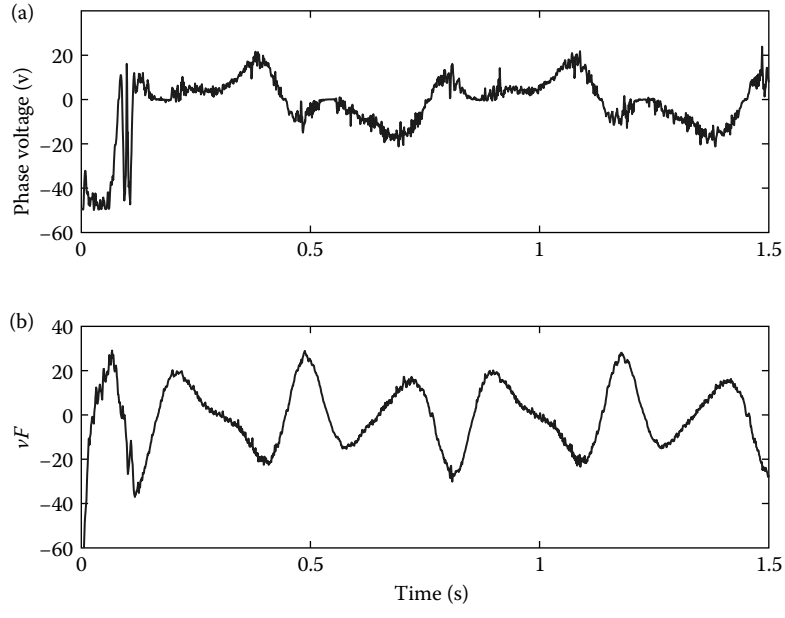

FIGURE 23.15 Phase voltage (a) and its filtered version (b).

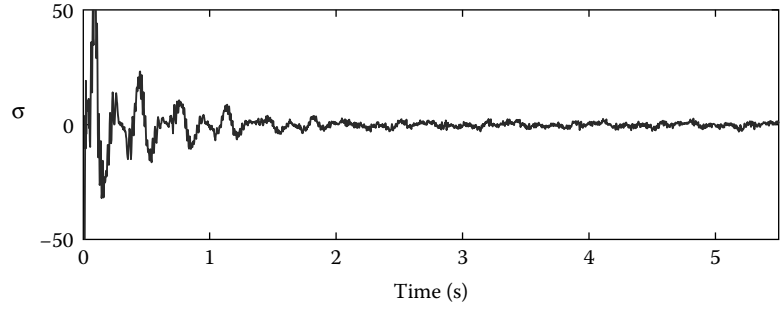

FIGURE 23.16 The history of voltage error σ.

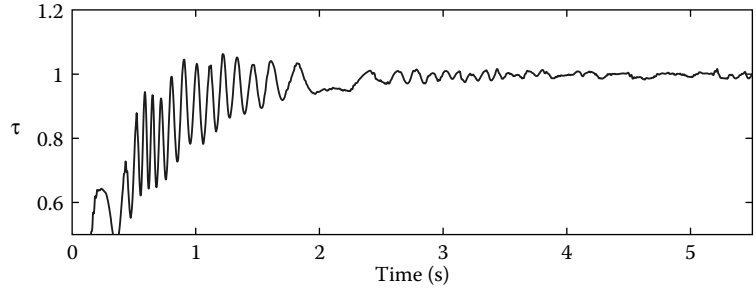

FIGURE 23.17 The history of the ratio of actual torque to command torque.

coefficients associated with the major torque harmonics, that is, 1st, 11th and 13th, are considered in the parameter update law. Figures 23.15 and 23.16 show trajectories of the phase voltage and its filtered version, respectively. The electromagnetic torque developed by the motor cannot be directly measured, but it can be calculated from Equation 23.81 based on the actual parameters obtained from the dynamometer. From the estimated and the actual Fourier coefficients, we can compute the ratio of the

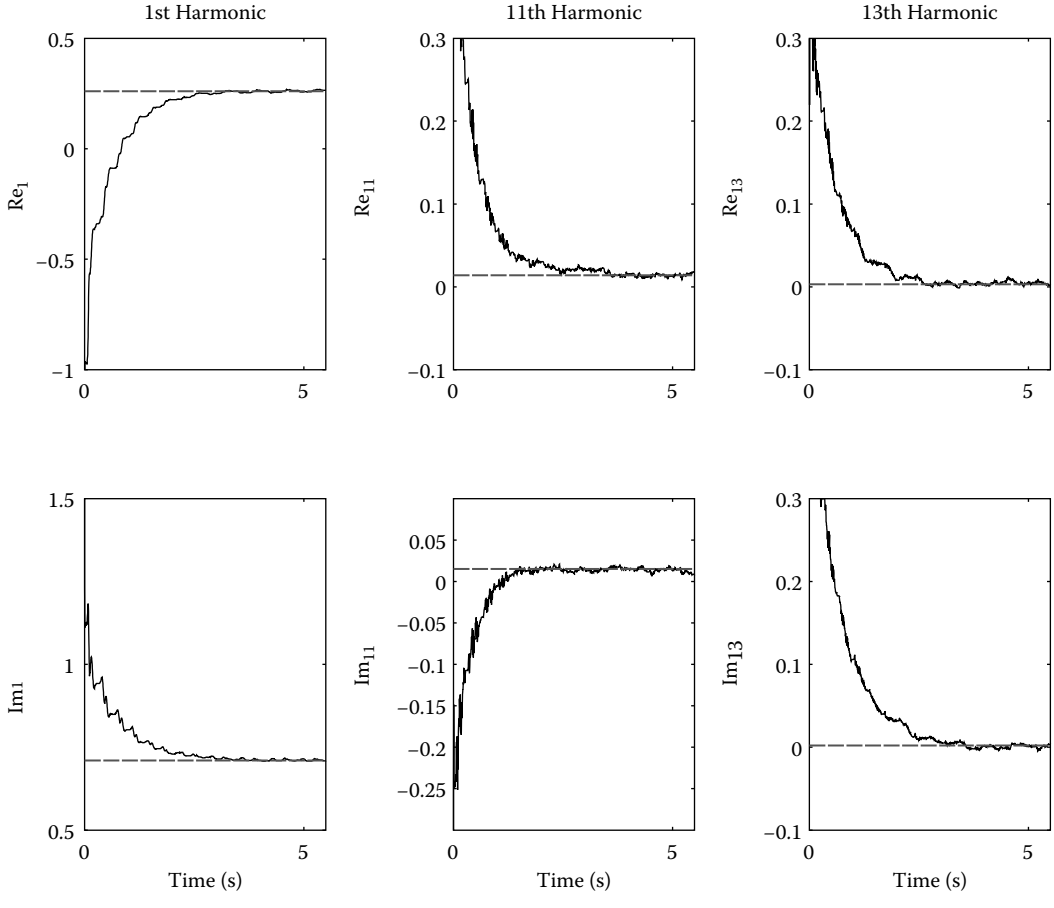

FIGURE 23.18 The history of the estimated parameters.

actual-to-command torques as

$$\frac{\tau}{\tau_d} = \sum_{k=1}^{p} u_k(\phi(\hat{\rho}), \theta) w_k(\rho, \theta).$$

The time history of the torque ratio is plotted in Figure 23.17. It is evident from the figure that after the tuning period, the motor torque tracks the command torque with a 2% accuracy, and that the controller tunes itself after a few seconds. The time history of the estimated Fourier coefficients is also illustrated in Figure 23.18. The convergence of the estimated parameters implies the richness of the input signals.

References

1. A. E. Fitzgerald, C. Kinsley, and A. Kusko, *Electric Machinery*. New York: McGraw-Hill Book Company, 1971.
2. P. C. Krause, *Analysis of Electric Machinery*. New York: McGraw-Hill, 1986.
3. K. Ramu, *Permanent Magnet Synchronous and Brushless DC Motors (Mechanical Engineering)*. New York: CRC Press, 2009.
4. F. Aghili, M. Buehler, and J. M. Hollerbach, Experimental characterization and quadratic programming-based control of brushless-motors, *IEEE Trans on Control Systems Technology*, vol. 11, no. 1, pp. 139–146, 2003.
5. S. J. Park, H. W. Park, M. H. Lee, and F. Harashima, A new approach for minimum-torque-ripple maximum-efficiency control of BLDC motor, *IEEE Trans. on Industrial Electronics*, vol. 47, no. 1, pp. 109–114, February 2000.
6. Y. Murai, Y. Kawase, K. Ohashi, and K. Okuyamz, Torque ripple improvement for brushless DC miniature motors, *IEEE Trans. Industry Applications*, vol. 25, no. 3, pp. 441–449, 1989.
7. N. Matsui, T. Makino, and H. Satoh, Autocompensation of torque ripple of direct drive motor by torque observer, *IEEE Trans. on Industry Application*, vol. 29, no. 1, pp. 187–194, January–February 1993.
8. D. G. Taylor, Nonlinear control of electric machines: An overview, *IEEE Control Systems*, vol. 14, no. 6, pp. 41–51, 1994.
9. C. French and P. Acarnley, Direct torque control of permanent magnet drives, *IEEE Trans. on Industry Applications*, vol. 32, no. 5, pp. 1080–1088, September–October 1996.
10. F. Aghili, M. Buehler, and J. M. Hollerbach, Optimal commutation laws in the frequency domain for PM synchronous direct-drive motors, *IEEE Transactions on Power Electronics*, vol. 15, no. 6, pp. 1056–1064, November 2000.
11. Y. Wang, D. Cheng, C. Li, and Y. Ge, Dissipative hamiltonian realization and energy-based L2-disturbance attenuation control of multimachine power systems, *IEEE Trans. on Automatic Control*, vol. 48, no. 8, pp. 1428–1433, August 2003.
12. D. Chen and B. Paden, Adaptive linearization of hybrid step motors: Stability analysis, *IEEE Trans. Automatic Control*, vol. 38, no. 6, pp. 874–887, June 1993.
13. H. Le-Huy, R. Perret, and R. Feuillet, Minimization of torque ripple in brushless dc motor drives, *IEEE Trans. Industry Applications*, vol. 22, no. 4, pp. 748–755, August 1986.
14. Ha and Kang, Explicit characterization of all feedback linearizing controllers for a general type of brushless dc motor, *IEEE Trans. Automatic Control*, vol. 39, no. 3, pp. 673–677, 1994.
15. M. Ilic'-Spong, R. Marino, S. M. Peresada, and D. G. Taylor, Feedback linearizing control of switched reluctance motors, *IEEE Trans Automatic Control*, vol. AC-32, no. 5, pp. 371–379, 1987.
16. H. W. Kuhn and A. W. Tucker, Nonlinear programming, in *Proc. Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley: University of California Press, pp. 481–492, 1951.
17. F. Aghili, M. Buehler, and J. M. Hollerbach, A modular and high-precision motion control system with an integrated motor, *IEEE/ASME Trans. on Mechatronics*, vol. 12, no. 3, pp. 317–329, June 2007.
18. K. R. Shouse and D. G. Taylor, A digital self-tuning controller for permanent-magnet synchronous motors, *IEEE Trans. Control Systems Technology*, vol. 2, no. 4, pp. 412–422, December 1994.
19. C. Delecluse and D. Grenier, A measurement method of the exact variations of the self and mutual inductances of a buried permanent magnet synchronous motor and its application to the reduction of torque ripples, in *5th International Workshop on Advanced Motion Control*, Coimbra, June 29–July 1, pp. 191–197, 1998.
20. F. Aghili, Adaptive reshaping of excitation currents for accurate torque control of brushless motors, *IEEE Trans. on Control System Technologies*, vol. 16, no. 2, pp. 356–364, March 2008.

24

Hybrid Model Predictive Control of the Boost Converter

Raymond A. DeCarlo
Purdue University

Jason C. Neely
Purdue University

Steven D. Pekarek
Purdue University

24.1 Introduction

Dc–dc converters are widely used in modern energy conversion systems. Examples of their application include power supplies and hybrid electric vehicles (HEVs), in addition to military, space, and industrial power systems. Regulation of the converter output voltage is accomplished through control of the on/off behavior of the semiconductor switch(es). Dc–dc converters have both continuous and discrete (switching) dynamics, thus making them hybrid systems.

Although the dynamics of a converter are hybrid (change with switch-state), a common step in existing control approaches is to create an average-value model (AVM) of a converter in which the dynamics of switching are averaged [1,2]. In average-value form, the slow circuit dynamics are modeled as a continuous-time system. Once an AVM is established, any one of a number of techniques developed for the control of linear or nonlinear continuous-time systems can be applied.

In this chapter, recently developed hybrid optimal control theory (HOCT) is adapted to the control of dc–dc converters. Specifically, control of switching in a dc–dc boost converter is determined by solving the so-called embedded optimal control problem (EOCP). The EOCP is discretized and implemented via a nonlinear model predictive control (MPC) scheme termed hybrid model predictive control (HMPC).

MPC is a discrete-time control strategy wherein the control is obtained at each sampling through finite-time horizon optimization using the present (measured) state of the system as the initial state. MPC requires a dynamical model to predict system response and a user-defined performance index (PI) to be optimized. At each sampling instant, the converter state is measured, which initiates the computation of piecewise constant switching control values over a finite horizon window so that a PI is minimized

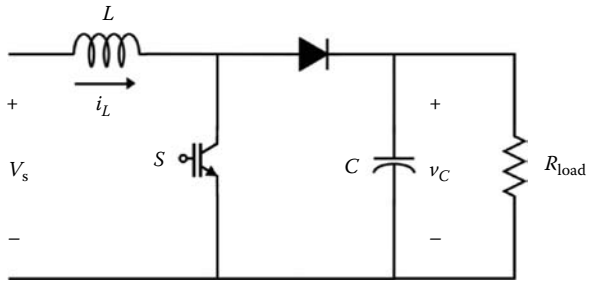

FIGURE 24.1 The boost converter produces a voltage higher than the source voltage. Regulation of the output voltage is accomplished through control of the on/off sequence of the switch state $s \in \{0, 1\}$.

over that window. The first control in this sequence is then applied to the converter, and the process is repeated.

In this chapter, HMPC is described for the real-time control of a boost converter (Figure 24.1), which is a circuit that produces an output voltage greater than the source voltage, $v_C > V_s$.

24.2 Hybrid State Model of the Boost Converter

From Figure 24.1, one observes that when $s = 1$, the inductor current increases, and the output voltage decays with time constant $\tau = R_{load}C$. When $s = 0$, energy stored in the inductor is used to charge the capacitor; inductor current decreases, and output voltage increases. If the switch remains *off* and the inductor current falls to zero, the converter enters a third mode of operation termed *discontinuous conduction*. For the control described herein, only two-mode circuit operation is allowed and maintained through constraints on the switch function $s(t)$.

Under the condition that discontinuous conduction is disallowed, the boost converter of Figure 24.1 can be modeled as a two-mode system determined by the switch state $s(t) \in \{0, 1\}$, "0" denoting "open" or "off" and "1" denoting "closed" or "on." In each switch position, the converter has a distinct linear state model with state vector $x = [i_L \;\; v_C]^T$ as follows:

$$\text{mode } s = 0: \;\; \dot{x}(t) = A_0 x(t) + b_0, \tag{24.1}$$

$$\text{mode } s = 1: \;\; \dot{x}(t) = A_1 x(t) + b_1, \tag{24.2}$$

with $x(t_0) = x_0$, where $t \geq t_0$ and system matrices are given by

$$A_0 = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{R_{load}C} \end{bmatrix}, \quad b_0 = \begin{bmatrix} \frac{V_s}{L} \\ 0 \end{bmatrix}, \tag{24.3}$$

$$A_1 = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_{load}C} \end{bmatrix}, \quad b_1 = \begin{bmatrix} \frac{V_s}{L} \\ 0 \end{bmatrix}. \tag{24.4}$$

It is noted for this system that V_s is not a control input. The hybrid model is expressed compactly as

$$\dot{x}(t) = (1 - s(t)) \cdot [A_0 x(t) + b_0] + s(t) \cdot [A_1 x(t) + b_1]. \tag{24.5}$$

Optimization of system (Equation 24.5) with respect to the switched control comprises the so-called switched optimal control problem (SOCP): find $s(t)$ to minimize a PI P_s subject to the model. However, the optimization is not convex for $s(t) \in \{0, 1\}$.

Derivation of the control begins by formulating the continuous-time EOCP for the boost converter [3]. A discrete-time form of the optimization is then developed and implemented as a nonlinear MPC strategy.

24.3 Formulating the Continuous-Time EOCP

The EOCP is formulated in two steps. First, an embedded form of Equation 24.5 is established wherein $s(t) \in \{0, 1\}$ is replaced with $\tilde{s}(t) \in [0, 1]$:

$$\dot{x}(t) = (1 - \tilde{s}(t)) \cdot [A_0 x(t) + b_0] + \tilde{s}(t) \cdot [A_1 x(t) + b_1], \tag{24.6a}$$

$$\tilde{s}(t) \in [0, 1], \tag{24.6b}$$

$$i_L(t) > 0, \tag{24.6c}$$

where Equation 24.6c maintains continuous conduction. Essentially, Equation 24.6b embeds the original switched system of Equation 24.5 into a continuously parameterized family of problems. The trajectories of system (Equation 24.5) are dense in those of system (Equation 24.6) [3].

Next, an embedded PI is set forth

$$P_s(t_0, t_f, x_0, \tilde{s}) = \int_{t_0}^{t_f} [\tilde{s}(t) F_1(t, x(t)) + (1 - \tilde{s}(t)) F_0(t, x(t))]\, dt, \tag{24.7}$$

where $F_0 \in C^1$ and $F_1 \in C^1$ determine the desired performance in the respective modes of operation, and $t \in [t_0, t_f]$ is the finite time interval for the optimization. The PI may also include a terminal penalty function [3], but a terminal penalty is not utilized in the system described here. With the embedded model and embedded PI defined, the EOCP is

$$\underset{\tilde{s}(t)}{\text{minimize}}\, P_s(t_0, t_f, x_0, \tilde{s}) \tag{24.8}$$

subject to Equations 24.6.

24.3.1 Performance Index for Boost Converter

The purpose of the controller is to track a commanded output voltage V_C^* with corresponding steady-state inductor current I_L^*. Integrands that have been validated experimentally (among many other possible user choices) are

$$F_0 = F_1 = C_I (i_L(t) - I_L^* - K(V_C^* - v_C(t)))^2 + C_V (v_C(t) - V_C^*)^2, \tag{24.9}$$

where we note that

$$I_L^* = \frac{(V_C^*)^2}{V_s R_{\text{load}}} \tag{24.10}$$

is obtained by equating *input power* and output *power* in steady state. The weighting constants $C_I \in \mathrm{R}^+$, $K \in \mathrm{R}^+$, and $C_V \in \mathrm{R}^+$ allow for adjustment of the tracking performance. Specifically, tracking performance depends on the two squared-error terms in Equation 24.9; these may be considered as a *current-mode compensation* term and a *voltage error* term, respectively. The second term in Equation 24.9 forces convergence of $(v_C - V_C^*)$ to zero, or equivalently convergence of the state $v_C \to V_C^*$. The first term in Equation 24.9 forces the state i_L to converge to $I_L^* + K(V_C^* - v_C)$ which in turn converges to I_L^* as $v_C \to V_C^*$. The structure of the *current-mode compensation* term is used to improve transient response by increasing the input power when $v_C < V_C^*$ and decreasing input power when $v_C > V_C^*$. The parameter

$K$is selected to adjust the level of the compensation current for a given voltage error. In practice, the controller parameters K, C_I, and C_V are determined empirically to provide a desired transient response.

It is shown in Theorem 9 of [3] that a solution for the EOCP exists for a class of nonlinear systems that are linear in control inputs u_0, u_1, and have a PI that is convex in u_0, u_1. Since the system considered here is linear and does not have u_0, u_1, these conditions are satisfied trivially. Thus, a solution exists for the EOCP given by Equation 24.8.

24.3.2 Relationship of EOCP and SOCP

In [3], it is demonstrated that except in certain rare circumstances, solutions to the SOCP can be found by solving the EOCP for $\tilde{s} \in [0, 1]$. If the resulting solution is bang-bang, it solves the SOCP. If the resulting solution is not bang-bang, one projects the solution onto the set $\{0, 1\}$ using in our case a duty cycle interpretation. Specifically there are four main points regarding the relationship between the EOCP and SOCP solutions as given in [3]:

1. The EOCP always has a solution, whereas the SOCP is not guaranteed to have a solution.
2. The value of P_s for the EOCP lower bounds the value of P_s for the SOCP.
3. When the EOCP has a bang-bang solution, $\tilde{s} \in \{0, 1\}$, that minimizes P_s, it also solves the original SOCP.
4. For $\tilde{s} \notin \{0, 1\}$, since the EOCP always has a solution and since the switched trajectories are dense in the trajectories of the embedded system, a solution to the original switched system can be made to approximate the EOCP solution to any desired accuracy.

The EOCP is solved numerically by forming a discrete-time embedded model and discrete-time embedded PI, and then finding the piecewise constant embedded switch states $\tilde{s}_k, \tilde{s}_{k+1}, \ldots, \tilde{s}_{k+N-1}$ that minimize the discretized P_s. Consistent with MPC philosophy [4], the first control in the sequence $\tilde{s}_k$ is then projected onto an $s(t) \in \{0, 1\}$ using a duty cycle interpretation. The process is then repeated over the indexed user-defined window of optimization. This process is herein referred to as the HMPC optimization. Minimization of the discrete-time PI is accomplished numerically using a constrained optimization method. Herein, an Active-Set method is employed.

24.4 Design of HMPC

Using the steps set forth in this section, the EOCP is discretized and cast as a HMPC problem that is solved numerically. The problem is discretized with time step $T_s = 1/F_s$, where F_s is the switching frequency. The discrete-time points are thus $t_k = k \cdot T_s$, $x_k = x(t_k)$.

It is assumed that circuit parameters V_s, L, C, and R_{load}, and switching frequency F_s are known. To account for source and load variation, a parameter estimator may be paired with the control to update the model as V_s and R_{load} change; however, for the system considered here, V_s and R_{load} remain constant so that the focus remains on the MPC strategy. Parameter estimators are described in [5,6].

Step 1: Discrete-Time Embedded Model

The discrete-time embedded model

$$x_{k+1} = (1 - \tilde{s}_k) \cdot ([I + T_s A_0]x_k + T_s b_0) + \tilde{s}_k \cdot ([I + T_s A_1]x_k + T_s b_1) \tag{24.11}$$

is developed using a Forward Euler (FE) derivative approximation; the discrete-time embedded switch state $\tilde{s}_k \in [0, 1]$ is held constant over $t_k \leq t < t_{k+1}$. Although alternate discrete-time approximations may be used, the FE method is selected in order to simplify the numerical optimization and has proven satisfactory in experiments.

Step 2: Compensation for Computation Delay

To compensate for computation delay, the state, $\hat{x}_{k+1}$, at t_{k+1} is estimated from a state measurement, $x_{m,k}$, at t_k, and the previously computed switch control, $\tilde{s}_k$. The estimated $\hat{x}_{k+1}$ is used to initialize the finite horizon MPC window using the formula

$$\hat{x}_{k+1} = (1 - \tilde{s}_k) \cdot ([I + T_s A_0]x_{m,k} + T_s b_0) + \tilde{s}_k \cdot ([I + T_s A_1]x_{m,k} + T_s b_1). \tag{24.12}$$

Step 3: Discrete-Time PI

As mentioned earlier, we denote the equilibrium state as $x^* = \begin{bmatrix} I_L^* & V_C^* \end{bmatrix}^T$, and the state error by $\bar{x} = x - x^*$. Therefore, the integrands of the PI, Equation 24.8, may be expressed in the more customary quadratic form

$$F_0 = F_1 = \bar{x}^{\mathrm{T}} Q \bar{x}, \tag{24.13}$$

where

$$Q = \begin{bmatrix} C_I & C_I K \\ C_I K & C_I K^2 + C_V \end{bmatrix} \succ 0. \tag{24.14}$$

The finite-time horizon window is divided into N partitions: $[t_{k+1}, t_{k+2}] \ldots [t_{k+N}, t_{k+N+1}]$, and the discrete-time embedded PI is established from Equation 24.7 using the trapezoidal rule. For a 2-partition MPC window, the PI is given by

$$P_s = \frac{T_s}{2}\left(\bar{x}_{k+1}^T Q \bar{x}_{k+1} + 2\bar{x}_{k+2}^T Q \bar{x}_{k+2} + \bar{x}_{k+3}^T Q \bar{x}_{k+3}\right). \tag{24.15}$$

Step 4: Inequality Constraints

Minimization of the discrete-time PI Equation 24.15 is subject to inequality constraints on the controls and state. To maintain continuous conduction, $i_L(t) > 0$, the discrete-time embedded inductor currents are subject to a lower bound: $i_L(k+1) \geq i_{\min}, i_L(k+2) \geq i_{\min}$. Using the discrete-time model, these lower bounds are represented as additional constraints on $\tilde{s}_{k+1}, \tilde{s}_{k+2}$. The inequality constraints are summarized as follows:

- Admissible controls

$$0 \leq \tilde{s}_{k+1}, \;\; \tilde{s}_{k+2} \leq 1. \tag{24.16}$$

- Controls that maintain continuous conduction

$$\tilde{s}_{k+1} \geq \frac{L}{T_s}\frac{(i_{\min} - \hat{i}_L(k+1))}{\hat{v}_C(k+1)} + \frac{(\hat{v}_C(k+1) - V_s)}{\hat{v}_C(k+1)}, \tag{24.17}$$

$$\tilde{s}_{k+1} + \tilde{s}_{k+2} \geq \frac{L}{T_s}\frac{(i_{\min} - \hat{i}_L(k+1))}{\hat{v}_C(k+1)} + 2\frac{(\hat{v}_C(k+1) - V_s)}{\hat{v}_C(k+1)}. \tag{24.18}$$

Inequality (Equation 24.17) is obtained by direct algebraic manipulation of Equation 24.11. The inequality (Equation 24.18) is also obtained using Equation 24.11 with the assumption that $v_C(k+1)/v_C(k+2) \approx 1$. The lower bound $i_{\min}$ requires knowledge of the switching function that will be used. As will be described in Step 5, a center-aligned pulse width modulation (PWM) will be implemented which leads to the lower bound

$$i_{\min} = \frac{T_s V_s}{2L}\frac{(\hat{v}_C(k+1) - V_s)}{\hat{v}_C(k+1)}. \tag{24.19}$$

The set of feasible controls is given by the intersection of constraint sets (Equations 24.16 through 24.18) and is denoted by S_{CCM}, which in our case is nonempty, compact, and convex.

At each interval, $\hat{i}_L(k+1)$ and $\hat{v}_C(k+1)$ are determined and the inequality constraints given by Equations 24.17 and 24.18 are redefined. However, since each of the constraints (Equations 24.16 through 24.18) is a convex set, the intersection of these is always a convex set [7].

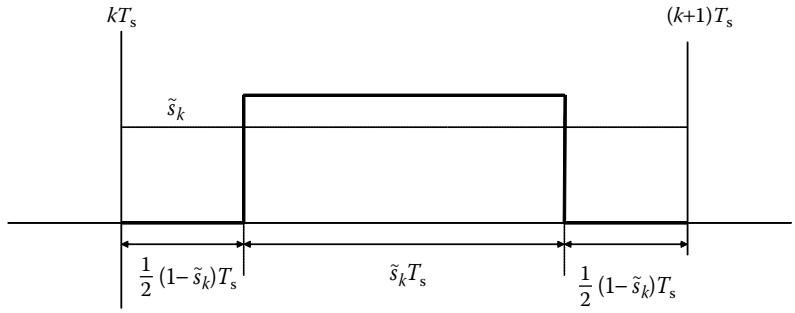

FIGURE 24.2 Projection algorithm for determining $s(t)$ from $\tilde{s}_k$ on the interval $kT_s \leq t < (k+1)T_s$. This PWM waveform is commonly referred to as "Center-aligned" PWM.

Step 5: Numerical Solution

The HMPC problem applied over the two partitions is expressed as the nonlinear program

$$\underset{\tilde{s}_{k+1},\tilde{s}_{k+2}\in S_{CCM}}{\text{minimize}} \quad P_s \tag{24.20}$$

subject to Equation 24.11. The set S_{CCM} is nonempty and compact, and P_s is continuous in $\tilde{s}_{k+1}, \tilde{s}_{k+2}$; thus, a solution exists for Equation 24.20 [8].

For the results presented in this chapter, the minimization over $\tilde{s}_{k+1}, \tilde{s}_{k+2}$ is accomplished using an Active Set algorithm that relies on Newton steps. The algorithm is explained in detail later in this section. Problem (Equation 24.20) is solved in less than one switch period, allowing for real-time implementation of the control.

Step 6: Projection Algorithm

Numerical solution of Equation 24.20 determines optimal values for $\tilde{s}_{k+1}, \tilde{s}_{k+2} \in [0, 1]$; however, the physical switch requires an on/off signal $s(t) \in \{0, 1\}$. Therefore, as a final step, $\tilde{s}_{k+1}$ is projected onto the actual switch set $s(t) \in \{0, 1\}$ using a duty cycle interpretation. The projection algorithm is illustrated in Figure 24.2. The switch signal $s(t)$ is then used to drive the switch in Figure 24.1. The embedded solution for the second partition $\tilde{s}_{k+2}$ is used to initialize the optimization for the next MPC window, which is referred to as *shift initialization* [9].

A block-diagram summarizing the HMPC control is provided in Figure 24.3.

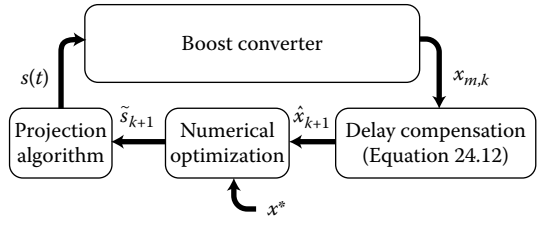

FIGURE 24.3 Controller Block Diagram, where $x_{m,k}$ is a state measurement taken at t_k, $\hat{x}_{k+1}$ is the predicted state at t_{k+1}, $x^* = [I_L^* \;\; V_C^*]^{\text{T}}$ is the desired state, $\tilde{s}_{k+1}$ is generated by minimizing the PI subject to constraints, and $s(t)$ is the on/off signal applied to the converter switch.

24.5 Numerical Optimization Algorithm

Optimization problems of the form (Equation 24.20) may be solved using any one of a number of methods, including Gradient Projection, Interior Point, or Active Set methods. We have selected an Active Set method because it has been shown to be well suited for low-dimensional problems with few constraints. The Active Set method employed here is similar to those found in [10,11]. To handle the model constraint, Equation 24.11 is substituted into Equation 24.15. In doing so, it is readily shown that P_s depends only on $\tilde{s}_{k+1}$, $\tilde{s}_{k+2}$, and $\hat{x}_{k+1}$.

The inequality constraints given by Equations 24.16 through 24.18 are written in the form $g_i(\tilde{s}_{k+1}, \tilde{s}_{k+2}) \leq 0$, where $i \in \{1, 2, \ldots, 6\}$. Specifically, Equation 24.16 gives:

$$g_1\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = -\tilde{s}_{k+1}, \tag{24.21a}$$

$$g_2\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = \tilde{s}_{k+1} - 1, \tag{24.21b}$$

$$g_3\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = -\tilde{s}_{k+2}, \tag{24.21c}$$

$$g_4\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = \tilde{s}_{k+2} - 1, \tag{24.21d}$$

and Equations 24.17 and 24.18 give

$$g_5\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = -\tilde{s}_{k+1} + \frac{L}{T_s}\frac{\left(i_{\min} - \hat{i}_L(k+1)\right)}{\hat{v}_C(k+1)} + \frac{\left(\hat{v}_C(k+1) - V_s\right)}{\hat{v}_C(k+1)}, \tag{24.22a}$$

$$g_6\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = -\tilde{s}_{k+1} - \tilde{s}_{k+2} + \frac{L}{T_s}\frac{\left(i_{\min} - \hat{i}_L(k+1)\right)}{\hat{v}_C(k+1)} + 2\frac{\left(\hat{v}_C(k+1) - V_s\right)}{\hat{v}_C(k+1)}. \tag{24.22b}$$

An inequality constraint is considered to be *active* when $g_i\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = 0$ and *inactive* when $g_i\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) < 0$. The indices of *active* constraints are members of the *working* set ϑ, that is, $g_i\left(\tilde{s}_{k+1}, \tilde{s}_{k+2}\right) = 0$ for $i \in \vartheta$. The set of active constraints must be linearly independent; therefore, since the optimization is performed in R^2, at most two constraints are active.

The optimization is summarized in the following three steps; Steps 1 and 2 include computations to initialize the optimization, and Step 3 describes the Active Set algorithm.

24.5.1 Beginning of Optimization

1. *Step 1:* Using the initial value $\hat{x}_{k+1}$, inequalities (Equation 24.22) are computed (which updates S_{CCM}).
2. *Step 2:* An initial feasible starting point is determined $(\tilde{s}_{k+1})^0, (\tilde{s}_{k+2})^0 \in S_{CCM}$. For the results presented here, an initial value is generated using shift initialization and then checked for feasibility. If $(\tilde{s}_{k+1})^0, (\tilde{s}_{k+2})^0 \notin S_{CCM}$, an orthogonal projection step is implemented to place the starting value back onto the interior of S_{CCM}.
3. *Step 3:* The Active Set algorithm is employed to minimize P_s. Beginning with iteration index $n = 1$ and all inequality constraints *inactive* (i.e., $\vartheta = \emptyset$), the algorithm is given as follows:
   Iterate
   a. Given $\left(\tilde{s}_{k+1}\right)^0, \left(\tilde{s}_{k+2}\right)^0$, and $\hat{x}_{k+1}$, the partial derivatives $\partial P_s/\partial \tilde{s}_{k+1}$, $\partial P_s/\partial \tilde{s}_{k+2}$, and $\partial^2 P_s/\partial \tilde{s}_{k+i}\partial \tilde{s}_{k+j}$ for $i, j \in \{1, 2\}$ are computed.
   b. Perform one Newton step to estimate the solution of the equality constrained problem:

$$\underset{(\tilde{s}_{k+1})^n, (\tilde{s}_{k+2})^n}{\text{minimize}} \; P_s \tag{24.23}$$

   subject to $g_i((\tilde{s}_{k+1})^n, (\tilde{s}_{k+2})^n) = 0$ for $i \in \vartheta$.
   In the first iteration ($n = 1$), this is always unconstrained (i.e., $\vartheta = \emptyset$).

c. Estimate the Lagrange multipliers λ_i for $i \in \vartheta$, and remove the constraint with the most negative Lagrange multiplier from ϑ.
d. Check feasibility of the solution generated in step (b). If $(\tilde{s}_{k+1})^n, (\tilde{s}_{k+2})^n \notin S_{\text{CCM}}$, project the solution back onto the boundary of S_{CCM} and activate the constraint that was violated in the update. Herein, this is done using backtracking.
e. If $n = 5$, return optimal solution $(\tilde{s}_{k+1})^*, (\tilde{s}_{k+2})^*$.

Else, increment iteration index: $n = n+1$, and go back to step a.

24.5.2 End Optimization

The above steps outline the Active Set algorithm used to solve the constrained HMPC optimization. The steps 3(a) through 3(d) are discussed in greater detail in the following.

24.5.2.1 Newton Step

The Newton step will depend on the elements of ϑ. For example, if $\vartheta = \emptyset$, the Newton step is implemented using

$$\begin{bmatrix} \tilde{s}_{k+1} \\ \tilde{s}_{k+2} \end{bmatrix}^n = \begin{bmatrix} \tilde{s}_{k+1} \\ \tilde{s}_{k+2} \end{bmatrix}^{n-1} - \begin{bmatrix} \dfrac{\partial^2 P_s}{\partial \tilde{s}_{k+1}^2} & \dfrac{\partial^2 P_s}{\partial \tilde{s}_{k+1} \partial \tilde{s}_{k+2}} \\ \dfrac{\partial^2 P_s}{\partial \tilde{s}_{k+1} \partial \tilde{s}_{k+2}} & \dfrac{\partial^2 P_s}{\partial \tilde{s}_{k+2}^2} \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\partial P_s}{\partial \tilde{s}_{k+1}} \\ \dfrac{\partial P_s}{\partial \tilde{s}_{k+2}} \end{bmatrix}. \tag{24.24}$$

If ϑ contains a single constraint, the optimization reduces to a line search. For example, if $\vartheta = \{2\}$, which is the constraint $\tilde{s}_{k+1} = 1$, then

$$(\tilde{s}_{k+1})^n = (\tilde{s}_{k+1})^{n-1}, \tag{24.25a}$$

$$(\tilde{s}_{k+2})^n = (\tilde{s}_{k+2})^{n-1} - \left(\frac{\partial^2 P_s}{\partial \tilde{s}_{k+2}^2} \right)^{-1} \left(\frac{\partial P_s}{\partial \tilde{s}_{k+2}} \right). \tag{24.25b}$$

If ϑ contains two constraints, then $(\tilde{s}_{k+1})^n$ and $(\tilde{s}_{k+2})^n$ are unchanged from their previous values. For the results presented here, we have selected a bound of $n = 5$ iterations. If the working set does not change from one iteration to the next, the solution is further refined by subsequent Newton steps.

Although the control problem is a constrained optimization, it is worth noting that the converter is unlikely to operate along a constraint boundary in steady state. In this case, the inequality constraints will not be active. If the inequality constraints are inactive for all iterations, the algorithm reduces to Newton–Raphson. In steady state, it is assumed the starting value is already "close" to the optimal value; so Newton–Raphson has second-order convergence; therefore, the error between the solution and the optimal value is considered insignificant after two or three Newton steps.

24.5.2.2 Lagrange Multipliers

In step 3(c), the Lagrange multipliers for the active constraints are estimated by minimizing $\left\| \sum_{i \in \vartheta} \lambda_i \nabla g_i + \nabla P_s \right\|$ over $\lambda_i, i \in \vartheta$ where ∇P_s is the gradient of P_s with respect to $\tilde{s}_{k+1}, \tilde{s}_{k+2}$. The constraint with the most negative Lagrange multiplier is removed from the working set.

$$\text{If } \exists \lambda_i < 0, \quad \text{then } \vartheta = \vartheta / i, \text{ where } i = \arg\min(\lambda_i).$$

24.5.2.3 Check Feasibility/Backtracking

In step 3(d), the feasibility of the values $(\tilde{s}_{k+1})^n$ and $(\tilde{s}_{k+2})^n$ is checked by testing the inactive constraints. If an inactive constraint is violated, the solution must be remapped to meet the new constraint.

To do so, the following quantities are computed:

$$\Delta\tilde{s}_{k+1} = (\tilde{s}_{k+1})^n - (\tilde{s}_{k+1})^{n-1}, \tag{24.26}$$

$$\Delta\tilde{s}_{k+2} = (\tilde{s}_{k+2})^n - (\tilde{s}_{k+2})^{n-1}, \tag{24.27}$$

and a scaling factor is calculated:

$$\alpha' = \max\left\{\alpha \mid g_i\left(\tilde{s}_{k+1} + \alpha\Delta\tilde{s}_{k+1}, \tilde{s}_{k+2} + \alpha\Delta\tilde{s}_{k+2}\right) \leq 0 \text{ for } i \notin \vartheta\right\}.$$

Subsequently, the solution is updated using

$$\begin{bmatrix}\tilde{s}_{k+1}\\ \tilde{s}_{k+2}\end{bmatrix}^n = \begin{bmatrix}\tilde{s}_{k+1}\\ \tilde{s}_{k+2}\end{bmatrix}^{n-1} + \alpha' \begin{bmatrix}\Delta\tilde{s}_{k+1}\\ \Delta\tilde{s}_{k+2}\end{bmatrix}. \tag{24.28}$$

In addition, the set ϑ must be updated to include a new active constraint,

$$\vartheta \leftarrow \vartheta \cup i \quad \text{where} \quad g_i\left(\tilde{s}_{k+1} + \alpha'\Delta\tilde{s}_{k+1}, \tilde{s}_{k+2} + \alpha'\Delta\tilde{s}_{k+2}\right) = 0.$$

It is noted that the above-described optimization algorithm is flexible and may be altered to meet user requirements. For example, this algorithm has been found to work well in practice using a Gauss–Newton step in place of a Newton step. In addition, convergence of the algorithm may be improved for jump changes in x^* by selecting an initial value $(\tilde{s}_{k+1})^0, (\tilde{s}_{k+2})^0$ using a method other than *shift initialization*.

24.5.2.4 Comments on the Numerical Solution and Selection of Parameters K, C_I, and C_V

In order to design an effective control, the user must select a PI based on their insight into the physical system; however, it is also beneficial to select the PI based on its numerical tractability. Nonlinear programs may have multiple local minima, making it difficult to locate the global minimum efficiently. However, the set S_{CCM} is always convex; if P_s is selected to be convex in $\tilde{s}_{k+1}, \tilde{s}_{k+2} \in S_{\text{CCM}}$, then Equation 24.20 is a convex program, and any local minimum is also the unique global minimum [7]. For the control considered here, the PI is selected to be a quadratic function of the state error with matrix $Q \succ 0$. With this formulation, the first and second derivatives are easily computed and an appropriate selection of Q, reduces problem (Equation 24.20) to a convex program for a given bound on $\|\tilde{x}_{k+1}\|$. In the numerical examples and hardware experiments that follow, the control parameters K, C_I, and C_V were selected to command the desired system response and also to permit convex optimization methods to be used. The optimization algorithm is shown to perform well for step changes in commanded output voltage of values up to 20 V.

24.5.2.5 Numerical Examples

Consider a boost converter with the following circuit parameter values: $V_s = 230$ V, $L = 1.0$ mH, $C = 100$ nF, $R_{\text{load}} = 100\ \Omega$, switching frequency $F_s = 15.68$ kHz, and controller parameter values $C_I = 1.0$, $K = 0.8$, $C_V = 0.5$. Two examples are considered, with different initial conditions and commanded output voltages consistent with transients demonstrated in hardware in the next section.

The first numerical example is for an initial state $\hat{x}_{k+1} = [5.0\,A\ \ 330\,V]^{\text{T}}$, a desired state $x^* = [5.3\,A\ \ 350\,V]^{\text{T}}$ and starting values $(\tilde{s}_{k+1})^0 = (\tilde{s}_{k+2})^0 = 0.5$. The progression of the solution iterates are illustrated in the $\tilde{s}_{k+1}, \tilde{s}_{k+2}$ plane in Figure 24.4. Also shown in Figure 24.4 are the level sets of the PI and the inequality constraints, indicated by a box for Equation 24.16 and two lines for Equations 24.17 and 24.18. Figure 24.5 shows the value of P_s at each iteration. The algorithm is applied for up to five iterations in this example. Although the initial values were not close to the solution, Figure 24.5 illustrates that the optimal control $(\tilde{s}_{k+1})^* = 1.00$, $(\tilde{s}_{k+2})^* = 0.00$ is determined in two iterations. It is noted that this

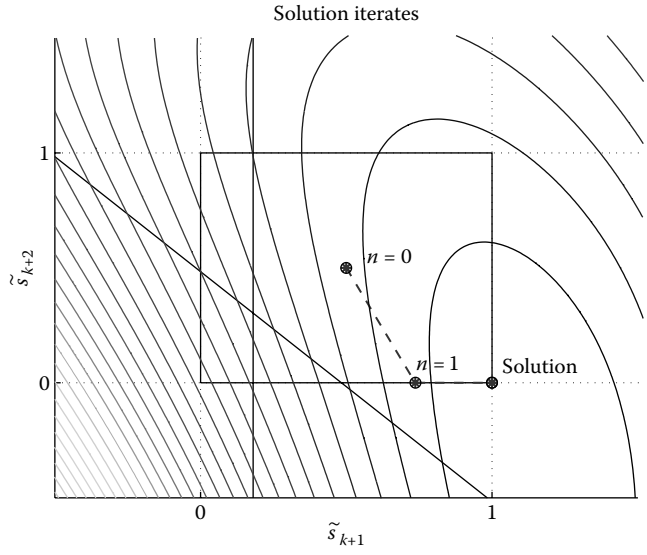

FIGURE 24.4 Illustration of Active Set solution showing solution iterates for an initial state $\hat{x}_{k+1} = [5.0\,A\ \ 330\,V]^{\mathrm{T}}$ and a commanded state $x^* = [5.3\,A\ \ 350\,V]^{\mathrm{T}}$. The algorithm yields a bang-bang solution: $(\tilde{s}_{k+1})^* = 1.00$, $(\tilde{s}_{k+2})^* = 0.00$, where $\vartheta = \{2, 3\}$.

constitutes a bang-bang solution since $\tilde{s} \in \{0, 1\}$ for each of the switch periods; thus, the EOCP solution also solves the SOCP.

The second numerical example is for an initial state $\hat{x}_{k+1} = [5.3\,A\ \ 350\,V]^{\mathrm{T}}$, a desired state $x^* = [5.0\,A\ \ 340\,V]^{\mathrm{T}}$, and for starting values $(\tilde{s}_{k+1})^0 = (\tilde{s}_{k+2})^0 = 0.6$. The algorithm yields the solution $(\tilde{s}_{k+1})^* = 0.218$, $(\tilde{s}_{k+2})^* = 0.343$. Evolution of the solution iterates and the value of P_s are provided in Figures 24.6 and 24.7.

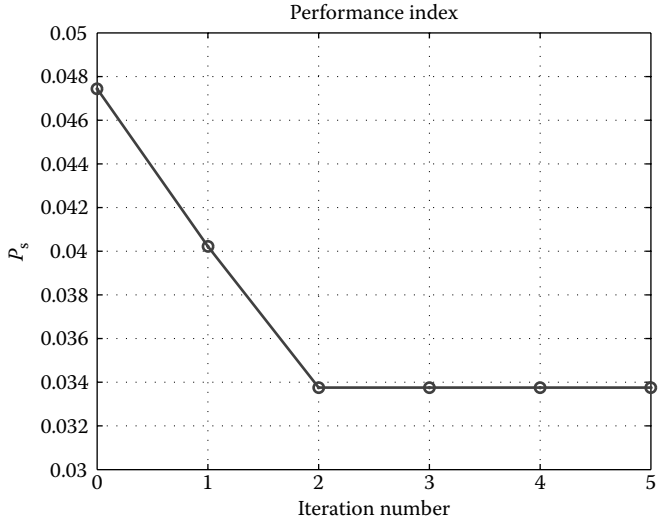

FIGURE 24.5 Values of P_s at each iteration for the first numerical example. The algorithm includes five iterations, but the solution is effectively determined in just two iterations.

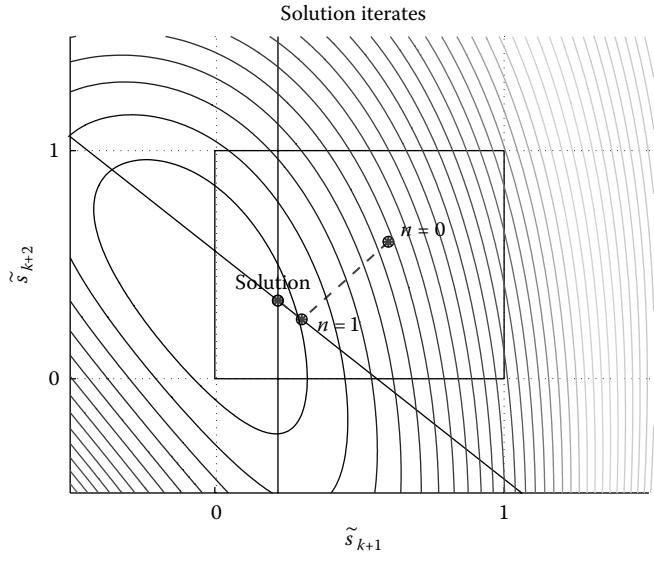

FIGURE 24.6 Illustration of Active Set solution showing solution iterates for an initial state $\hat{x}_{k+1} = [5.3\,A \;\; 350\,V]^{\mathrm{T}}$ and a commanded state $x^* = [5.0\,A \;\; 340\,V]^{\mathrm{T}}$. The algorithm yields the solution: $(\tilde{s}_{k+1})^* = 0.218$, $(\tilde{s}_{k+2})^* = 0.343$, where $\vartheta = \{5, 6\}$.

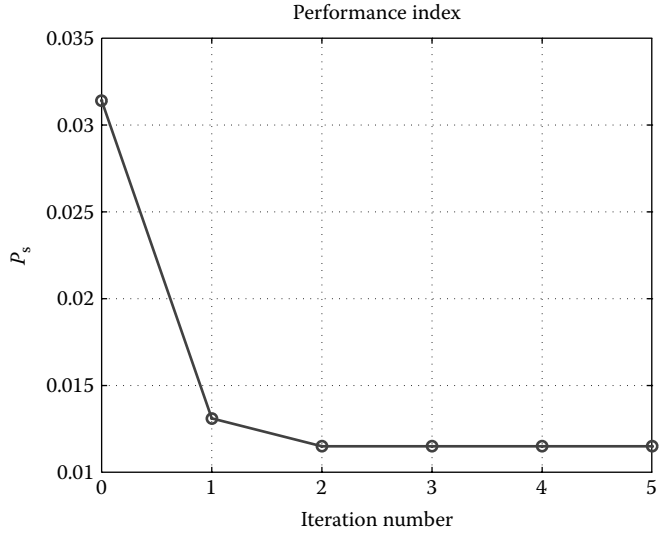

FIGURE 24.7 Values of P_s at each iteration for the second numerical example. The algorithm includes five iterations, but the solution is effectively determined in just two iterations.

24.6 Hardware Implementation

The HMPC is applied to a boost converter with nominal circuit parameter values: $V_s = 230$ V, $L = 1.0$ mH, $C = 100$ nF, and $R_{\mathrm{load}} = 100\ \Omega$, the switching frequency is $F_s = 15.68$ kHz, and PI parameter values are given: $C_V = 0.5$, $C_I = 1.0$, and $K = 0.8$. These parameters are identical to those given in the numerical examples above.

The source voltage V_s was provided to the boost converter by a Sorensen 300–33T dc power supply. All computations for implementing the HMPC were performed by a TMS320C6711 DSP onboard the Toro PCI card by Innovative Integration. The voltage is measured by a Tektronix P5200 differential voltage probe, and the current is measured using a Tektronix A6303 current sensor with AM503B amplifier. The sensor outputs are connected directly to Analog-to-Digital channels on the Toro PCI card.

The DSP samples voltage and current measurements at time t_k, predicts the state $\hat{x}_{k+1}$, solves for $\tilde{s}_{k+1}$ using the Active Set algorithm, and produces an analog signal between 0 and 5 V that corresponds linearly to values for the embedded switch state $\tilde{s}(t) \in [0, 1]$. The embedded solution $\tilde{s}(t)$ is provided to a center-aligned PWM peripheral that implements the projection algorithm.

A single experiment is done with two changes to the commanded output voltage. Beginning with the converter in steady state with $V_C^* = 330\,\text{V}$, the commanded output voltage was then stepped to $V_C^* = 350\,\text{V}$ and subsequently to $V_C^* = 340\,\text{V}$. The output voltage, inductor current, and analog signal $\tilde{s}(t)$ are shown in Figure 24.8. The dynamic response is strong; the rise time of the output voltage is less than 500 μs without overshoot or oscillation. The voltage drop from 350 to 340 V is slightly slower due to the time constant $R_{\text{load}}C$ and due to the fact that the controller maintains continuous conduction.

A close-up of the measured response during the first transient is provided in Figure 24.9. Figure 24.9 clearly indicates that, as V_C^* is stepped from 330 to 350 V, the embedded switch is $\tilde{s} = 1$ for one period

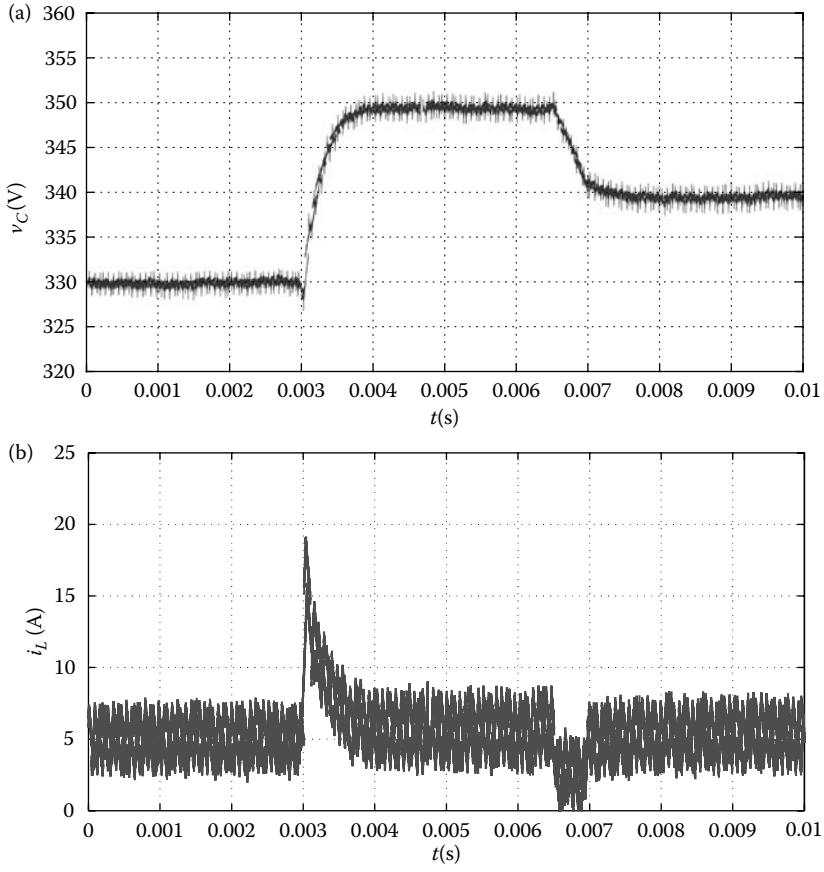

FIGURE 24.8 Hardware results for step change in desired voltage: $V_C^* = 330\,\text{V}$ to $V_C^* = 350\,\text{V}$ to $V_C^* = 340\,\text{V}$, including the measured (a) capacitor voltage; (b) inductor current; and (c) embedded switch state waveforms.

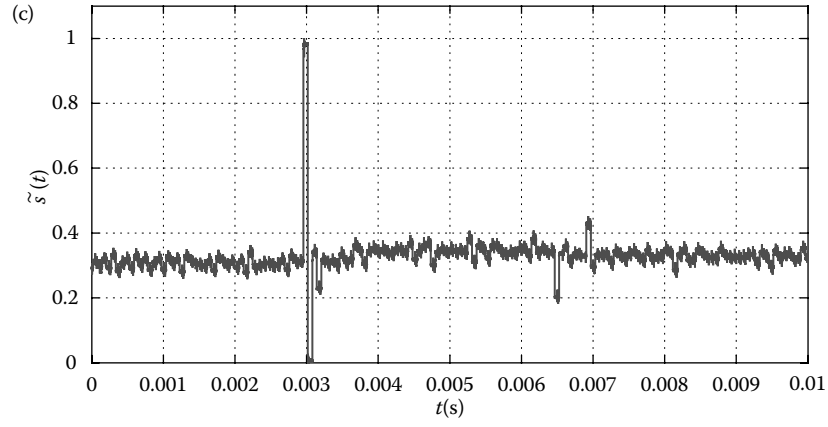

FIGURE 24.8 Continued.

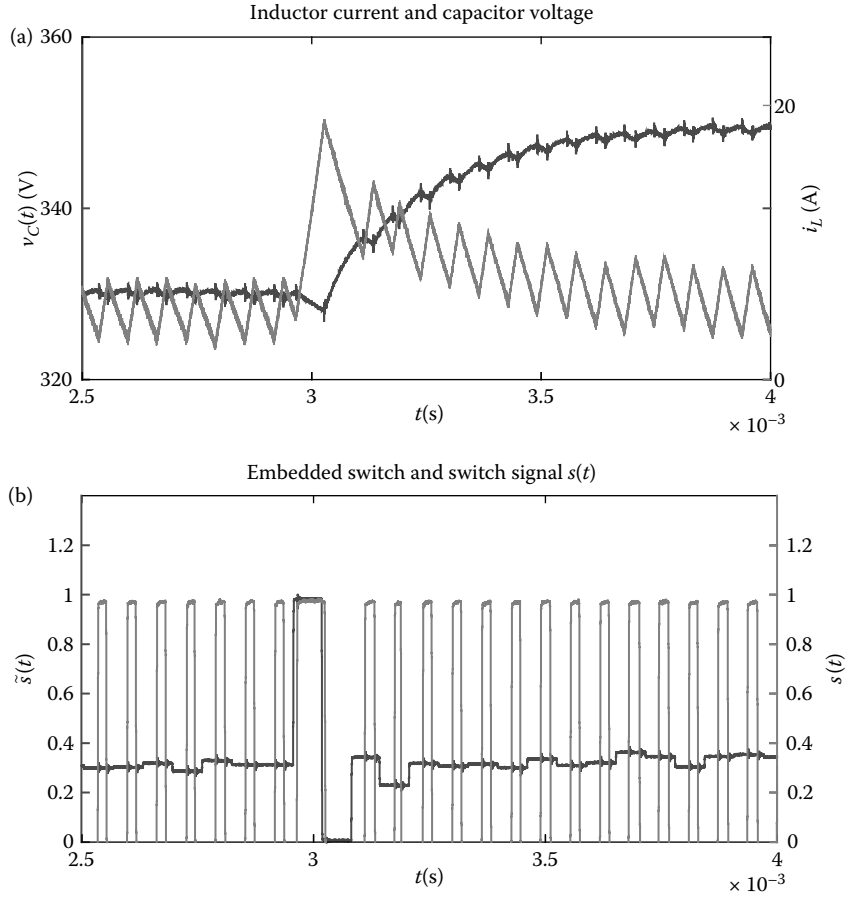

FIGURE 24.9 Close-up of (a) inductor current and capacitor voltage and (b) embedded switch state and gate drive signal for step change in desired voltage: $V_C^* = 330$ V to $V_C^* = 350$ V.

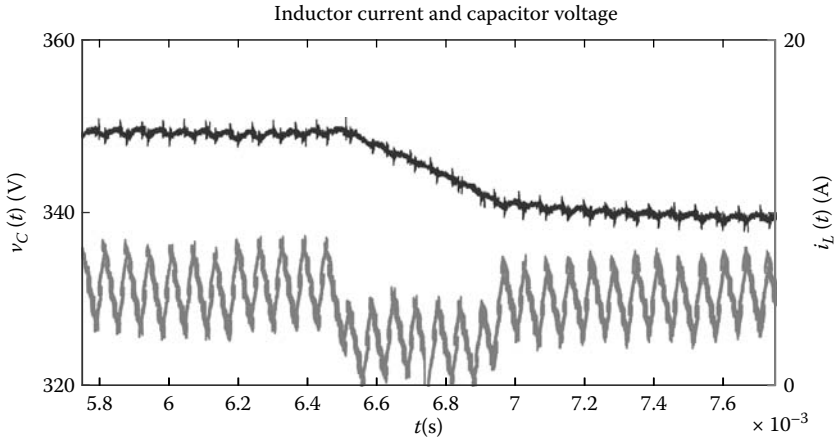

FIGURE 24.10 Close-up of inductor current and capacitor voltage for step change in desired voltage: $V_C^* = 350$ V to $V_C^* = 340$ V. The HMPC maintains the constraint $i_L(t) > 0$.

and then $\tilde{s} = 0$ for the next period, which is consistent with the bang-bang solution indicated in the first numerical example above.

A close-up of the converter state is provided in Figure 24.10 for the second transient. Figure 24.10 illustrates that the converter is maintained in continuous conduction ($i_L(t) > 0$). It is also noted that, as V_C^* is stepped from 350 to 340 V, the embedded switch is $\tilde{s} = 0.203$ for one period and then $\tilde{s} = 0.338$ for another period, which is similar to the solution indicated in the second numerical example above (Figure 24.6).

24.6.1 Practical Considerations

The hardware results presented in this section do not consider the case when the source voltage and load resistance values change. In a practical dc–dc converter, the control must regulate the output voltage in the presence of source and load variation. This can be accomplished using a parameter estimator that updates estimates of source voltage and load resistance based on state measurements and updates the dynamic model. Combined HMPC and parameter estimator schemes are described in [5,6].

24.7 Summary

In this chapter, an HMPC strategy for switching converters has been realized that may be implemented in real time. Within HMPC, an embedded form of the hybrid state model is created and is used to predict converter dynamics over a finite horizon window. Switching is then determined such that predicted state trajectories minimize a user-defined PI. Methods for designing the controller are provided and results of hardware implementation are given to demonstrate controller efficacy.

In particular, a step-by-step guide is given for defining the discrete-time optimization, an Active Set algorithm is provided for solving the optimization, and numerical examples illustrate the operation of the algorithm.

References

1. Krein, P.T.; *Elements of Power Electronics*, Oxford University Press, New York and Oxford, 1998.

2. Krein, P.T.; Bentsman, J.; Bass, R.M.; Lesieutre, B.L.; On the use of averaging for the analysis of power electronic systems, *IEEE Transactions on Power Electronics*, 5(2), 182–190, 1990.
3. Bengea, SC.; DeCarlo, R.A.; Optimal control of switching systems, *Automatica*, 41(1), 11–27, 2005.
4. Camacho, E.; Bordons, C.; *Model Predictive Control*. Springer-Verlag, Berlin, 2004.
5. Oettmeier, F.M.; Neely, J.; Pekarek, S.; DeCarlo, R.; Uthaichana, K.; MPC of switching in a boost converter using a hybrid state model with a sliding mode observer, *IEEE Transactions on Industrial Electronics*, 56(9), 3453–3466, 2009.
6. Neely, J.; Pekarek, S.; DeCarlo, R.; Hybrid optimal-based control of a boost converter, *Twenty Fourth Annual IEEE Applied Power Electronics Conference and Exposition, 2009. APEC 2009*, Washington, DC, pp. 1129–1137, February 15–19, 2009.
7. Boyd, S.; *Convex Optimization*; Cambridge University Press, Cambridge, UK, 2004.
8. Wade, W.R.; *An Introduction to Analysis*; 3rd edition, Prentice-Hall, Englewood Cliffs, NJ, pp. 274–275, 2004.
9. Diehl, M.; Ferreau, H.J.; Haverbeke, N.; Efficient numerical methods for nonlinear MPC and moving horizon estimation; *International Workshop on Assessment and Future Directions of NMPC*; Pavia, Italy, September 5–9, 2008.
10. Nocedal, J.; Wright, J.W.; *Numerical Optimization*, 2nd edition, Springer-Verlag, New York, NY, 2006.
11. Allgöwer, F.; Zheng, A.; *Nonlinear Model Predictive Control*, Birkhäuser-Verlag, Berlin, pp. 335–346, 2000.

VI

Networks

25

The SNR Approach to Networked Control

Eduardo I. Silva
Federico Santa María Technical University

Juan C. Agüero
The University of Newcastle

Graham C. Goodwin
The University of Newcastle

Katrina Lau
The University of Newcastle

Meng Wang
The University of Newcastle

25.1 Introduction

Control has undergone several distinct phases of development. There are various ways that one can classify the different phases. An overriding issue has been the way that actuators, controllers, and sensors are interconnected. There has been a progression from direct mechanical interconnection to pneumatic and hydraulic interconnection to electrical connection and, finally networked and/or wireless connection. The last phase has opened up entirely new challenges for the control system design community.

Standard control theory [1] assumes perfect communications between plant and controllers. The advances in communications technology have, on the other hand, motivated the use of general purpose communication networks in control [2]. Figure 25.1 shows a Networked Control System (NCS), where the input commands and the measurements used by the controller are transmitted through a network.

One of the most important observations about networked control is that there exist additional degrees of freedom in the design relative to the classical (nonnetworked) situation. In particular, one needs

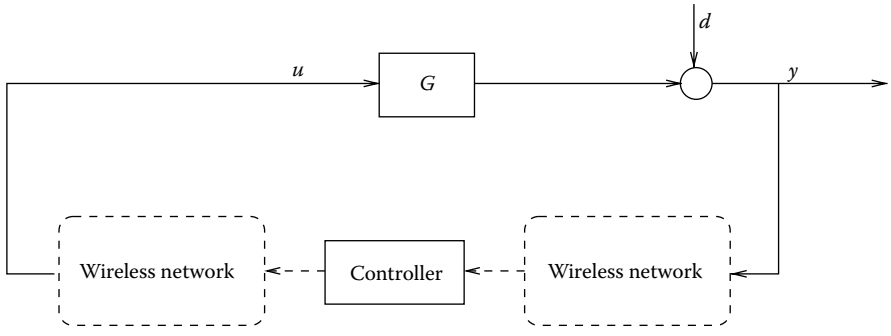

FIGURE 25.1 Networked control system.

to prepare signals prior to transmission over the network (via some form of coding) and it is then necessary to reinterpret the signals upon receipt (via some form of decoding). This new control architecture increases applicability and reduces cost when compared to hard-wired solutions. However, there are also drawbacks to the use of networked control architectures. For example, typical communication links are subject to data-rate limits [3], are prone to data-loss [4], and may experience random delays [5,6]. Dealing with these issues goes beyond standard control theory and a new integrated approach, lying at the interface between communication and control theory, has emerged inside the control community. A unified viewpoint that addresses all aspects of NCSs is, as yet, unavailable. However, many interesting results have been obtained. Good survey papers are contained in the special issue [7].

The aim of this chapter is to give a brief introduction to one particular viewpoint, namely the signal-to-noise ratio (SNR) approach to the design of NCSs. This approach is relatively simple. It is, basically, a method where the channel is replaced by an additive noise source where the associated variance appears as a degree of freedom in the design. It is thus readily understandable by practicing engineers. Notwithstanding the simplicity of the approach, it yields important insights into the design and performance of NCSs. An advantage of the SNR approach is that it leads to simple methods for designing networked control architectures based on linear time invariant filters and linear control design methodologies. It is thus immediately useful in practical situations (e.g., WCDMA power control, Section 25.2).

The emerging literature on networked control contains many deep insights, for example, the minimal average data-rate across a channel necessary to stabilize an unstable system [3]. These results typically depend on complex design of appropriate controllers and coder/decoder pairs. Interestingly, some of these results can be *formally* rederived by using simple additive noise models. Indeed, it will be shown below that, under suitable assumptions on the control architectures and encoding/decoding policies, it is possible to understand the impact of either average data-rate limits, or data-dropouts, in the context of SNR constraints in an additive white-noise (AWN) channel [8–10]. These results allow one to utilize SNR-related results (e.g., [11,12]) to draw conclusions that are valid in a broader context.

25.2 Motivational Case Study: WCDMA Power Control

Mobile telecommunications is a rapidly evolving area that has had a huge impact on modern society. These systems depend upon sophisticated control systems for their successful operation. In this section we use these systems as examples to illustrate emerging ideas in NCSs. The algorithms in everyday use include real-world examples of constrained multivariable, nonlinear stochastic NCSs having quantized data and experiencing random delays and packet loss.

We next give an introduction to the concepts, nomenclature and phenomenological aspects of wideband code-division multiple access (WCDMA) communication systems.

Third-generation (3G) cellular communication systems have been globally deployed to meet the growing demand for higher data-rate communication and multimedia services. Among 3G technical standards, WCDMA technology [13] has become the most widely adopted air interface. Transmission power control, particularly in the uplink (connection from the mobile to the base station (BS)), is vital for successful operation of WCDMA systems.

In a WCDMA cell (the area of coverage of particular BS) all of the pieces of user equipment (UE) operate within the same frequency band and their signals are separated, at the BS, by the use of unique spreading codes [14]. Consider the situation where one UE, UE1, is very close to the BS while another, UE2, is located at the cell edge. The path loss (channel gain) difference between the two UEs can be up to 70 dB [13]. Unless some control is exercised UE1 would block the communication of UE2 and also many other UEs to the BS. In addition, the signal path between the UE and the BS is not unique and is constantly changing due to UE movement across the cell. This is the result of different reflections and scattering of the transmitted signal, which cause time-varying time delays and frequency selective (or nonselective) channels. At the receiver the signals arriving from different paths add constructively (or destructively) to compose the final received signal. This gives rise to *fading*.

To handle these, so called, *near–far* and *fading* problems, a power control scheme is required to ensure that the received signal power from all UEs is approximately constant at the BS. Specifically, to maximize the overall system capacity, the transmission power of each UE is adjusted such that each received signal-to-interference ratio (SIR) is equal to the lowest allowable level consistent with a given target data rate [13].

A simplified block diagram of a typical inner loop for WCDMA uplink power control is shown in Figure 25.2. This setting will be used as an example of NCS. Note that in the telecommunications literature, it is common to express quantities on a logarithmic scale (dB). Thus, whenever we refer to the power control problem in WCDMA we use a log-scale. We also use z as the forward shift operator. We denote the transmitted power of the UE (also known as mobile station (MS)) by p, the channel gain by g and the interference from other users by I. Thus, the received SIR (on a logarithmic scale) is given by

$$y = p + g - I. \tag{25.1}$$

A target SIR r is provided by an outer control loop operating at a much lower rate [13]. The tracking error

$$e \triangleq r - y \tag{25.2}$$

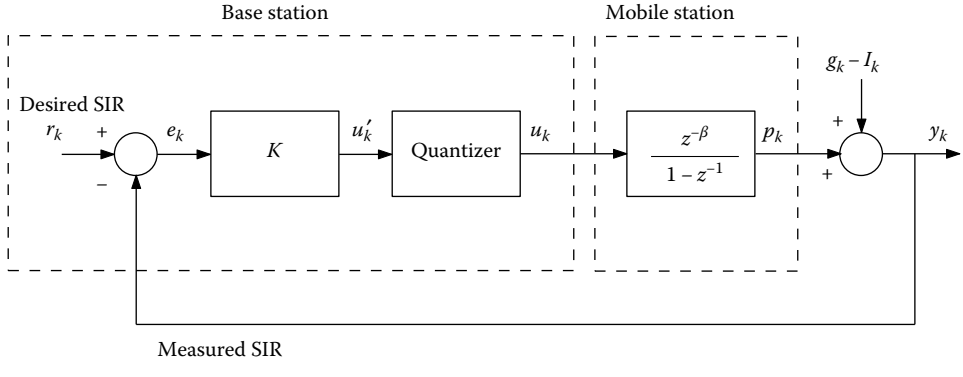

FIGURE 25.2 Simplified block diagram of the WCDMA inner power control loop.

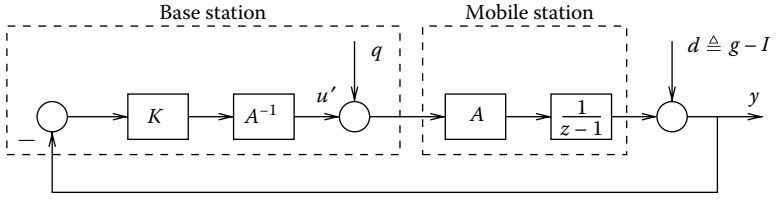

FIGURE 25.3 Use of additional degrees of freedom in the WCDMA inner power control loop.

is then fed to a controller K to calculate the power control increment u. Note that typically one bit is sent to the MS through the downlink control channel for power adjustment. This is the motivation for including the block denoted "Quantizer" in Figure 25.2. In Figure 25.2, β denotes a loop delay. The sampling period is here selected as one WCDMA slot time, that is, 667 μs [15].

We see that the control loop of Figure 25.2 involves quantization. Also, it is possible to lose bits when transmitting the signal u. Thus, one can see that this represents a quintessential example of a real-world NCS.

In the following sections we present core ideas regarding the SNR approach to NCS design. We will return to the power control example for WCDMA in Section 25.7. To motivate some of the issues studied in the remainder of this chapter, we next present a simple example inspired by the WCDMA power control problem.

Consider the NCS of Figure 25.2, with loop delay $\beta = 1$, white-noise disturbance $d \triangleq g - I$, no reference (i.e., $r = 0$), and $K = 1$. (This choice for K corresponds to the usual choice in practice [16].) Define $G \triangleq z^{-\beta}/1 - z^{-1} = 1/z - 1$. Assume that the quantizer is uniform, and that it can be modeled as an AWN source q [17]. Also assume that one is allowed to modify the WCDMA feedback loop as shown in Figure 25.3. In that figure, A is a stable, minimum-phase and biproper linear time invariant (LTI) filter that can be chosen by the designer. If the quantization noise q is uncorrelated with d, and d is assumed to be white, then the input to the quantizer u' and the received SIR y have stationary variances given by

$$\sigma_{u'}^2 = \sigma_d^2 \left\| A^{-1}S \right\|_2^2 + \sigma_q^2 \left\| GS \right\|_2^2, \tag{25.3a}$$

$$\sigma_y^2 = \sigma_d^2 \left\| S \right\|_2^2 + \sigma_q^2 \left\| AGS \right\|_2^2, \tag{25.3b}$$

where σ_q^2 and σ_d^2 are the variances of q and d, $S \triangleq (1+G)^{-1}$, and $\|X\|_2^2 \triangleq \frac{1}{2\pi}\int_{-\pi}^{\pi} XX^H d\omega^*$. The variance of q depends on the quantizer parameters which, in turn, have to be chosen in accordance with the statistics of u'. Assume that relationship translates into the following SNR constraint (a detailed discussion of this fact is included in Section 25.6.2):

$$\frac{\sigma_{u'}^2}{\sigma_q^2} \leq \Gamma, \tag{25.4}$$

for some $\Gamma > 0$ that depends on the number of quantization levels. Using Equation 25.4 in Equation 25.3 yields, for any σ_q^2 (i.e., any quantizer parameters) and any filter A,

$$\sigma_y^2 \geq \sigma_d^2 \left\| S \right\|_2^2 + \frac{\sigma_d^2 \left\| A^{-1}S \right\|_2^2}{\Gamma - \left\| GS \right\|_2^2} \left\| AGS \right\|_2^2 \tag{25.5}$$

$$\geq \sigma_d^2 \left\| S \right\|_2^2 + \frac{\sigma_d^2}{\Gamma - \left\| GS \right\|_2^2} \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} |G|\,|S|^2 \right)^2 \triangleq [\sigma_y^2]_{\text{opt}}, \tag{25.6}$$

* Here, $(\cdot)^H$ stands for the conjugate transpose of $(\cdot)$.

where Equation 25.6 follows from the Cauchy–Schwartz inequality. In Equation 25.5 equality holds if the quantizer parameters are chosen such that

$$\sigma_q^2 = \sigma_d^2 \left\| A^{-1} S \right\|_2^2 \left(\Gamma - \|GS\|_2^2 \right)^{-2}. \tag{25.7}$$

Equality in Equation 25.6 holds if the filter A satisfies $|A| = \sqrt{|G^{-1}|}$. In practice, the latter condition requires the approximation of $\sqrt{|G^{-1}|}$ by a stable, minimum-phase and biproper LTI filter A [18,19].

The reader might have noticed that in Equations 25.5 and 25.7 an implicit constraint on Γ is imposed, namely, $\Gamma > \|GS\|_2^2$. It can be shown that this condition is equivalent to the existence of a filter A that renders the NCS stable in the mean square sense. This issue is explored in greater detail in Section 25.5.

Since $G = 1/(z-1)$, we have from Equations 25.5 and 25.6 that the best achievable performance for the proposed architecture is given by

$$[\sigma_y^2]_{\text{opt}} = \left(2 + \frac{1.62}{\Gamma - 1} \right) \sigma_d^2. \tag{25.8}$$

On the other hand, if one retains the standard choice $A = 1$, then

$$\sigma_y^2 = \left(2 + \frac{2}{\Gamma - 1} \right) \sigma_d^2. \tag{25.9}$$

We see from the above that a simple modification of the architecture of Figure 25.2 allows one to improve performance. The improvements are more dramatic if Γ is small, which corresponds to a quantizer with few levels (Section 25.6.2).

The previous example utilized a relatively simple architectural change in the NCS of Figure 25.2. However, more complex choices are also possible. To introduce such architectures, we will begin by describing the general setup adopted in this chapter.

25.3 A General Setup for the Analysis of NCSs

In this and the next three sections, we focus on the NCS of Figure 25.4, which includes the WCDMA power control loop of Figure 25.2 as a special case. In Figure 25.4, G is a given single-input single-output (SISO) LTI plant, M and N are LTI systems to be designed, d models output disturbances, u is the control input, and y is a signal available for measurement. The setup of Figure 25.4 includes a nontransparent channel in the feedback path with input $\bar{v}$ and output $\bar{w}$. As suggested by the dashed line in Figure 25.4, we will also consider situations where the channel output may be available at the sending end with one sample of delay. The latter assumption is natural in some settings, while inappropriate in others. We will examine both cases in this chapter. The blocks $\mathcal{E}$ and $\mathcal{D}$ on either side of the channel correspond to (possibly) nonlinear systems that map the real-valued signal v into the sequence of channel symbols $\bar{v}$ (e.g., quantized values in digital channels), and the channel output $\bar{w}$ into the real-valued signal w, respectively.

Our aim is to design the LTI systems M and N and the blocks $\mathcal{E}$ and $\mathcal{D}$ in Figure 25.4 so as to minimize the stationary variance of y, subject to the communication constraints imposed by the channel. Our analysis is intended to cover the following scenarios:

- *AWN channel:* In this channel model, the data sent over the channel is additively corrupted by a noise source and there exists a maximum allowable channel input power [11]. Although very simple, this channel model lies at the very foundation of communication theory [20], and is usually used to model wireless links at the physical level [21]. For this model, the feedback path around the channel is usually not available.

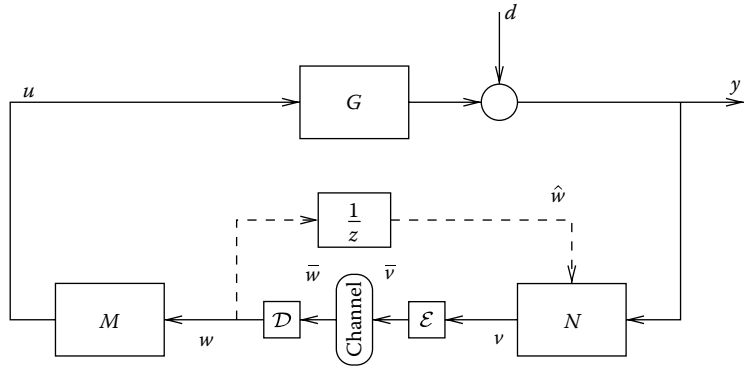

FIGURE 25.4 General networked control architecture.

- *Noiseless digital channel with finite alphabet:* This channel model can be used to describe situations where the communication link can transmit, without errors, symbols from a given finite set [22,23]. This corresponds to an abstraction which is useful when error correcting algorithms are employed in a way such that the higher level communication protcol observes transparent communication. In contrast to the previous case, assuming feedback around the channel is natural here since no channel errors are considered.
- *Noiseless digital channel with constrained average data rate:* This channel model corresponds to a relaxation of the previous one. In this case, the constraints are imposed, not on the cardinality of the channel alphabet, but on the time-average length of the binary words that represent the symbols in that alphabet [24,25]. By doing so, one can use information theoretic ideas [20] to design systems where communication takes place over digital links. As in the previous case, feedback around the channel is a natural assumption here.
- *Bernoulli erasure channel:* This channel model is used to describe situations where data is prone to get lost with a given probability (e.g., when wireless links are employed) [4]. In this scenario, it is common for the underlying protocol to provide acknowledgements that testify successful transmissions (TCP-like protocols; [4]). If that is the case, then the assumption of feedback around the channel is justified. If no such acknowledgements are present (UDP-like protocols), then feedback around the channel cannot be exploited.

Remark 25.1

In this chapter we treat the cases with and without channel feedback in parallel. If, at some stage, we do not explicitly mention if channel feedback is available or not, then our discussion applies, *mutatis mutandis*, to both cases.

Control problems where communication takes place over the channels mentioned above have received much attention in the literature (see the references above and those in [7]). In this chapter we show that all of the cases mentioned above can be handled, in a unified fashion, by reducing the corresponding design problem to a problem of control system design over a SNR constrained AWN channel. Such problems are addressed in Sections 25.4 and 25.5. Later, in Section 25.6, we give further details as to how the above channels can be viewed as SNR constrained channels.

To simplify our subsequent exposition we introduce the following standard assumptions:

Assumption 25.1:

1. *The plant G is SISO, LTI, strictly proper, and has a stabilizable and detectable underlying realization.*

2. *The initial state of the plant and LTI systems M and F (collectively referred to as x_o) are jointly second order random variables.*
3. *The disturbance d is a zero mean second order wide sense stationary (wss) sequence, uncorrelated with x_o, and having power spectral density $|\Omega_d|^2 > 0$.*

25.4 Architectures for Control over SNR Constrained AWN Channels

Consider the NCS of Figure 25.4 and assume that the link between v and w is a scalar SNR constrained AWN channel (Figure 25.5). Such a channel has a scalar input v and a scalar output w related via

$$w = v + q, \tag{25.10}$$

where q is a zero mean white noise sequence, uncorrelated with $[x_o^T \;\; d^T]^T$, having a finite variance σ_q^2 that can be chosen arbitrarily subject to the SNR constraint

$$\gamma \triangleq \frac{\sigma_v^2}{\sigma_q^2} \leq \Gamma, \tag{25.11}$$

for some given and finite Γ. In Equation 25.11, σ_v^2 denotes the stationary variance of v, which exists whenever Assumption 25.1 holds, $\sigma_q^2 < \infty$, and the feedback loop of Figure 25.5 is internally stable.

A key property of SNR constrained AWN channels is that one can choose the noise variance σ_q^2. As will be made explicit in Section 25.6, this seemingly unusual property arises naturally when one studies the communication channels mentioned in Section 25.3.

We focus on the following variants of the architecture of Figure 25.5:

1. *Perfect reconstruction coding scheme:* Assume that an LTI plant G_o is given and an LTI controller C_o is designed so as to provide satisfactory performance under the assumption of perfect communications, that is, $u = y$. In this case it is natural to choose M and N so as to achieve unity transfer from y to u, and thus preserve the closed-loop design relations, and make the effects of the communication link purely additive [19]. We can thus think of G as comprising both the plant G_o and the controller C_o, with $G \triangleq G_o C_o$.

   For the situation described above, we propose to use the architecture of Figure 25.6, where A and F are LTI systems to be designed. In order to account for the possibility that w is available at the sending end with one sample of delay, we constrain F to be strictly proper in this case. For this

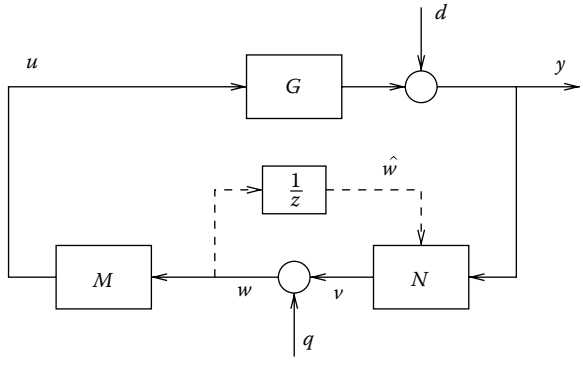

FIGURE 25.5 General NCS closed over AWN channel.

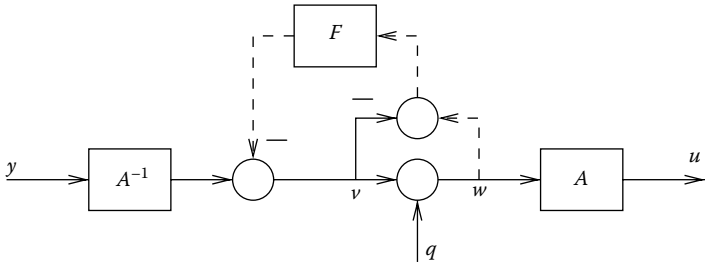

FIGURE 25.6 Perfect reconstruction coding scheme.

architecture,

$$u = y + A(1 - F)q. \tag{25.12}$$

Thus, by proper choice of A and F one can spectrally shape the effects of the noise q on u and hence, on the output y. For the scenario when it is not possible to have feedback around the channel, $F = 0$ is the only admissible choice.

The architecture of Figure 25.6 will be referred to as a perfect reconstruction coding scheme.

When a perfect reconstruction coding scheme is used, then internal stability and well posedness of the NCS of Figure 25.5 is equivalent to A being stable, biproper and minimum-phase, and F being stable and strictly proper.

2. *One-block architecture:* In this case we fix $M = 1$ and consider N as the design parameter (Figure 25.7). The role of $N \triangleq [N_1 \;\; N_2]$ is that of a standard controller and thus the conditions on N that ensure internal stability and well posedness of the resulting NCS are the standard ones [26]. In this case,

$$u = q + N_1 y + N_2 \hat{w}. \tag{25.13}$$

For the scenario when feedback around the channel is available, then each of the components of N can be freely chosen, and we refer to the scheme as a one-block architecture with feedback. For the alternative scenario when feedback is not available, then $N = [N_1 \;\; 0]$, and the resulting controller will be referred to as a one-block architecture with no feedback (or one degree of freedom architecture).

3. *Two-block architecture.* This architecture corresponds to that of Figure 25.5, with no constraints upon M or N, except for internal stability and well posedness of the resulting NCS. This architecture corresponds to the most general architecture that can be constructed using only LTI blocks and (possibly) feedback around an SNR constrained AWN channel.

   For the scenario when feedback around the channel is available, then each of the components of N can be freely chosen, and we refer to the scheme as a two-block architecture with feedback. For the scenario when feedback is not available, then $N = [N_1 \;\; 0]$, and the resulting controller will be referred to as a two-block architecture with no feedback.

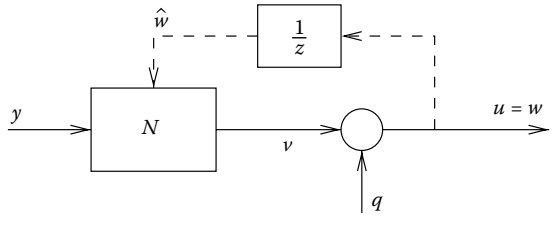

FIGURE 25.7 One-block control architecture.

The set of architectures presented above is by no means exhaustive. However, they correspond to paradigmatic cases of relevance to many applications.

25.5 Optimal Design of NCSs over SNR Constrained AWN Channels

Before addressing the optimal design of the architectures presented in Section 25.4, we will begin by focusing on the conditions that allow one to achieve stability while satisfying the SNR constraint. Such a study is fundamental to assess the feasibility of the design problems.

25.5.1 Mean Square Stability

We adopt the following notion of stability:

Definition 25.1:

*Consider the linear system $x(k+1) = Ax(k) + Bw(k)$, where $k \in \mathbb{N}_0$, A, B are constant matrices of appropriate dimensions, $x(k) \in \mathbb{R}^n$ is the system state at time instant k, $x(0) = x_o$, x_o is a second order random variable, and the input w is a second order wss process uncorrelated with x_o. The system is said to be mean square stable (MSS) if and only if there exist finite $\mu \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n\times n}$, $M \geq 0$, such that**

$$\lim_{k\to\infty} \mathbb{E}\{x(k)\} = \mu, \quad \lim_{k\to\infty} \mathbb{E}\left\{x(k)x(k)^T\right\} = M,$$

regardless of the initial state x_o. ■

It is well-known [27] that, for linear systems, MSS is equivalent to internal stability (in the standard sense [26]). Hence, provided Assumption 25.1 holds and $\sigma_q^2 < \infty$, the architectures presented in Section 25.4 will be MSS if and only if either A and F, or M and N internally stabilize G.

The next theorem characterizes the minimal SNR that is compatible with MSS for all of the architectures described above, under the assumption that feedback around the channel is available [8,12].

Theorem 25.1:

Consider the NCS of Figure 25.5, where q is the noise in an SNR constrained AWN channel, and Assumption 25.1 holds. For the scenario when feedback around the channel is available, and either M and N are such that the control architecture corresponds to a one- or two-block architecture, or G is assumed to be stabilizable with unity feedback (i.e., with $u = y$), and M and N are such that the control architecture corresponds to a perfect reconstruction coding scheme, then

$$\inf\{\gamma : \textit{MSS holds}\} = \gamma_\infty \triangleq \left(\prod_{i=1}^{n_p} |p_i|^2\right) - 1, \tag{25.14}$$

where p_i denotes the ith strictly unstable pole of G and the minimization is carried out with respect to the channel noise variance σ_q^2, and the LTI filters in the corresponding architecture. The infimal SNR γ_∞ is, in general, not achievable unless $\sigma_q^2 \to \infty$.† ■

* $\mathbb{E}\{\cdot\}$ denotes the expectation operator.

† That is, γ_∞ is an infimum and not a minimum of the SNR γ.

We conclude from Theorem 25.1 that, when feedback around the channel is available, it is possible to choose a noise variance σ_q^2 and the LTI filters in any of the architectures under consideration so as to achieve MSS if and only if the bound on the channel SNR Γ satisfies $\Gamma > \gamma_\infty$. Interestingly, the minimal SNR compatible with MSS, that is, γ_∞, is, for all the architectures, a function of the unstable poles of G only.

We note that in the case of perfect reconstruction coding schemes, G is assumed to be stabilizable with unity feedback. This means that the poles p_i in Equation 25.14 correspond to the unstable poles of the underlying plant G_o and those of the controller C_o (Section 25.4). This implies that, for strongly stabilizable plants, there always exists a controller C_o such that the SNR requirements for MSS are equal in all three architectures (with feedback). However, if the plant is not strongly stabilizable, then every stabilizing C_o is unstable. Hence, for such plants, the SNR requirements for MSS will be higher when a perfect reconstruction coding scheme is used.

A key element of Theorem 25.1 is that each of the architectures exploits feedback around the channel. For the alternative scenario when such feedback is not available, one has the following result [11,12]:

Theorem 25.2:

Consider the NCS of Figure 25.5, where q is the noise in an SNR constrained AWN channel, no feedback around the channel is available, and Assumption 25.1 holds. Then:

1. *If G is assumed to be stabilizable with unity feedback, and M and N are such that the control architecture corresponds to a perfect reconstruction coding scheme with $F = 0$, then*

$$\inf\{\gamma : \textit{MSS holds}\} = ||T||_2^2 \geq \gamma_\infty, \tag{25.15}$$

   where $T \triangleq G(1-G)^{-1}$ and $||\cdot||_2$ denotes the usual norm in $\mathcal{L}_2$ [26]. Equality in Equation 25.15 holds if and only if $\left(\prod_{i=1}^{n_p} \frac{1-zp_i}{z-p_i}\right)\frac{1}{1-G}$ is constant.
2. *If M and N are such that the control architecture corresponds to a one- or two-block architecture without feedback, then*

$$\inf\{\gamma : \textit{MSS holds}\} = \gamma_{\infty,\,\mathrm{nf}} \triangleq \left(\prod_{i=1}^{n_p} |p_i|^2\right) - 1 + \Delta_G, \tag{25.16}$$

   where p_i is as in Theorem 25.1, $\Delta_G \geq 0$ depends on the nonminimum phase zeros and relative degree of G, and $\Delta_G = 0$ if and only if the plant is either stable, or has no finite nonminimum phase zeros outside the closed unit disk, and has relative degree one (Equation 34 in [11] for an explicit expression for Δ_G).

In both Equations 25.15 and 25.16, the minimization is carried out as in Theorem 25.1 with the additional constraint $F = 0$ or $N_2 = 0$. As before, the infimal SNRs are, in general, not achievable unless $\sigma_q^2 \to \infty$.

We see from Theorem 25.2 that, for the scenarios where no feedback around the channel is available, the minimal SNR compatible with MSS depends on more than the plant unstable poles. In the case of a perfect reconstruction coding scheme, the whole plant G plays a role, whilst in the case of one- or two-block architectures, the plant relative degree and its nonminimum phase zeros become features of importance. It follows from Theorems 25.1 and 25.2 that, in general, the SNR requirements on the channel for MSS, when no feedback around the channel is available, are more stringent than those that apply for the scenarios when such feedback can be exploited. This is natural since, in the former case, the control architectures have less degrees of freedom than the architectures that exploit channel feedback.

So far, we have explored the interplay between the SNR constraint Γ and the MSS of diverse architectures for SNR constrained AWN channels. The next three sections address the problem of optimally designing the various blocks that appear in these architectures.

25.5.2 Design for the Perfect Reconstruction Coding Scheme

In this case, the designer can choose the postfilter A, the feedback filter F and the channel noise variance σ_q^2. We will begin by considering the design of perfect reconstruction coding schemes with no feedback, that is, with $F = 0$. The next result presents closed form expressions for both the best performance and the choices of A and σ_q^2 that allow one to asymptotically achieve such performance [18,19]:

Theorem 25.3:

Consider the NCS of Figure 25.5, where q is the noise in an SNR constrained AWN channel, no feedback around the channel is available, and Assumption 25.1 holds. If G is stabilizable with unity feedback, M and N are such that the control architecture corresponds to a perfect reconstruction coding scheme with $F = 0$, *and* $\Gamma > ||T||_2^2$, *then:*

1. *The best achievable performance is given by*

$$\inf\{\sigma_y^2 : \gamma \leq \Gamma, \textit{MSS holds}\} = ||(T+1)\Omega_d||_2^2 + \frac{\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} |(T+1)|\,|T|\,|\Omega_d|\,d\omega\right)^2}{\Gamma - ||T||_2^2}, \tag{25.17}$$

   and the SNR constraint is active at the optimum.
2. *In order to attain a performance arbitrarily close to optimal, whilst satisfying the SNR constraint with equality, it suffices to pick a stable, minimum phase and biproper A that provides a sufficiently good approximation to the idealized optimal choice for A, namely to* A_i *satisfying*

$$|A_i|^2 = \frac{|\Omega_d|}{|G|}, \tag{25.18}$$

   and to choose $\sigma_q^2 = \frac{||A^{-1}(T+1)\Omega_d||_2^2}{\Gamma - ||T||_2^2}$.

If the square root on the RHS of Equation 25.18 is rational and has no poles or zeros on the unit circle, then no approximations are needed when choosing A. Accordingly, the best performance is actually achievable. Even if this is not the case, then it is normally sufficient to consider reasonably low order filters to approximate A_i [19].

We now return to the general case of perfect reconstruction coding schemes and consider the scenario where feedback around the channel is available, that is, we assume that F can also be chosen. The following theorem is a consequence of the results in [18]:

Theorem 25.4:

Consider the NCS of Figure 25.5, where q is the noise in an SNR constrained AWN channel, feedback around the channel is available, and Assumption 25.1 holds. If G is assumed to be stabilizable with unity feedback, M and N are such that the control architecture corresponds to a perfect reconstruction coding scheme, and $\Gamma > \gamma_\infty$, *then:*

1. *The best achievable performance is given by*

$$\inf\{\sigma_y^2 : \gamma \leq \Gamma, \textit{MSS holds}\} = ||(T+1)\Omega_d||_2^2 + \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\lambda_\Gamma Y}{2\left(\sqrt{Y^2 + \lambda_\Gamma |T+1|^2} + Y\right)}\,d\omega, \tag{25.19}$$

where $Y \triangleq |T+1|^2\,|G\Omega_d|$, *and* λ_Γ *is the unique positive real satisfying*

$$\Gamma = g(\lambda_\Gamma) \triangleq \exp\left(\frac{1}{\pi}\int_{-\pi}^{\pi} \ln\left(\sqrt{\frac{Y^2}{\lambda_\Gamma}+|T+1|^2}+\frac{Y}{\sqrt{\lambda_\Gamma}}\right)d\omega\right)-1. \tag{25.20}$$

Moreover, the SNR constraint is active at the optimum.

2. *In order to attain performance arbitrarily close to optimal, whilst satisfying the SNR constraint with equality, it suffices to pick a stable, minimum phase and biproper A, and a stable and strictly proper F, such that they provide sufficiently good approximations to the corresponding idealized optimal choices, namely to* A_i *and* F_i *satisfying*

$$|A_i|^2 = \frac{|\Omega_d|}{|1-F_i|\,|G|}, \quad |1-F_i| = \frac{2(\Gamma+1)\alpha_\Gamma}{\sqrt{Y^2+4(\Gamma+1)\alpha_\Gamma^2\,|T+1|^2}+Y}, \tag{25.21}$$

where $\alpha_\Gamma \triangleq \frac{1}{2}\sqrt{\frac{\lambda_\Gamma}{\Gamma+1}}$, and to choose

$$\sigma_q^2 = \alpha_\Gamma. \tag{25.22}$$

Theorem 25.4 gives, for any feasible upper bound Γ on the channel SNR, a closed form expression for the best performance achievable when a perfect reconstruction coding scheme is employed. Our result is given in terms of the scalar parameter λ_Γ that satisfies $g(\lambda_\Gamma) = \Gamma$ (see Equation 25.20). Since $g(\cdot)$ is a monotone function of is argument, it follows that finding λ_Γ reduces to a simple numerical problem that can be addressed using standard algorithms. As was the case in Theorem 25.2, it is usually sufficient to consider reasonably low-order filters to approximate A_i or F_i.

25.5.3 Design for One-Block Architectures

In the previous section we focused on the case where the plant G can be stabilized with unity feedback. If one removes that assumption, then one-block architectures become suitable choices.

Irrespective of whether one uses a one-block architecture with or without feedback, the resulting NCS can be written in the generic form shown in Figure 25.8, where P is partitioned into blocks P_{ij} such that

$$\begin{bmatrix} y \\ v \\ \hline \bar{y} \end{bmatrix} = \left[\begin{array}{c|c} P_{11} & P_{12} \\ \hline P_{21} & P_{22} \end{array}\right] \begin{bmatrix} d \\ q \\ \hline \bar{u} \end{bmatrix}. \tag{25.23}$$

For one-block architectures without feedback, we have (*cf.* Figures 25.5 and 25.7)

$$P = \left[\begin{array}{cc|c} 1 & G & G \\ 0 & 0 & 1 \\ \hline 1 & G & G \end{array}\right], \quad K = N_1, \quad \bar{y} = y, \quad \bar{u} = v. \tag{25.24}$$

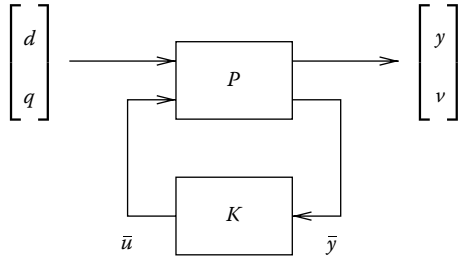

FIGURE 25.8 Equivalent rewriting of one-block architectures ($\bar{u}$ and $\bar{y}$ are defined in Equations 25.24, 25.25).

If feedback is available, then

$$P = \left[\begin{array}{cc|c} 1 & G & G \\ 0 & 0 & 1 \\ \hline 1 & G & G \\ 0 & z^{-1} & z^{-1} \end{array}\right], \quad K = N, \quad \bar{y} = \begin{bmatrix} y \\ \hat{w} \end{bmatrix}, \quad \bar{u} = v. \tag{25.25}$$

In contrast to the case considered in the previous section, there does not exist an *explicit closed form* expression for the optimal performance when one-block architectures are employed. Several approaches have been proposed in the literature and we refer the reader to [12] for details. Here, we will content ourselves with showing that the optimal design for one-block architectures can be addressed by using standard procedures.

Consider Figure 25.8. Our aim is to find

$$J_{\text{opt}} \triangleq \inf\{\sigma_y^2 : \ \gamma \leq \Gamma, \text{MSS holds}\}, \tag{25.26}$$

where the optimization is carried out over all finite σ_q^2 and all the filters $K \in \mathcal{S}$, where $\mathcal{S}$ is the set of all LTI and proper K that ensure internal stability and well posedness of the feedback loop of Figure 25.8. Given Theorem 25.1, it is clear that, for one-block architectures with feedback, the optimization problem in Equation 25.26 is feasible if and only if $\Gamma > \gamma_\infty$. If no feedback is available, then the condition for feasibility becomes $\Gamma > \gamma_{\infty,\text{nf}}$ (Theorem 25.2).

By definition of γ (Equation 25.11),

$$J_{\text{opt}} = \inf_{\sigma_q^2 < \infty} \inf_{K \in \mathcal{S}} \{\sigma_y^2 : \ \sigma_v^2 \leq \Gamma \, \sigma_q^2 \}. \tag{25.27}$$

For any fixed σ_q^2, the inner problem in Equation 25.27 is a standard quadratic optimal control problem subject to a quadratic constraint. Thus, if feasible, the inner problem in Equation 25.27 can be addressed using any standard method based on, for example, LMIs [28], inner outer factorizations [12], and so on. A necessary and sufficient condition for the inner problem in Equation 25.27 to be feasible is that

$$\sigma_q^2 \in \left\{ x^2 : 0 < x^2 < \infty \text{ and } \inf_{K \in \mathcal{S}} \sigma_v^2 \leq \Gamma \, x^2 \right\}. \tag{25.28}$$

Again, testing whether or not σ_q^2 satisfies (Equation 25.28) reduces to a standard unconstrained quadratic optimal control problem.

It follows from the above that by employing any line search solver, coupled with a method for solving quadratically constrained quadratic optimal control problems, one will be able to easily obtain (an approximation of) J_{opt}, and the corresponding optimal noise variance σ_q^2 and optimal LTI filter N.

The following result presents conditions that guarantee that the SNR constraint is active at the optimum [12]:

Lemma 25.1:

Consider the NCS of Figure 25.5, where q is the noise in an SNR constrained AWN channel, Assumption 25.1 holds, and $M = 1$. If feedback around the channel is available (resp. not available), $\Gamma > \gamma_\infty$ (resp. $\Gamma > \gamma_{\infty,\text{nf}}$), and the transfer function from the disturbance d to the channel input v is nonzero at the optimum, then the SNR constraint is active at the optimum (or can be made active, without compromising optimality).

We note that if the transfer function from d to v is zero at the optimum, then optimal performance would be achieved without sending any information about d over the unreliable channel. Clearly, this case is of little interest in NCSs. We thus conclude that the SNR constraint will be active at the optimum for most cases of interest.

25.5.4 Design for the General Architecture

We now focus on the general architecture of Figure 25.5. In principle, one may think that it suffices to rewrite the NCS as shown in Figure 25.8 and to apply the ideas in the previous section. We show below that this is not the case.

To fix ideas, we consider a two-block architecture without feedback (similar comments apply to two-block architectures with feedback). The resulting NCS can be written as in Figure 25.8 with

$$P = \left[\begin{array}{cc|cc} 1 & 0 & 0 & G \\ 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & G \\ 0 & 1 & 1 & 0 \end{array}\right], \quad K = \begin{bmatrix} N_1 & 0 \\ 0 & M \end{bmatrix}, \quad \bar{y} = \begin{bmatrix} y \\ w \end{bmatrix}, \quad \bar{u} = \begin{bmatrix} v \\ u \end{bmatrix}. \tag{25.29}$$

The zero entries in K originate from the communication constraints, that is, from the fact that u cannot depend on y, and that v cannot depend on w. We see that the optimal design of K becomes an optimal control problem with *sparsity constraints* on the controller. Such problems are known to be inherently difficult since, in general, there exists no known means to convexify them [29]. The widest known class of sparsity constrained problems for which a convex reformulation is possible has been identified in [29]. Unfortunately, as a simple calculation shows, the problem at hand does not fit into that class of problems.

It follows that the joint optimal design of M and N in Figure 25.5 is a hard problem. A simple way to overcome this difficulty is to use the ideas presented in the previous sections in an iterative fashion. There exist at least two possibilities:

1. *Design via a sequence of one-block problems:* Irrespective of whether or not feedback around the channel is available, one can use the results in Section 25.5.3 to design M and N as follows: fix M (e.g., $M = 1$) and choose N so as to optimize performance subject to the SNR constraint Γ. For that choice of N, choose M so as to optimize performance subject to the SNR constraint Γ, and repeat *ad infinitum*. At each step of the procedure outlined above, one needs to solve an optimization problem that fits into the class of one-block problems studied in Section 25.5.3, but, of course, with a different P block. We leave the details to the reader.
2. *Design via perfect reconstruction coding scheme and one-block problem:* As an alternative (but not equivalent) method, one can write $M = CA$ and consider N as in the case of perfect reconstruction coding schemes. The resulting control architecture is illustrated in Figure 25.9.

   As a first design step, we propose to choose C as a stabilizing controller for G when $\hat{u} = y$. Then, design A and F using the results of Section 25.5.2 with plant given by GC. As a third step, we

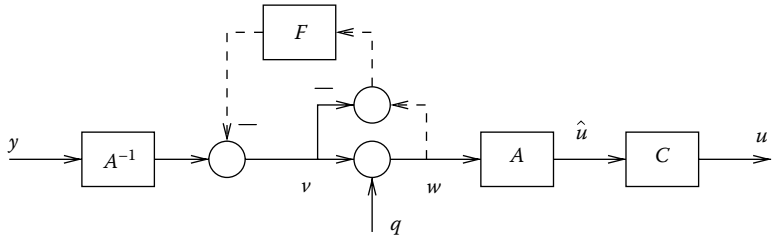

FIGURE 25.9 Perfect reconstruction coding scheme plus stabilizing controller.

propose to use the ideas of Section 25.5.3 to optimally design C for the choices of filters made in the previous step. To that end, note that for given A and F the architecture of Figure 25.9 can be written as in Figure 25.8 with

$$P = \left[\begin{array}{cc|c} 1 & 0 & G \\ A^{-1} & -F & A^{-1}G \\ \hline 1 & A(1-F) & G \end{array}\right], \quad K = C, \quad \bar{u} = u, \quad \bar{y} = \hat{u}. \tag{25.30}$$

Repeat the above procedure *ad infinitum*.

The procedures outlined above converge, at least, to a local minimum. Different initial choices for M or C can be utilized to reduce the conservatism of the approach.

25.6 Turning Communication Constraints into SNR Constraints

In the previous two sections we have focused on SNR constrained AWN channels. In particular, we studied the interplay between MSS and SNR constraints for the three architectures described in Section 25.4, and also presented design guidelines for them.

In this section we show that the results in Section 25.5 can be readily applied to the NCS of Figure 25.4 when the channel is either an AWN channel, a noiseless digital channel with finite alphabet, a noiseless digital channel with constrained average data-rate, or a Bernoulli erasure channel. To that end, an appropriate choice for $\mathcal{E}$ and $\mathcal{D}$ is instrumental.

25.6.1 AWN Channels

The AWN channel is the simplest channel where a proper choice for $\mathcal{E}$ and $\mathcal{D}$ turns the link between v and w in Figure 25.4 into an SNR constrained AWN channel.

In an AWN channel, the input $\bar{v}$ and output $\bar{w}$ are related via [21, Chapter 4]

$$\bar{w} = \bar{v} + \bar{q}, \tag{25.31}$$

where $\bar{q}$ is a fixed white-noise sequence, uncorrelated with $[x_o^T \;\; d^T]^T$, and where the stationary variance of $\bar{v}$ is subject to $\sigma_{\bar{v}}^2 \leq V$.

Clearly, the signal $\bar{v}$ is a scaled version of the signal of interest. It is thus reasonable to use an AWN channel with both pre- and postscaling factors and, accordingly, we choose $\mathcal{E}(v(k)) = \alpha^{-1}v(k)$ and $\mathcal{D}(\bar{w}(k)) = \alpha\bar{w}(k)$, as depicted in Figure 25.10a. In that figure, α is a real parameter to be chosen by the designer. It is immediate to see that the signals v and w in Figure 25.10a are related via $w = v + q$, where the equivalent noise $q \triangleq \alpha\bar{q}$ is a zero mean white noise sequence with variance $\sigma_q^2 = (\alpha\sigma_{\bar{q}})^2$, which is uncorrelated with $[x_o^T \;\; d^T]^T$. Since α is a designer's choice, σ_q^2 is also a design parameter. With the previous definitions, the variance constraint on $\bar{v}$ becomes equivalent to

$$\gamma = \frac{\sigma_v^2}{\sigma_q^2} \leq \frac{V}{\sigma_{\bar{q}}^2} = \Gamma, \tag{25.32}$$

that is, equivalent to an SNR constraint.

We conclude from the above discussion that the results of Section 25.5 can be directly applied to NCSs with pre- and postscaled AWN channels. Once σ_q^2 has been chosen in the equivalent SNR constrained design problem, then the scaling factor α defining $\mathcal{E}$ and $\mathcal{D}$ can be obtained from $\alpha^2 = \sigma_q^2\sigma_{\bar{q}}^{-2n}$.

We acknowledge that assuming the existence of feedback around an AWN channel may be inappropriate in some situations. Indeed, if such a transparent feedback channel were available, then it could be used to replace the forward AWN channel between $\bar{v}$ and $\bar{w}$, and no communication constraints would arise.

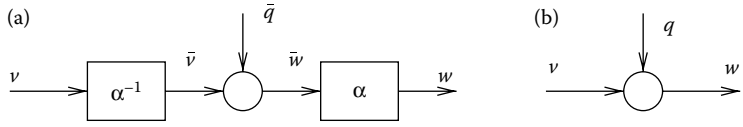

FIGURE 25.10 (a) AWN channel with pre- and postscaling factors, and (b) equivalent rewriting.

25.6.2 Noiseless Digital Channels with Finite Alphabet

In a noiseless digital channel with finite alphabet, the input $\bar{v}$ and output $\bar{w}$ are related via

$$\bar{w}(k) = \bar{v}(k) \tag{25.33}$$

whenever $\bar{v}(k)$ belongs to a given finite set called the channel alphabet. Without loss of generality, we assume the channel alphabet to be given by $\{-\Delta(L-1)/2, \ldots, -\Delta, 0, \Delta, \ldots, \Delta(L-1)/2\}$ for some odd positive integer L, and some positive real Δ.

To deal with a finite alphabet digital channel, $\mathcal{E}$ must contain a quantizer. We focus on the simplest quantizer, a (L-level) finite uniform quantizer, that is, we consider $\mathcal{E}$ defined via

$$\mathcal{E}(v(k)) = \mathcal{Q}_L(v(k)) \triangleq \begin{cases} V & \text{if } v(k) > V + \Delta/2, \\ \Delta \text{ round } \left(\frac{v(k)}{\Delta}\right) & \text{if } |v(k)| \le V + \Delta/2, \\ -V & \text{if } v(k) < -V - \Delta/2, \end{cases} \qquad V \triangleq \frac{\Delta(L-1)}{2}, \tag{25.34}$$

where L is as above, Δ is the quantization step, and round $(\cdot)$ denotes rounding toward the nearest integer (Figure 25.11). We also make $\mathcal{D}(\bar{w}(k)) = \bar{w}(k)$, and define the quantization noise sequence q via

$$q \triangleq w - v. \tag{25.35}$$

Quantization is a highly nonlinear operation. Hence, the exact characterization of the quantization noise q is difficult. However, a useful approximation that is frequently used in the literature is to model the uniform quantization noise q as a white-noise sequence, uniformly distributed in $\left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$, and uncorrelated with the input of the quantizer [17]. (In our case it makes more sense to assume q to be uncorrelated with $[x_o^T \;\; d^T]^T$.)

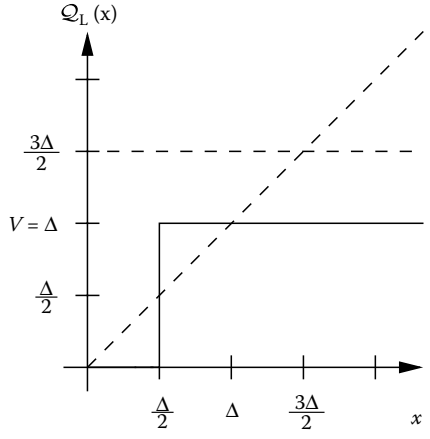

FIGURE 25.11 3-level finite uniform quantizer (only $x \ge 0$ is shown).

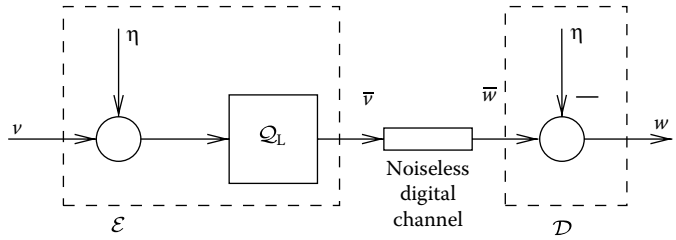

FIGURE 25.12 Dithered quantizer used to transmit over finite alphabet digital channel.

The AWN model for quantization does not hold exactly in general. It is possible, however, to establish necessary and sufficient conditions on the input v in order for the additive noise model to be valid (e.g., [30]). These results assume, among other things that either $L \to \infty$ (i.e., that the quantizer is infinite) or that $|v(k)| \leq V + \Delta/2$ for every time instant (i.e., that no overload occurs). The results are, however, quite restrictive to be useful in general. Nevertheless, if Δ is small compared with the magnitude of v, the quantizer does not overload, and the samples of the input v have a smooth probability density, then the AWN model approximately holds with a high degree of accuracy [31,32].

A well-known way of making the AWN model valid is to add an auxiliary random signal η to the signal of interest prior to quantization, and then subtract it at the receiving end (Figure 25.12) [31,32]. The auxiliary random signal is called a *dither signal*, and is assumed to be available at both the sending and receiving ends. In this case,

$$\mathcal{E}(v(k)) = \mathcal{Q}_L(v(k) + \eta(k)), \quad \mathcal{D}(\bar{w}(k)) = \bar{w}(k) - \eta(k) \tag{25.36}$$

and thus

$$w(k) = \mathcal{Q}_L(v(k) + \eta(k)) - \eta(k). \tag{25.37}$$

The next theorem ensures that, even for quantizers embedded within feedback systems, subtractive dither makes the AWN model exact [33]:

Theorem 25.5:

Consider the feedback scheme of Figure 25.4 where w and v are related via Equation 25.37, and Assumption 25.1 holds. If perfect communication (i.e., $w = v$) renders the feedback scheme internally stable and well-posed, $\mathcal{Q}_L$ does not overload, and the dither η is a random sequence such that the probability density function of $\eta(k)$ given (η^{k-1}, d, x_o) satisfies $f(\eta(k)|\eta^{k-1}, d, x_o) = f(\eta(k)) \sim \text{Unif}\left(-\Delta/2, \Delta/2\right)$, then the quantization noise $q \triangleq w - v$ is such that the probability density function of $q(k)$ given (q^{k-1}, d, x_o) satisfies $f(q(k)|q^{k-1}, d, x_o) = f(q(k)) \sim \text{Unif}\left(-\Delta/2, \Delta/2\right)$.

The previous theorem assumes that no-overload occurs in the quantizer (equivalently, that the quantizer is an infinite one, i.e., $L \to \infty$). If the quantizer is finite and both the disturbances d and initial states x_o have bounded support, then the dynamic range V can be adjusted so as to prevent overload. If, on the other hand, we consider the case of processes with unbounded support, then the stability of the feedback system in Figure 25.4 cannot be guaranteed [3]. In practice, however, this is not so important since it is possible to guarantee small overload probability, as discussed below.

Consider an L-level uniform quantizer with quantization step Δ. Assume that v is asymptotically wss and focus on the steady state. Choose the quantizer dynamic range V so that

$$V + \frac{\Delta}{2} \geq \alpha\sigma_x, \tag{25.38}$$

where α is the quantizer loading factor, and σ_x is the standard deviation of the actual input to the quantizer, that is, $x = v$ if an undithered quantizer is used and $x = v + \eta$ if dither is present. By choosing a suitable value for α, it is possible to attain arbitrarily small overload probabilities. For example, if x is Gaussian distributed and $\alpha = 4$, then (25.38) implies an overload probability of 6.33×10^{-5}. This is the so-called 4σ-rule [17].

In the case of undithered finite uniform quantizers, and provided the additive noise model for quantization holds, Equation 25.38 becomes equivalent to

$$\gamma = \frac{\sigma_v^2}{\sigma_q^2} \leq \frac{3L^2}{\alpha^2} = \Gamma, \tag{25.39}$$

where $\sigma_q^2 = \Delta^2/12$ is the variance of the equivalent quantization noise q. On the other hand, if we consider a dithered finite uniform quantizer, and the assumptions of Theorem 25.5 hold, then the additive noise model for quantization holds and Equation 25.38 becomes equivalent to

$$\gamma = \frac{\sigma_v^2}{\sigma_q^2} \leq \frac{3L^2}{\alpha^2} - 1 = \Gamma. \tag{25.40}$$

(We see from Equations 25.39 and 25.40 that, for a fixed number of quantization levels L and fixed loading factor α, a smaller SNR is achieved when using dithered quantizers. This is due to the fact that the dynamic range of a dithered quantizer needs to accommodate both the dither and the signal of interest.)

From the previous analysis we conclude that, if one uses a finite quantizer, aims at small overload probability, and either assumes the noise model to hold or uses subtractive dither, then an SNR constrained AWN channel arises. In this case, choosing the noise variance σ_q^2 amounts to choosing the quantization step Δ, whilst the SNR constraint aims at preventing overload.

The above discussion implies that all the results of Section 25.5 can be applied in this case with the choice $\Gamma = \frac{3L^2}{\alpha^2}$ when no dither is used, or with $\Gamma = \frac{3L^2}{\alpha^2} - 1$ when a dithered quantizer is employed. Note that once σ_q^2 has been chosen in the equivalent SNR constrained design problem, then the quantization step can be calculated via $\Delta = \sqrt{12\sigma_q^2}$.

For noiseless digital channels, the output is always equal to the input. Thus, it is natural to assume that feedback around the channel is available in this case.

Remark 25.2

We remind the reader that mean square stability of the NCS of Figure 25.4 cannot be guaranteed with the proposed control architecture, unless both x_o and d have finite support [3]. Moreover, even in that case, guaranteeing no overload may require a very large loading factor α, thus degrading performance for a given number of quantization levels. In our experience, the 4σ rule works very well in practice but the reader is warned that no rigorous guarantees can be provided if quantizer overload occurs.

25.6.3 Noiseless Digital Channels with Constrained Average Data Rate

In a noiseless digital channel with constrained average data rate,

$$\bar{w}(k) = \bar{v}(k) \tag{25.41}$$

whenever $\bar{v}(k)$ is a binary symbol, and the average expected length of these symbols (in bits per sample), say $\mathcal{R}$, is upper bounded by $\hat{\mathcal{R}}$, that is,

$$\mathcal{R} \triangleq \lim_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k-1} R(i) \leq \hat{\mathcal{R}}, \tag{25.42}$$

where $R(i)$ is the expected length (in bits) of the channel input $\bar{v}(i)$.

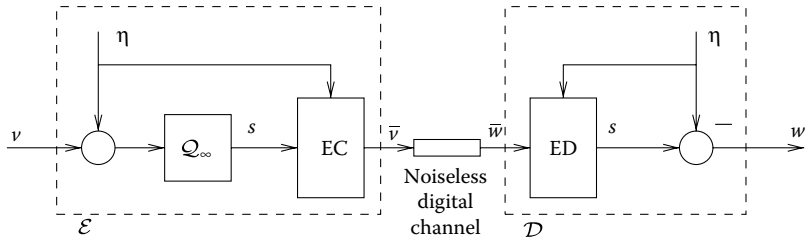

FIGURE 25.13 Entropy coded dithered quantizer used to transmit data over noiseless digital channel.

For these channels, we propose to use an entropy coded dithered quantizer (ECDQ) [34] as the $\mathcal{E}-\mathcal{D}$ pair in Figure 25.4. The structure of an ECDQ is shown in Figure 25.13, where $\mathcal{Q}_\infty$ is an infinite uniform quantizer with quantization step Δ, and the dither η is as before. The new blocks, namely EC and ED, form a lossless encoder–decoder pair. (Also called an entropy-encoder entropy-decoder pair [20, Chapter 5]). At each time instant, the quantizer output s is used by the EC to construct the binary word $\bar{v}(k)$ via*

$$\bar{v}(k) = \mathcal{H}_k\left(s(k), \eta(k)\right), \tag{25.43}$$

where $\mathcal{H}_k$ is a time-varying deterministic mapping from the countable quantizer output alphabet to a set of prefix-free binary words [20]. The output of the EC is then losslessly transmitted and, at the receiving end, the ED recovers s via

$$s(k) = \mathcal{H}_k^{-1}\left(\bar{w}(k), \eta(k)\right), \quad \bar{w}(k) = \bar{v}(k). \tag{25.44}$$

Finally, the dither is subtracted from s to construct w. In Equation 25.44, $\mathcal{H}_k^{-1}$ is a time varying mapping that satisfies $\mathcal{H}_k^{-1}(\mathcal{H}_k(s(k), \eta(k))) = s(k)$ at all time instants. The latter condition implies that the EC–ED pairs considered here operate in real time, without delay.

The maps $\mathcal{H}_k$ and $\mathcal{H}_k^{-1}$ depend on the conditional distribution of $s(k)$, given $\eta(k)$. Since the EC–ED pair is lossless, Theorem 25.5 guarantees that, under mild assumptions, $q \triangleq w - v$ is an i.i.d. sequence, uniformly distributed in $\left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$ and independent of $[x_o^T \;\; d^T]^T$. This is valid for any lossless EC–ED pair. Moreover, this also implies that the distribution of $s(k)$ given $\eta(k)$ can be calculated independently of the choice of EC and ED and, thus, these devices can be actually designed (even in closed loop; [33,34]).

A key property of ECDQs is stated next:

Theorem 25.6:

Consider the feedback scheme of Figure 25.4 where w and v are related by the ECDQ described above (Figure 25.13). If perfect communication (i.e., $w = v$) renders the feedback scheme internally stable and well-posed, Assumption 25.1 holds, and the dither is chosen as in Theorem 25.5, then there exists an EC–ED pair such that the average data-rate $\mathcal{R}$ satisfies

$$\mathcal{R} < \frac{1}{2}\ln\left(1+\gamma\right) + \frac{1}{2}\log_2\left(\frac{2\pi e}{12}\right) + 1, \quad \gamma \triangleq \frac{\sigma_v^2}{\Delta^2/12}, \tag{25.45}$$

where σ_v^2 is the stationary variance of the ECDQ input v.

---

* Here, we focus on memoryless EC–ED pairs, that is, only the actual signal and dither samples are used by the EC. A more general treatment can be found in [8].

Remark 25.3

The last two terms in Equation 25.45 arise due to the fact that ECDQs generate a quantization noise q that is uniformly distributed and not Gaussian, and because practical EC–ED pairs are not perfectly efficient [20] (see details in [8]).

By virtue of Theorems 25.5 and 25.6, it follows that when an ECDQ is employed to deal with noiseless digital channels, the link between v and w in Figure 25.4 behaves like an SNR constrained AWN channel, where

$$\gamma = \frac{\sigma_v^2}{\sigma_q^2} \leq 2^{2(\hat{\mathcal{R}} - \frac{1}{2}\log_2\left(\frac{2\pi e}{12}\right) - 1)} - 1 = \Gamma, \quad \sigma_q^2 = \frac{\Delta^2}{12}. \tag{25.46}$$

As in the previous case, choosing the equivalent noise variance amounts to choosing the quantization step Δ. On the other hand, the upper bound on γ in Equation 25.46 ensures that the average data-rate across the channel satisfies the constraint in Equation 25.42.

Remark 25.4

The inequality in Equation 25.45 is in general not tight. Thus, even though choosing Γ as above guarantees $\mathcal{R} < \hat{\mathcal{R}}$, it may yield conservative results.

As opposed to the case of digital channels with finite alphabets, the use of an ECDQ makes the link between v and w in Figure 25.4 a true SNR constrained AWN channel. Thus, all results of Section 25.5 rigorously apply to the channels considered in this section. It is worth noting that, for the same reasons as given above for the finite alphabet digital channel, assuming that feedback is available around the channel is natural in this case.

As an illustration of the use of our results, we will consider the problem of mean square stabilization by means of an architecture with feedback. From Theorems 25.1 and 25.6 we conclude that it is possible to achieve MSS in the NCS of Figure 25.4 at average data rates that satisfy

$$\mathcal{R} < \sum_{i=1}^{n_p} \log_2 |p_i| + \frac{1}{2}\log_2\left(\frac{2\pi e}{12}\right) + 1. \tag{25.47}$$

That is, the simple architecture proposed here has the ability to stabilize a given plant at average data-rates that are no more than $\frac{1}{2}\log_2\left(\frac{2\pi e}{12}\right) + 1$ ($\approx$1.254) bits/sample away from the absolute minimal average data-rate for stability identified in [24], namely $\sum_{i=1}^{n_p} \log_2 |p_i|$. This rate loss is, in our view, compensated by the simplicity of the SNR formulation advocated here. These findings are part of a more elaborate framework to deal with control system design subject to average data-rate constraints, as presented in [8,33].

25.6.4 Bernoulli Erasure Channels

In a Bernoulli erasure channel the input $\bar{v}$ and output $\bar{w}$ are related via

$$\bar{w}(k) = \theta(k)\bar{v}(k), \tag{25.48}$$

where θ is a sequence of i.i.d. binary random variables, being independent of $[x_o^T \;\; d^T]^T$, and such that $\mathcal{P}\{\theta(k) = 1\} = p < 1$.

For Bernoulli erasure channels we make $\bar{v} = v$, $w = \bar{w}$, and redraw Figure 25.4 as shown in Figure 25.14a, where H is an LT system depending on G, N, and M. We also consider an auxiliary situation where the erasure channel of Figure 25.14a has been replaced by a scalar additive noise channel plus a gain equal to the successful transmission probability p (Figure 25.14b). If T_{dv_p} (resp. T_{qv_p}) denotes the closed loop transfer function from d (resp. q) to the auxiliary signal v_p in the feedback system of Figure 25.14b, then we have the following result [33]:

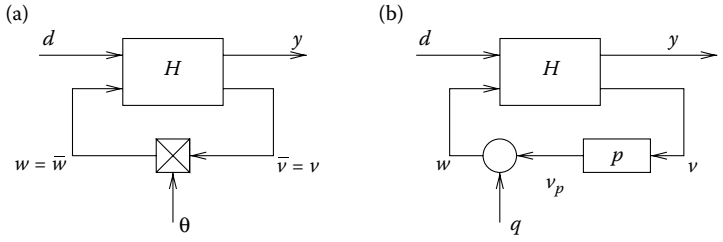

FIGURE 25.14 LTI system N with feedback over (a) erasure channel with dropout probability $1-p$, and (b) additive noise channel with gain p.

Theorem 25.7:

Consider the NCS of Figure 25.4, suppose that Assumption 25.1 holds, and assume that the channel is described by Equation 25.48 with θ *as above. Assume that* q *is a zero mean white noise sequence, uncorrelated with* $[x_o^T \;\; d^T]^T$. *Then, the NCS of Figure 25.4 is MSS if and only if the LTI loop of Figure 25.14b is internally stable and*

$$\sigma_q^2 = \frac{\left|\left|T_{dv_p}\Omega_d\right|\right|_2^2}{\frac{p}{1-p} - \left|\left|T_{qv_p}\right|\right|_2^2} \tag{25.49}$$

is nonnegative and finite. Moreover, if the NCS of Figure 25.4 is MSS, then the stationary spectral densities of the signals in the switched system of Figure 25.14a are equal to those of the corresponding signals in the LTI system of Figure 25.14b.

We conclude from Theorem 25.7 that stationary second-order and MSS-related properties of the considered NCS (equivalently, of the switching system of Figure 25.14a) can be studied by means of the simpler LTI system in Figure 25.14b, where the unreliable channel has been replaced by a gain p followed by an AWN channel with input v_p, output w, and subject to the stationary equality SNR constraint

$$\gamma_p \triangleq \frac{\sigma_{v_p}^2}{\sigma_q^2} = \frac{p}{1-p} \triangleq \Gamma_p. \tag{25.50}$$

(In Equation 25.50 we have used Equation 25.49 and the expression for the stationary variance v_p obtained from Figure 25.14b.)

In contrast to the channels studied so far, in the present case we arrive at an equality constraint on the equivalent channel SNR (and not an upper bound on it). This fact is not an issue since, as mentioned in Section 25.5, in most optimization problems subject to inequality SNR constraints, the SNR constraints are active at the optimum. Thus, all results in Section 25.5 can be applied to this case, with the obvious changes to account for the gain p. This implies that some simple modifications in the definition of P in Figure 25.8, and a slight change in the architecture for perfect reconstruction coding schemes, should be made. One possibility is shown in Figure 25.15.

For Bernoulli erasure channels, as was the case of the AWN channel studied previously, the equivalence between communication constraints and SNR constraints is exact. In other words, the SNR constraint corresponds to an alternative way of looking at the original communication constraint and not, as in the noiseless digital channel cases, a consequence of certain specific choices for the blocks $\mathcal{E}$ and $\mathcal{D}$. (In the present situation, however, varying σ_q^2 has no clear physical meaning, whilst having $\gamma_p < \Gamma_p$ makes no sense.)

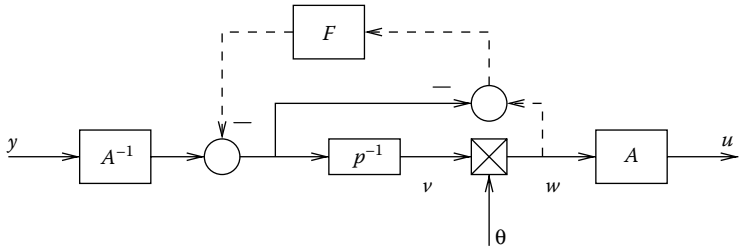

FIGURE 25.15 Perfect reconstruction coding scheme modified to account for the equivalent gain p in NCSs closed over Bernoulli erasure channels.

In the present case, the existence of feedback around the channel is equivalent to the existence of packet acknowledgements in the underlying communication protocol (TCP protocols [4]). If no packet acknowledgements are available (UDP protocols), then the assumption of channel feedback is invalid.

We conclude this section by showing how our approach allows one to rederive known results. We assume that an architecture with feedback is employed. Theorems 25.1 and 25.7 (and some further manipulations) allow one to conclude that it is possible to guarantee MSS of the NCS of Figure 25.4 if and only if the successful transmission probability p satisfies

$$p > 1 - \frac{1}{\prod_{i=1}^{n_p} |p_i|^2}. \tag{25.51}$$

The bound on p in Equation 25.51 is identical to the bound provided in [4]. However, our results hold when LTI architectures are employed, whereas the results in [4] consider time varying control schemes. We thus see that, for any Bernoulli erasure channel and a TCP communication protocol, the class of SISO plant models for which the time-varying schemes of [4] achieve MSS, is the same class for which our proposal achieves MSS. A thorough discussion of this result, and of the corresponding results for the case where no packet acknowledgements are available, can be found in [33].

25.7 Application of the SNR Approach to WCDMA Power Control

We now return to the WCDMA power control problem used as motivation in Section 25.2. The control loop of Figure 25.2 contains only one degree of design freedom, namely the choice of K. However, as implied by the results presented in Sections 25.3 through 25.6, optimal performance in the face of communication constraints can only be achieved by using additional degrees of freedom. We show here that this is indeed the case for the WCDMA power control loop.

We consider four control architectures:

- *Control Law 1:* This architecture corresponds to a one-block architecture without feedback. (If $N = [-1\ 0]$ and the quantizer parameters are *fixed* at $V = 1$ and $\Delta = 1$, then this architecture reduces to the typical control scheme used in practice [16].)
- *Control Law 2:* This architecture corresponds to a two-block architecture without feedback. For the design of this architecture we will perform two iterations of the first algorithm outlined in Section 25.5.4.
- *Control Law 3:* Here, we use a one-block architecture with feedback.
- *Control Law 4:* This control law corresponds to a two-block architecture with feedback. As with Control Law 2, we will perform two iterations of the first algorithm outlined in Section 25.5.4.

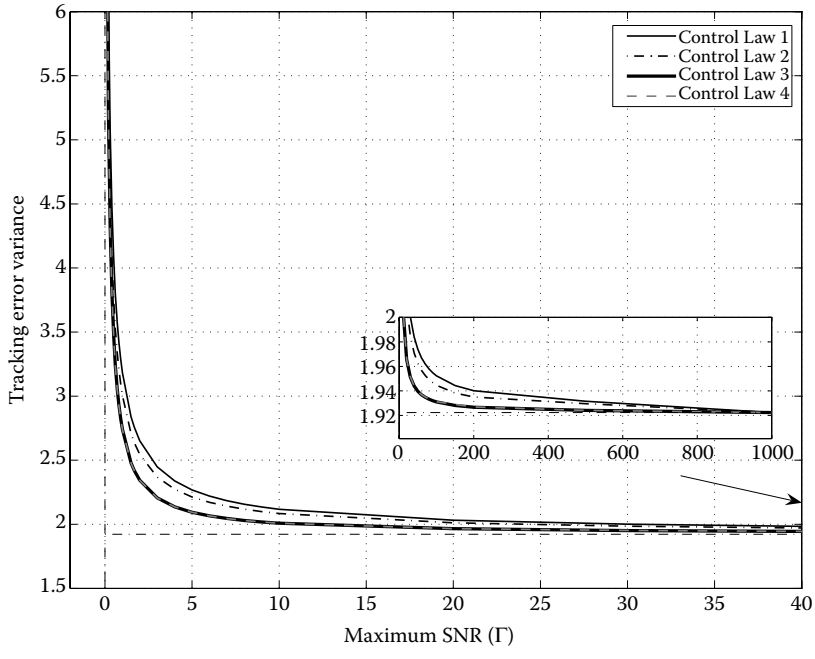

FIGURE 25.16 Achieved tracking error variance as a function of the maximum SNR Γ.

Note that, in practice, some of these architectures cannot be used. This is because of legacy constraints arising from existing implementations. However, it is interesting to see what benefits, if any, would be possible if one were able to redesign the system without regard for these legacy issues.

We let the loop delay $\beta = 2$, that is, we assume*

$$G = \frac{1}{z(z-1)}, \tag{25.52}$$

and note that the fact that the plant G is marginally stable implies that $\gamma_\infty = \gamma_{\infty,\mathrm{nf}} = 0$. The assumed model for the disturbance $d \triangleq g - I$ is $d = H_o n$, where

$$H_o = \frac{z}{z - 0.96} \tag{25.53}$$

and n is a zero mean white noise sequence of variance σ_n^2. The SIR reference value r is taken to be zero without loss of generality.

We begin by considering the linear NCS of Figure 25.5 and study the performance achieved by the above control laws for several values of the maximum available SNR Γ, assuming $\sigma_n^2 = 1$. The results are shown in Figure 25.16.

For each architecture, the achievable performance becomes increasingly worse as $\Gamma \to \gamma_\infty = 0$, and the best nonnetworked performance is recovered when $\Gamma \to \infty$ (see horizontal dash-dotted line). This behavior is as expected. Interestingly, the performance gains arising from the use of Control Laws 2 and 3 are only significant for small SNRs (say $\Gamma < 20$). It is also seen that, unsurprisingly, the two-block architecture without feedback achieves better performance than the corresponding one-block architecture. However, the two-block architecture with feedback brings no benefits when compared with the (simpler

* For design purposes, we modified the location of the unstable pole of G from $z = 1$ to $z = 1.0001$, so as to avoid convergence problems when solving the quadratic optimization problems associated with the designs.

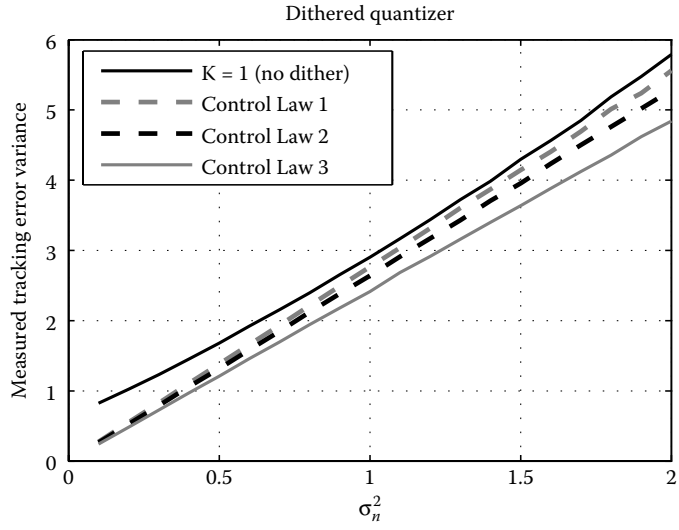

FIGURE 25.17 Measured tracking error variance as a function of the disturbance variance when a dithered quantizer is used.

to design and to implement) one-block architecture with feedback. Indeed, we notice that, having optimally chosen N (considering $M = 1$) at a first design stage, the optimal block M obtained at the second design stage was almost equal to unity for all values of Γ considered.*

We now consider a situation very close to the practical WCDMA setup, where the channel between BS and MS is a digital channel with a 3 symbol alphabet.† We thus consider (both dithered and undithered) quantizers with $L = 3$ levels. For design purposes, we use the SNR constrained additive noise model considering loading factors $\alpha_{\text{dither}} = \sqrt{10}$ for the dithered quantizer case, and $\alpha_{\text{no-dither}} = 4$ or the un-dithered quantizer case. These choices make the SNRs equal in both situations, ensuring a fair comparison between them.

We consider four architectures. The first one corresponds to the standard choice in practice, that is, $N = [-1\ \ 0]$, $M = 1$, and an undithered quantizer with *fixed* parameters $\Delta = 1$ and $V = 1$ [16]. (The choice for N and M amounts to choosing $K = 1$ in Figure 25.2.) The remaining three architectures correspond to Control Laws 1–3. We do not consider Control Law 4 because it provides performance levels that were identical to those achieved by the one-block architecture with feedback (recall the results on Figure 25.16).

Figures 25.17 and 25.18 show the measured variance of the tracking error as a function of the disturbance variance σ_n^2 for the four architectures, when dithered and undithered quantizers are employed. The simulation results correspond to averages over 20 realizations, each 10^5 samples long and, in the case of Control Laws $1-3$, an actual finite 3-level quantizer with parameters $\Delta = \sqrt{12\sigma_{q,\text{opt}}^2}$ and $V = \Delta(L-1)/2$ (here, $\sigma_{q,\text{opt}}^2$ is the optimal quantization noise variance) is employed.

In the dithered quantizer case, the proposed architectures outperform the standard one for all values of σ_n^2. The one-block architecture with feedback provides better performance for all cases, except when σ_n^2 is small where all proposed architectures perform similarly. It is worth noting that, for dithered quantizers, simulation results are indistinguishable from the predictors made by analytical calculations. However, in the undithered quantizer case, the situation is different since the additive noise model is no longer

* This result is consistent with the results for the three degree-of-freedom architecture proposed in [35] (which is a rewriting of a two-block architecture with feedback).

† We note, however, that in reality only two symbols are allowed.

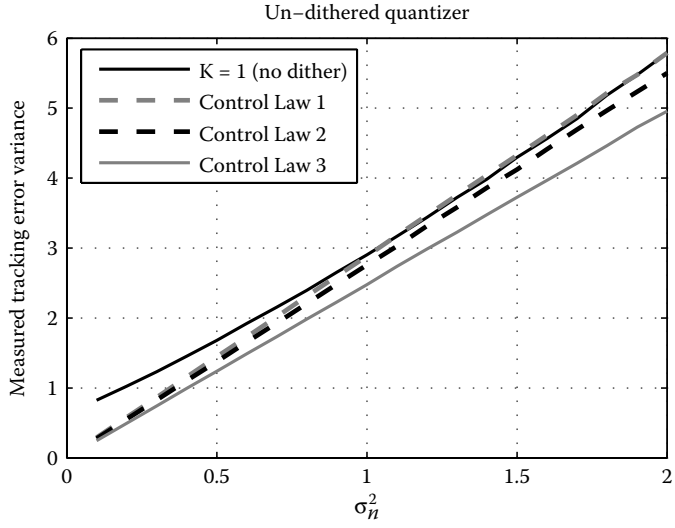

FIGURE 25.18 Measured tracking error variance as a function of the disturbance variance when an undithered quantizer is used.

(strictly) valid. It is seen in Figure 25.18 that the performance of Control Laws 1 and 2 are worse than in the dithered quantizer case, whilst the performance achieved by Control Law 3 does not present noticeable changes. In our opinion, the fact that Control Law 3 has feedback around the quantizer helps compensate for nonlinear effects, thus making the noise model more appropriate than in the other cases. Despite the above, it is seen that Control Laws 2 and 3 outperform the standard control law for all values of σ_n^2. Moreover, Control Law 1 outperforms the standard control law for $\sigma_n^2 \leq 1$.

25.8 Conclusions

This chapter has reviewed the SNR approach to NCS design. This approach is relatively simple and can be understood with little more than standard linear time-invariant systems theory. The methodology recovers many of the results currently available in the networked control literature, and applies to many problems of practical interest including control problems involving:

- AWN channels
- Noiseless digital channels with finite alphabet
- Noiseless digital channels with constrained average data-rate
- Bernoulli erasure channels

Acknowledgment

The first author, Eduardo I. Silva, acknowledges the support from CONICYT through grant FONDECYT 3100024.

References

1. G. C. Goodwin, S. Graebe, and M. E. Salgado. *Control System Design*. Prentice-Hall, Englewood Cliffs, New Jersey, 2001.
2. D. Hristu-Varsakelis and W. Levine (Eds.). *Handbook of Networked and Embedded Systems* Birkhäuser, Boston, Massachusets, 2005.
3. G. Nair, F. Fagnani, S. Zampieri, and R. Evans. Feedback control under data rate constraints: An overview. *Proceedings of the IEEE*, 95(1):108–137, 2007.
4. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1):163–187, 2007.
5. J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.
6. Y. Tipsuwan and M. Y. Chow. Control methodologies in networked control. *Control Engineering Practice*, 11:1099–1111, 2003.
7. P. Antsaklis and J. Baillieul. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1):5–8, 2007.
8. E. I. Silva, M. S. Derpich, and J. Østergaard. A framework for control system design subject to average data-rate constraints. Submitted to *IEEE Transactions on Automatic Control*, 2010.
9. E. I. Silva, M. S. Derpich, J. Østergaard, and D. E. Quevedo. Simple coding for achieving mean square stability over bit-rate limited channels. In *Proceedings of the 46th IEEE Conference on Decision and Control*, Cancún, México, 2008.
10. E. I. Silva, G. C. Goodwin, and D. E. Quevedo. On the design of control systems over unreliable channels. In *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.
11. J. H. Braslavsky, R. H. Middleton, and J. S. Freudenberg. Feedback stabilization over signal-to-noise ratio constrained channels. *IEEE Transactions on Automatic Control*, 52(8):1391–1403, 2007.
12. E. I. Silva, G. C. Goodwin, and D. E. Quevedo. Control system design subject to SNR constraints. *Automatica*, 46(2):428–436, 2010.
13. H. Holma and A. Toskala. *WCDMA for UMTS: HSPA Evolution and LTE*, 4th ed. Wiley, New York, 2007.
14. A. J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley, Reading, MA, 1995.
15. 3GPP TS 25.2111. *Physical Channels and Mapping of Transport Channels onto Physical Channels (FDD)*, Release 1999.
16. F. Gunnarsson. Power control in wireless networks: Characteristics and fundamentals. In M. Guizani, editor, *Wireless Communications Systems and Networks*, Chapter 7, pp. 179–208. Kluwer Academic Publishers, Dordrechet 2004.
17. N. Jayant and P. Noll. *Digital Coding of Waveforms. Principles and Approaches to Speech and Video*. Prentice-Hall, Englewood Cliffs, NJ, 1984.
18. M. S. Derpich, E. I. Silva, D. E. Quevedo, and G. C. Goodwin. On optimal perfect reconstruction feedback quantizers. *IEEE Transactions on Signal Processing*, 56(8):3871–3890, 2008.
19. G. C. Goodwin, D. E. Quevedo, and E. I. Silva. Architectures and coder design for networked control systems. *Automatica*, 44(1):248–257, 2008.
20. T. M. Cover and J. A. Thomas. *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., New York, 2006.
21. A. Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, 2005.
22. M. Fu and L. Xie. The sector bound approach to quantized feedback control. *IEEE Trans. Autom. Control*, 50(11):1698–1711, 2005.
23. D. Nesic and D. Liberzon. A unified framework for design and analysis of networked and quantized control systems. *IEEE Transactions on Automatic Control*, 54(4):732–747, 2009.
24. G. Nair and R. Evans. Stabilizability of stochastic linear systems with finite feedback data rates. *SIAM Journal on Control and Optimization*, 43(2):413–436, 2004.
25. S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. IEEE Transactions on Automatic Control, 49(9):1549–1561, 2004.
26. K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1996.
27. K. J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, New York, 1970.
28. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, Pennsylvania, 1994.

29. M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, 2006.
30. B. Widrow and I. Kollár. *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge University Press, New York, NY, 2008.
31. W.R. Bennet. Spectra of quantized signals. *Bell Syst. Tech. J.*, 27(4):446–472, 1948.
32. R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6), 1998.
33. E. I. Silva. *A Unified Framework for the Analysis and Design of Networked Control Systems*. PhD thesis, School of Electrical Eng. and Comp. Sci., The University of Newcastle, Australia, 2009.
34. R. Zamir and M. Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 38(2):428–436, 1992.
35. J. C. Agüero, G. C. Goodwin, K. Lau, M. Wang, E. I. Silva, and T. Wigren. Three-degree of freedom adaptive power control for CDMA cellular systems. In *IEEE Global Communications Conference (Globecom)*, Honolulu, HI, 2009.

26

Optimization and Control of Communication Networks

Srinivas Shakkottai
Texas A&M University

Atilla Eryilmaz
The Ohio State University

26.1 Introduction

The Internet is perhaps the largest distributed control system ever built. It consists of millions of nodes each interacting with a subset of other nodes, sending and receiving packets by sharing available bandwidth in a fair and resource-efficient manner. The system is so robust that it is practically taken for granted, with service outages making headlines. In this chapter we present an analytical framework that ties ideas of distributed control with those of fair resource allocation in communication networks.

The fundamental guiding principle behind the Internet is that control decisions should be made on an end-to-end basis. In other words, there should be no centralized entity that makes resource allocation decisions. Figure 26.1 presents a simple schematic representation of the Internet. A *flow* between a Webserver and an end user consists of data packets from the Web server and acknowledgments from the end user. The packets traverse a *route* that consists of routers that decide which direction to forward packets. The ownership of the network itself is divided among several Internet service providers (ISPs).

In Figure 26.1, the Web server must decide the rate at which packets are sent, based on the state of the route that the packets traverse—if there are several competing flows, it must be aware of them and back off so as to not monopolize the bandwidth. It is aided in these decisions by signals at the routers, which could be in the form of dropping packets (if queues at the router overflow) or by marking them

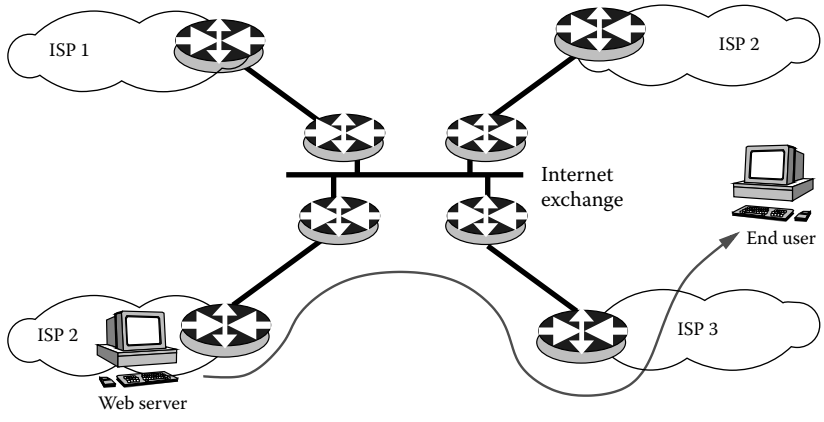

FIGURE 26.1 A schematic representation of an Internet flow. Control is end-to-end, and routers simply store-and-forward packets.

in some way. These signals propagate to the end user who propagates them back to the server using acknowledgment packets, with prearranged responses to lost or marked packets. Note that each router does not communicate directly with the source or destination, but merely stores-and-forwards packets as needed.

This chapter deals with the design and analysis of control systems for Internet flows. Our objective will be to design source laws at the server end that would respond to network feedback in such a way that fair resource allocation is achieved. Our focus will be primarily on deterministic analysis of the system, with a brief discussion of stochastic aspects at the end of the chapter. For a more comprehensive study of Internet control systems, the reader is referred to [1,2].

26.2 Network Utility Maximization

We first develop a utility maximization framework for our network resource allocation problem, following seminal work by Kelly et al. [3–6]. As we saw in Section 26.1, the Internet can be thought of consisting of a set of traffic sources $\mathcal{S}$ and a set of links $\mathcal{L}$. Each link $l \in \mathcal{L}$ has some finite capacity c_l. Each source would like to send traffic to some destination in the network, and uses a fixed route $r \subset \mathcal{L}$ to reach its destination. The utility that the source obtains from transmitting data on route r at rate x_r is denoted by $U_r(x_r)$. We assume that the utility function is continuously differentiable, nondecreasing, and strictly concave. Here, the concavity assumption follows from the ideas that the user experiences diminishing returns per unit capacity received (along all the links of its route). For example, a user would feel the effect of a rate increase from 1 to 100 kbps much more than an increase from 1 to 1.1 Mbps although the increase is the same in both cases. The network must somehow allocate capacity on the links such that the sum total user utility is maximized. In other words, the problem that the network faces is the following optimization problem:

$$\max_{x_r} \sum_{r \in \mathcal{S}} U_r(x_r) \tag{26.1}$$

subject to the constraints

$$\sum_{r:l \in r} x_r \leq c_l, \quad \forall l \in \mathcal{L}, \tag{26.2}$$

$$x_r \geq 0, \quad \forall r \in \mathcal{S}. \tag{26.3}$$

The constraints above indicate that the link capacities are finite, and that each user must receive a nonnegative transmission rate. The constraints form a convex set, and since the utility functions are strictly concave, the problem has a unique solution.

Example 1

The above ideas are illustrated using the following example that appears in [2]. Consider the two-link network shown in Figure 26.2.

The network consists of two links A and B and three sources, with routes as shown. Let the capacity of link A be $C_A = 2$ and that of link B be $C_B = 1$. The utility maximization problem is

$$\log x_0 + \log x_1 + \log x_2$$

subject to

$$x_0 + x_1 \leq 2,$$
$$x_0 + x_2 \leq 1,$$

and x_0, x_1 and x_2 are all nonnegative. Since $\log x \to -\infty$ as $x \to 0$, the optimal solution will allocate nonnegative rates. So we can drop the nonnegativity constraints. The Lagrange dual is given by

$$L(x, \lambda) = \log x_0 + \log x_1 + \log x_2 - \lambda_A(x_0 + x_1) - \lambda_B(x_0 + x_2),$$

where x is the vector of transmission rates allocated to the sources and λ is the vector of Lagrange multipliers. Then setting $\partial L/\partial x_r = 0$ for each r yields

$$x_0 = \frac{1}{\lambda_A + \lambda_B}, \quad x_1 = \frac{1}{\lambda_A}, \quad x_2 = \frac{1}{\lambda_B}.$$

Using $x_0 + x_1 = 2$ and $x_0 + x_2 = 1$ gives

$$\lambda_A = \frac{\sqrt{3}}{\sqrt{3}+1}, \quad \lambda_B = \sqrt{3}.$$

Hence, the solution is

$$\hat{x}_0 = \frac{\sqrt{3}+1}{3+2\sqrt{3}}, \quad \hat{x}_1 = \frac{\sqrt{3}+1}{\sqrt{3}}, \quad \hat{x}_2 = \frac{1}{\sqrt{3}}.$$

26.3 Fairness

The utility maximization approach taken in the previous section, aimed at ensuring that the sum total "satisfaction" among users is maximized. We will show in this section that it can also be viewed in terms of ensuring a certain predefined fairness condition among the users. In other words, the network can

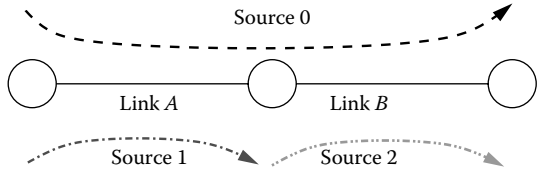

FIGURE 26.2 A simple two-link network.

assume certain utility functions on the users and solve the optimization problem of Section 26.2. We will see that each assumption on the utility function would lead to a different kind of fair resource allocation among users. Consider (strictly concave, increasing) utility functions of the form

$$U_r(x_r) = \frac{x_r^{1-\alpha}}{1-\alpha}, \tag{26.4}$$

for some $\alpha_r > 0$. Resource allocation using the above utility function is called α-*fair*, developed by Mo and Walrand [7,8]. Different values of α yield different types of fairness.

First consider $\alpha = 2$. This implies that $U_r(x_r) = -1/x_r$. Here, maximizing the sum of user utilities is equivalent to minimizing $\sum_r 1/x_r$. We can think of $1/x_r$ as the delay in associated with transferring a file of size 1. In other words, the objective is to minimize the total file transfer delay of all users, and we refer to this fairness metric as *minimum potential delay* fairness.

Now consider the case when $\alpha \to 1 \cdot \alpha = 1$. Maximizing the sum of $x_r^{1-\alpha}/(1-\alpha)$ yields the same optimum as maximizing the sum of

$$\frac{x_r^{1-\alpha} - 1}{1-\alpha}.$$

From L'Hospital's rule, we get

$$\lim_{\alpha \to 1} \frac{x_r^{1-\alpha} - 1}{1-\alpha} = \log x_r.$$

In other words, $U_r(x_r) = w_r \log x_r$ in this case. Using the fact that the log function is continuous and concave, we can see that the optimal allocation that solves the network utility maximization problem, $\{\hat{x}_r\}$, satisfies

$$\sum_r \frac{x_r - \hat{x}_r}{\hat{x}_r} \leq 0,$$

where $\{x_r\}$ is any other feasible allocation. In other words, if we increase the rate allocation to some user, his percentage increase in rate will be more than offset by percentage decrease in rate for some other user. Such an allocation is called *proportionally fair*. If the utilities are chosen such that $U_r(x_r) = w_r \log x_r$, where w_r is some weight, then the resulting allocation is said to be *weighted proportionally fair*.

Finally, consider the case when $\alpha \to \infty$. Let $\hat{x}_r(\alpha)$ be the α-fair allocation. Then, by concavity

$$\sum_r \frac{x_r - \hat{x}_r}{\hat{x}_r^{\alpha}} \leq 0.$$

Considering some flow s, the above expression can be rewritten as

$$\sum_{r:\hat{x}_r \leq \hat{x}_s} (x_r - \hat{x}_r)\frac{\hat{x}_s^{\alpha}}{\hat{x}_r^{\alpha}} + (x_s - \hat{x}_s) + \sum_{i:\hat{x}_i > \hat{x}_s} (x_i - \hat{x}_i)\frac{\hat{x}_s^{\alpha}}{\hat{x}_i^{\alpha}} \leq 0.$$

If α is large, the third term in the above expression to be negligible. Hence, if $x_s > \hat{x}_s$, then the allocation for at least one user whose rate is such that $\hat{x}_r \leq \hat{x}_s$ would decrease. This property that "there is no alternative allocation that gives more to one user without leaving a less fortunate user worse off than before" is called *max–min* fair.

26.4 Distributed Control and Stability

Thus far we have discussed a utility maximization approach to resource allocation in networks. However, such a centralized allocation approach is infeasible in the Internet due to the existence of many millions of simultaneous flows routed across multiple service providers. In order to allow for end-to-end control, we

need to develop a framework for distributed control in order to attain our optimization goals. The control system is visualized as consisting of sources $r \in \mathcal{S}$ that choose their respective rates of transmission x_r, and links $l \in \mathcal{L}$ that declare a *price* p_l based on the load that they experience. We will see what these prices translate to in the network context in the next few sections. We introduce a matrix R which is called the routing matrix of the network. The (l, r) element of this matrix is given by

$$R_{lr} = \begin{cases} 1 & \text{if route } r \text{ uses link } l \\ 0 & \text{else} \end{cases}$$

Define

$$y_l = \sum_{s:l \in s} x_s, \tag{26.5}$$

which is the load on link l. Using the elements of the routing matrix, y_l can also be written as

$$y_l = \sum_{s:l \in s} R_{ls} x_s.$$

Letting y be the vector of all y_l ($l \in \mathcal{L}$), we have

$$y = Rx \tag{26.6}$$

Let $p_l(t)$ denote the price of link l at time t, that is,

$$\begin{aligned} p_l(t) &= f_l\left(\sum_{s:l \in s} x_s\right) \\ &= f_l(y_l(t)), \end{aligned} \tag{26.7}$$

where $f_l(.)$ is an increasing, continuous price function that maps link load to the link price. The price of a route is the sum of link prices p_l of all the links in the route. Hence, we define the price of route r to be

$$q_r = \sum_{l:l \in r} p_l(t). \tag{26.8}$$

Also, let p be the vector of all link prices and q be the vector of all route prices. We thus have

$$q = R^T p \tag{26.9}$$

Expressions 26.6 and 26.9 provide linear relationships between the control at the sources and the control at the link. The above relationships are illustrated in Figure 26.3.

We will be studying the stability and convergence properties of different source and link control laws in the following sections. At this point we recall notions of stability of distributed systems that arise from Lyapunov theory. Consider a dynamical system represented by

$$\dot{x} = g(x), \quad x(0) = x_0, \tag{26.10}$$

where it is assumed that $g(x) = 0$ has a unique solution. Call this solution 0. Here x and 0 can be vectors.

Definition 26.1:

The equilibrium point 0 *of Equation 26.10 is said to be*

- *Stable if, for each* $\varepsilon > 0$, *there is a* $\delta = \delta(\varepsilon) > 0$, *such that*

$$\|x(t)\| \leq \varepsilon, \quad \forall t \geq 0, \quad \textit{if } \|x_0\| \leq \delta.$$

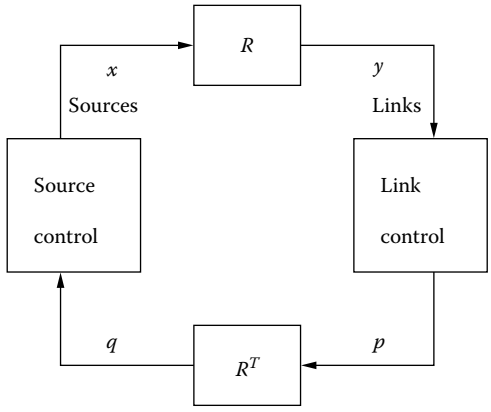

FIGURE 26.3 Network resource allocation. Sources respond to the prices fed back from the links.

- *Asymptotically stable if there exists a* $\delta > 0$ *such that*

$$\lim_{t\to\infty} \|x(t)\| = 0$$

  for all $\|x_0\| \leq \delta$.
- *Globally, asymptotically stable if*

$$\lim_{t\to\infty} \|x(t)\| = 0$$

  for all initial conditions x_0.

Lyapunov's theorem, which uses the Lyapunov function to test for the stability of a dynamical system, can be stated as follows [9]. ■

Theorem 26.1:

Let $x = 0$ *be an equilibrium point for* $\dot{x} = f(x)$ *and* $D \subset \mathbb{R}^n$ *be a domain containing* 0. *Let* $V : D \to \mathbb{R}$ *be a continuously differentiable function such that*

$$V(x) > 0, \quad \forall x \neq 0$$

and $V(0) = 0$. *Now we have the following conditions for the various notions of stability.*

1. *If* $\dot{V}(x) \leq 0\ \forall x$, *then the equilibrium point is stable.*
2. *In addition, if* $\dot{V}(x) < 0, \forall x \neq 0$, *then the equilibrium point is asymptotically stable.*
3. *In addition to (1) and (2) above, if* V *is radially unbounded, that is,*

$$V(x) \to \infty, \quad \textit{when } \|x\| \to \infty,$$

   then the equilibrium point is globally asymptotically stable. ■

Note that the above theorem also holds if the equilibrium point is some $\hat{x} \neq 0$. In this case, consider a system with state vector $y = x - \hat{x}$ and the results immediately apply.

26.5 Primal Algorithm for Distributed Utility Maximization

We relax the utility maximization problem in Equation 26.1 so as to allow simple algorithm design. We associate a penalty function for exceeding the capacity of each link and try to maximize the difference of utility minus penalty, that is, we define

$$V(x) = \sum_{r \in \mathcal{S}} U_r(x_r) - \sum_{l \in \mathcal{L}} B_l \left(\sum_{s:l \in s} x_s \right), \tag{26.11}$$

where x is the vector of rates of all sources. $B_l(.)$ is the penalty for exceeding the link capacity and is assumed to be convex, increasing and continuously differentiable. Equivalently,

$$B_l \left(\sum_{s:l \in s} x_s \right) = \int_0^{\sum_{s:l \in s} x_s} f_l(y)\, dy, \tag{26.12}$$

where $f_l(.)$ is an increasing, continuous function. We call $f_l(y)$ the price function, associated with link l that we saw in Section 26.4. Clearly, B_l defined in the above fashion is convex, and since U_r is strictly concave, $V(x)$ is strictly concave. We assume that U_r and f_l are such that the maximization of Equation 26.11 results in a solution with $x_r \geq 0\ \forall r \in \mathcal{S}$. The maximizer of Equation 26.11 is obtained by differentiation and is given by

$$U_r'(x_r) - \sum_{l:l \in r} f_l \left(\sum_{s:l \in s} x_s \right) = 0, \quad r \in \mathcal{S}. \tag{26.13}$$

We choose a simple gradient ascent algorithm in order to maximize the relaxed problem (Equation 26.11), first presented in [6]. Consider the algorithm

$$\dot{x}_r = k_r(x_r) \left(U_r'(x_r) - \sum_{l:l \in r} f_l \left(\sum_{s:l \in s} x_s \right) \right), \tag{26.14}$$

where $k_r(.)$ is nonnegative, increasing and continuous. The algorithm is obtained by simply setting $\dot{x}_r$ proportional to the gradient of Equation 26.11 in the dimension x_r. The stationary point of Equation 26.14 satisfies Equation 26.13 and hence maximizes (Equation 26.11). The algorithm follows the idea of decreasing rate when price is high, and increasing it otherwise.

We use Lyapunov theory to prove that it indeed converges to the stationary point. Now, $V(x)$ as defined in Equation 26.11 is a strictly concave function. Let $\hat{x}$ be its unique maximum. Then, $V(\hat{x}) - V(x)$ is nonnegative and is equal to zero only at $x = \hat{x}$. Thus, $V(\hat{x}) - V(x)$ is a possible candidate Lyapunov function for the system (Equation 26.14). We use this Lyapunov function in the theorem below.

Theorem 26.2:

Consider a network in which all sources follow the primal control algorithm (Equation 26.14*). Assume that the functions $U_r(.), k_r(.)$ and $f_l(.)$ are such that $W(x) = V(\hat{x}) - V(x)$ is such that $W(x) \to \infty$ as $||x|| \to \infty$, $\hat{x}_i > 0$ for all i, and $V(x)$ is as defined in Equation 26.11. Then, the controller in Equation 26.14 is globally asymptotically stable and the equilibrium value maximizes Equation 26.11.*

Proof. Differentiating $W(.)$, we get

$$\begin{aligned}\dot{W} &= -\sum_{r\in\mathcal{S}} \frac{\partial V}{\partial x_r}\dot{x}_r \\ &= -\sum_{r\in\mathcal{S}} k_r(x_r)(U_r'(x_r) - q_r)^2 < 0, \quad \forall x \neq \hat{x},\end{aligned} \tag{26.15}$$

and $\dot{W} = \forall x = \hat{x}$. Thus, all the conditions of the Lyapunov theorem are satisfied and we have proved that the system state converges to $\hat{x}$. ■

In the proof of the above theorem, we have assumed that utility, price and scaling functions are such that $W(x)$ has some desired properties. For example, if $U_r(x_r) = w_r \log(x_r)$, and $k_r(x_r) = x_r$, then the primal resource allocation algorithm for source r becomes

$$\dot{x}_r = w_r - x_r \sum_{l:l\in r} f_l(y_l),$$

and thus the unique equilibrium point is $w_r/x_r = \sum_{l:l\in r} f_l(y_l)$. If $f_l(.)$ is any polynomial function, then $V(x)$ goes to $-\infty$ as $||x|| \to \infty$ and thus, $W(x) \to \infty$ as $||x|| \to \infty$.

26.6 Dual Algorithm for Distributed Utility Maximization

We have just seen how to solve a relaxed version of the network utility maximization problem. We now consider a controller based on duality that naturally yields the exact solution. Consider the resource allocation problem (Equation 26.1). The Lagrange dual of the problem is

$$D(p) = \max_{\{x_r>0\}} \sum_r U_r(x_r) - \sum_l p_l \left(\sum_{s:l\in s} x_s - c_l \right), \tag{26.16}$$

where the p_ls are the Lagrange multipliers. Then the dual problem is simply

$$\min_{p\geq 0} D(p).$$

We again use a gradient algorithm to solve the above (descent in this case), developed in [6,10]. Now, in order to achieve the maximum in Equation 26.16, x_r must satisfy

$$U_r'(x_r) = q_r, \tag{26.17}$$

or equivalently,

$$x_r = U_r'^{-1}(q_r), \tag{26.18}$$

where, $q_r = \sum_{l:l\in r} p_r$, is the price of route r. Since

$$\frac{\partial D}{\partial p_l} = \sum_{r:l\in r} \frac{\partial D}{\partial q_r}\frac{\partial q_r}{\partial p_l},$$

we have from Equations 26.18 and 26.16 that

$$\frac{\partial D}{\partial p_l} = \sum_{r:l\in r} \frac{\partial U_r(x_r)}{\partial p_l} - (y_l - c_l) - \sum_i p_i \frac{\partial y_i}{\partial p_l}, \tag{26.19}$$

where the x_r above is the optimizing x_r in Equation 26.16.

In order to evaluate the above, we first compute $\partial x_r/\partial p_l$. Differentiating Equation 26.17 with respect to p_l yields

$$U_r''(x_r)\frac{dx_r}{dp_l} = 1$$
$$\Rightarrow \qquad \frac{\partial x_r}{\partial p_l} = \frac{1}{U_r''(x_r)}$$

Substituting in Equation 26.19 gives

$$\frac{\partial D}{\partial p_l} = \sum_{r:l\in r}\frac{U_r'(x_r)}{U_r''(x_r)} - (y_l - c_l) - \sum_i p_i \sum_{r:l\in r}\frac{1}{U_r''(x_r)} \tag{26.20}$$
$$= c_l - y_l, \tag{26.21}$$

where we have interchanged the last two summations in Equation 26.20 and used the facts $U_r'(x_r) = q_r$ and $q_r = \sum_{l\in r} p_l$. The above is the gradient of the Lagrange dual, and from Equations 26.18 and 26.21, we have the following dual control (gradient descent) algorithm:

$$x_r = U_r'^{-1}(q_r) \quad \text{and} \tag{26.22}$$
$$\dot{p}_l = h_l(y_l - c_l)_{p_l}^+, \tag{26.23}$$

where, $h_l > 0$ is a constant and $(g(x))_y^+$ denotes

$$(g(x))_y^+ = \begin{cases} g(x), & y > 0, \\ \max(g(x), 0), & y = 0, \end{cases}$$

The modification implies that p_l is nonnegative (which is valid, since by the Karush–Kuhn–Tucker (KKT) conditions the optimal p_l is nonnegative). If $h_l = 1$, the price update above has the same dynamics as the dynamics of the queue at link l. Thus, the queue length naturally provides price information.

We again use Lyapunov techniques in order to prove that the algorithm converges to the optimal solution; we present a version of the proof in [11]. Let the maximizer of Equation 26.1 be denoted by $\hat{x}$. Suppose that in the dual formulation (Equation 26.16) given q, there exists a unique p such that $q = R^T p$ (i.e., R has full row rank), where R is the routing matrix. At the optimal solution

$$\hat{q} = R^T \hat{p},$$

and the KKT conditions imply that, at each link l, either

$$\hat{y}_l = c_l$$

if the constraint is active or

$$\hat{y}_l < c_l \quad \text{and} \quad \hat{p}_l = 0$$

if the link is not a fully utilized. Note that under the full row rank assumption on R, $\hat{p}$ is also unique. Then we have the following theorem.

Theorem 26.3:

Under the assumption that given q, there exists a unique p such that $q = R^T p$, the dual algorithm is globally asymptotically stable.

Proof. Consider the Lyapunov function

$$V(p) = \sum_{l \in \mathcal{L}} (c_l - \hat{y}_l) p_l + \sum_{r \in \mathcal{S}} \int_{\hat{q}_r}^{q_r} (\hat{x}_r - (U_r')^{-1}(\sigma))\, d\sigma.$$

Then we have

$$\begin{aligned}
\frac{dV}{dt} &= \sum_l (c_l - \hat{y}_l)\dot{p}_l + \sum_r (\hat{x}_r - (U_r')^{-1}(q_r))\dot{q}_r \\
&= (c - \hat{y})^T \dot{p} + (\hat{x} - x)^T \dot{q} \\
&= (c - \hat{y})^T \dot{p} + (\hat{x} - x)^T R^T \dot{p} \\
&= (c - \hat{y})^T \dot{p} + (\hat{y} - y)^T \dot{p} \\
&= (c - y)^T \dot{p} \\
&= \sum_l h_l (c_l - y_l)(y_l - c_l)^+_{p_l} \\
&\leq 0.
\end{aligned}$$

Also, $\dot{V} = 0$ only when each link satisfies either $y_l = c_l$ or $y_l < c_l$ with $p_l = 0$. Finally, since $U_r'(x_r) = q_r$ at each time instant, all the KKT conditions are satisfied. The system converges to the unique optimal solution of Equation 26.1. ■

26.7 Cross-Layer Design for Wireless Networks

So far, we have focused on the network utility maximization problem in wired networks, where the goal has been to develop rate control algorithms for flows with fixed routes that converge to the optimal flow rate allocation. In this section, we show that the approach of using optimization and control theories can be significantly expanded to cover wireless networks and to add routing and medium access (also called scheduling) decisions to the rate controller. Such joint algorithms are called "cross-layer" since the decisions traverse the transport, networking, and medium access control (MAC) layers of the standard network hierarchy.

Contrary to wired networks where each link $l \in \mathcal{L}$ had a fixed capacity c_l associated with it, in wireless networks the capacities of every link depends on the transmission activity in neighboring nodes. Thus, we must extend our network model to incorporate such interference effects. Since we are also interested in finding optimal routes, we need to modify the definition of a session to release the fixed route assumption. The corresponding model is provided next.

As before, we let $\mathcal{N}$ be the set of nodes and $\mathcal{L}$ be the set of permissible hops. Let $\mathcal{F}$ be the set of flows in the system. Since we do not associate flows with fixed routes anymore, we define each flow f by its beginning node $b(f)$ and its ending node $e(f)$. Our resource allocation algorithm will find the optimal (potentially multipath) route for each flow. An example network is depicted in Figure 26.4.

We assume that each node, say i, maintains a separate queue, for every other node, say d, in the network so that every packet that enters i and that is destined to d is maintained in the corresponding queue. Let r_{ij} be the rate at which transmission takes place from node i to node j. In a wireless setting, the rates between the various nodes are interrelated due to interference effects. We consider a simple model to describe this interference although the results can be generalized significantly [12–16]. Let $\{A_m\}$ $m = 1, 2, \ldots, M$ be a collection of subsets of $\mathcal{L}$, where A_m is a set of hops that can be scheduled simultaneously under the interference constraints. Each A_m is called a *feasible schedule* and M is the number of possible feasible schedules. If schedule A_m is used, then $r_{ij} = 1$ if $(i, j) \in A_m$, and $r_{ij} = 0$ otherwise. Thus, all the scheduled links can transmit at rate 1 while the unscheduled hops remain silent. The network has a choice of

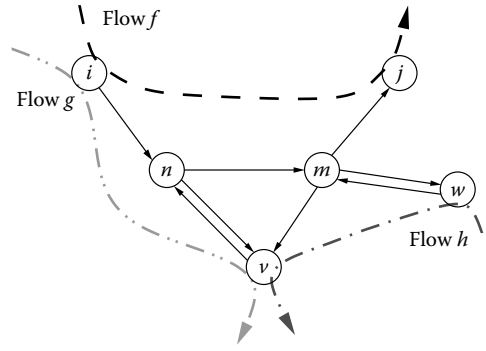

FIGURE 26.4 An example network model with $b(f) = i$, $e(f) = j$, $b(g) = i$, $e(g) = v$, and $b(h) = w$, $e(h) = v$.

implementing any one of the feasible schedules at each time instant. Suppose that π_m is the fraction of time that the network chooses to use schedule m. Then, the average rate allocated to hop (i,j) is given by

$$R_{ij} = \sum_{m:(i\,j)\in A_m} \pi_m.$$

In order to guarantee stability of the queues, we need to impose a constraint that the average influx of traffic into each queue must be no more than its outflux (Equation 26.24). To facilitate such formulation, we denote the total exogenously generated flow rate that enters node i to be conveyed to node d as

$$x_i^d := \sum_{\{f:b(f)=i,e(f)=d\}} x_f,$$

and denote the inflow rate allocated for destination d at node i by $R^d_{in(i)}$ and the outflow rate $R^d_{out(i)}$. Thus, at each node i,

$$\sum_d R^d_{in(i)} := \sum_j R_{ji} \quad \text{and} \quad \sum_d R^d_{out(i)} := \sum_k R_{ik}.$$

For simplicity, we will only consider the case where $U_f(x_f) = w_f \log x_f$, although the approach applies to a large class of utility functions (e.g. [13,17,18]). Then, the optimization problem for the resource allocation problem is given by

$$\max_{x,\pi,R\geq 0} \sum_f w_f \log x_f,$$

subject to the following constraints:

$$R^d_{in(i)} + x_i^d \leq R^d_{out(i)}, \quad \forall \text{ nodes } i,\ d \neq i \tag{26.24}$$

$$R^d_{in(i)} = \sum_j R^d_{ji}, \quad \forall i,j,d \neq i \tag{26.25}$$

$$R^d_{out(i)} = \sum_j R^d_{ij}, \quad \forall i,j,d \neq i \tag{26.26}$$

$$\sum_d R^d_{ij} = R_{ij}, \quad \forall i,j,d \neq i \tag{26.27}$$

$$R_{ij} = \sum_{m:(i,j)\in A_m} \pi_m, \quad \forall i,j \tag{26.28}$$

$$\sum_{m=1}^{M} \pi_m = 1. \tag{26.29}$$

Let the Lagrange multiplier corresponding to constraint (Equation 26.24) be denoted by p_{id}. Appending the constraints (Equation 26.24) to the objective, we get

$$\max_{x,\pi,R\geq 0} \sum_f w_f \log x_f - \sum_i \sum_{d\neq i} p_{id} \left(R^d_{in(i)} + x^d_i - R^d_{out(i)} \right).$$

By manipulating the summations, the above expression can be rewritten as

$$\max_{x\geq 0} \left(\sum_f w_f \log x_f - \sum_f p_{ib(f)} x_f \right) + \max_{\pi,R\geq 0} \left(\sum_i \sum_{d\neq i} p_{id} (R^d_{out(i)} - R^d_{in(i)}) \right).$$

If the optimal values of the Lagrange multipliers are known, then we have to solve two problems: the congestion control problem

$$\max_{x\geq 0} \left(\sum_f w_f \log x_f - \sum_f p_{ib(f)} x_f \right),$$

and the scheduling problem

$$\max_{\pi,R\geq 0} \left(\sum_i \sum_{d\neq i} p_{id} (R^d_{out(i)} - R^d_{in(i)}) \right), \tag{26.30}$$

where the scheduling problem is subject to the constraints Equations 26.25 through 26.29. To solve the congestion control problem, we use the primal algorithm at each source f:

$$\dot{x}_f(t) = \left(\frac{w_f}{x_f(t)} - p_{b(f)e(f)}(t) \right)^+_{x_f(t)}, \tag{26.31}$$

and the dual algorithm for price update at each node i, for each destination d:

$$\dot{p}_{id}(t) = \left(x^d_i(t) + R^d_{in(i)}(t) - R^d_{out(i)}(t) \right)^+_{p_{nd}(t)}. \tag{26.32}$$

In the price-update algorithm, the rates $R^d_{in(i)}$ and $R^d_{out(i)}$ are calculated at each time instant by solving Equation 26.30. If the optimal solution to Equation 26.30 is not unique, any one of the optima can be used.

To show convergence of the algorithm, we will use a theorem similar to the Lyapunov theorem, namely LaSalle's invariance principle, that is designed for situations where the time-derivative of a Lyapunov function is zero at more than one point. Consider the differential equation $\dot{y} = f(y(t))$. Then we have the following theorem [9]:

Theorem 26.4: LaSalle's Invariance Principle

Let $W : D \to R$ be a radially unbounded (i.e., $\lim_{||y||\to\infty} W(y) = \infty$, $y \in D$), continuously differentiable, positive definite function such that $\dot{W}(y) \leq 0$ for all $y \in D$. Let $\mathcal{E}$ be the set of points in D where $\dot{W}(y) = 0$. Let $\mathcal{M}$ be the largest invariant set in $\mathcal{E}$ (i.e., $\mathcal{M} \subseteq \mathcal{E}$ and if $y(0) \in \mathcal{M}$, then $y(t) \in \mathcal{M}$ $\forall\, t \geq 0$). Then, every solution starting in D approaches $\mathcal{M}$ as $t \to \infty$.

In order to study the convergence of the controller, we will use the following Lyapunov function:

$$W(x,q) \triangleq \frac{1}{2}\sum_{f\in\mathcal{F}}(x_f-\hat{x}_f)^2+\frac{1}{2}\sum_i\sum_{d\neq i}(p_{id}-\hat{p}_{nd})^2, \tag{26.33}$$

where the quantities with hats are any one of the solutions to the network utility maximization problem. In our subsequent analysis, we will focus on the case where the solution is unique, which allows us to talk about convergence to single point. All the arguments can be applied to the more general case when the system converges to the *set* of optima. From now on, in double summations involving a node i and destination d, for ease of notation, we will assume that $d \neq i$ without explicitly stating it. We are now ready to prove the following theorem:

Theorem 26.5:

Starting from any $x(0)$ and $q(0)$, the rate vector $x(t)$ converges to $\hat{x}$ as $t \to \infty$ and

$$\hat{p}_{b(f)e(f)} = w_f/\hat{x}_f \quad \forall f.$$

Further, the queue-length vector $p(t)$ approaches the bounded set

$$\left\{p \geq 0 : \sum_{i,d}(p_{id}-\hat{p}_{id})\left(R^d_{out(i)}-R^d_{in(i)}+(x^d_i-\hat{x}^d_i)\right)=0\right\}.$$

Proof. Differentiating the Lyapunov function with respect to time and dropping the (t) for notational convenience we obtain

$$\dot{W} = \sum_f (x_f-\hat{x}_f)\left(\frac{1}{x_f}-p_{b(f)e(f)}\right)+\sum_{i,d}(p_{id}-\hat{p}_{id})\left(x^d_i+R^d_{in(i)}-R^d_{out(i)}\right)^+_{p_{id}} \tag{26.34}$$

$$\leq \sum_f (x_f-\hat{x}_f)(\frac{1}{x_f}-p_{b(f)e(f)})+\sum_{i,d}(p_{id}-\hat{p}_{id})\left(x^d_i+R^d_{in(i)}-R^d_{out(i)}\right). \tag{26.35}$$

The last inequality follows by noting that Equations 26.35 and 26.34 are equal if the projection in Equation 26.34 is inactive and if the projection is active, the expression in Equation 26.34 is zero, while the expression in Equation 26.35 is positive due to the fact that $p_{id} = 0$ and the term inside the parenthesis is negative (otherwise, the projection will not be active). Using the fact $w_f/\hat{x}_f = \hat{p}_{b(f)e(f)}$ and adding and subtracting terms, we get

$$\dot{W} = \sum_f (x_f-\hat{x}_f)\left(\frac{1}{x_f}-\frac{1}{\hat{x}_f}+\hat{p}_{b(f)e(f)}-p_{b(f)e(f)}\right)+\sum_{i,d}(p_{id}-\hat{p}_{id})\left(x^d_i-\hat{x}^d_i\right)$$
$$+\sum_{i,d}(p_{id}-\hat{p}_{id})\left(\hat{x}^d_i+R^d_{in(i)}-R^d_{out(i)}\right).$$

Noting that

$$\sum_{i,d}(p_{id}-\hat{p}_{id})(x^d_i-\hat{x}^d_i) = -\sum_f (x_f-\hat{x}_f)(\hat{p}_{b(f)e(f)}-p_{b(f)e(f)}),$$

we get

$$\dot{W} = \sum_f (x_f-\hat{x}_f)\left(\frac{1}{x_f}-\frac{1}{\hat{x}_f}\right)+\sum_{i,d}(p_{id}-\hat{p}_{id})\left(\hat{x}^d_i+R^d_{in(i)}-R^d_{out(i)}\right).$$

Let us now examine each of the terms in the right-hand side of the above equation. It is easy to see that

$$(x_f - \hat{x}_f)\left(\frac{1}{x_f} - \frac{1}{\hat{x}_f}\right) \leq 0.$$

From constraint (Equation 26.24),

$$\hat{x}_i^d \leq \hat{R}_{out(i)}^d - \hat{R}_{in(i)}^d,$$

where we recall that the quantities with hats are the optimal solution to the network utility maximization problem. Thus,

$$\sum_{i,d} p_{id}\left(\hat{x}_i^d + R_{in(i)}^d - R_{out(i)}^d\right) \leq 0,$$

since the rates $R_{in(i)}^d$ and $R_{out(i)}^d$ solve Equation 26.30 and $\hat{R}_{out(i)}^d$ and $\hat{R}_{in(i)}^d$ are feasible solutions to the outflow and inflow rates at node i, for destination d. From the KKT conditions

$$p_{id}\left(\hat{R}_{in(i)}^d + \hat{x}_i^d - \hat{R}_{out(i)}^d\right) = 0.$$

Thus,

$$-\sum_{i,d} \hat{p}_{id}\left(\hat{x}_i^d + R_{in(i)}^d - R_{out(i)}^d\right) = -\sum_{i,d} \hat{p}_{id}\left(\hat{R}_{out(i)}^d - \hat{R}_{in(i)}^d - R_{out(i)}^d + R_{in(i)}^d\right) \leq 0,$$

since the $\hat{R}$'s solve the scheduling problem with $\hat{p}$'s as the weights in the scheduling algorithm. Thus, $\dot{W} \leq 0$. To apply LaSalle's invariance principle, let us consider the set of points $\mathcal{E}$, where $\dot{W} = 0$. The set $\mathcal{E}$ is given by the set of points (x, p) that satisfy

$$\left\{ x_f = \hat{x}_f, \sum_{i,d} (p_{id} - \hat{p}_{id})\left(\hat{x}_i^d + R_{in(i)}^d - R_{out(i)}^d\right) = 0 \right\}.$$

We claim that the largest invariant set $\mathcal{M} \subseteq \mathcal{E}$ is further characterized by $p_{b(f)e(f)} = w_f/\hat{x}_f$. To see this, note that if this condition is violated, then the congestion controller (Equation 26.31) will change the rate from $\hat{x}_f$ and hence the system will move outside $\mathcal{E}$. Thus, LaSalle's invariance principle applies and the theorem is proved. ■

Next, we focus on the scheduling problem (Equation 26.30) to investigate how it can be implemented. By rearranging the summation and using the definitions of $R_{in(i)}^d$ from Equation 26.25 and $R_{out(i)}^d$ from Equations 26.26 and 26.30 can be rewritten as

$$\max \sum_{i,k} \sum_{d} R_{ik}^d (p_{id} - p_{kd}).$$

Using the fact that $\sum_d R_{ik}^d = R_{ik}$, the scheduling problem becomes

$$\max \sum_{i,k} R_{ik} \max_d (p_{id} - p_{kd}).$$

Using Equations 26.28 and 26.29, the scheduling problem further reduces to

$$\max_{\sum_m \pi_m = 1} \sum_{\{i,k,m:(i,k)\in A_m\}} \pi_m \max_d (p_{id} - p_{kd}) \tag{26.36}$$

$$= \max_{\sum_m \pi_m = 1} \sum_m \pi_m \sum_{(i,k)\in A_m} \max_d (p_{id} - p_{kd}) \tag{26.37}$$

$$= \max_m \sum_{(i,k)\in A_m} \max_d (p_{id} - p_{kd}), \tag{26.38}$$

where the last equality follows from the fact that the expression in Equation 26.37 is maximized when π_m is set to 1 for the A_m that maximizes the expression in Equation 26.38, and zero for all the rest. Thus, the scheduling problem becomes one of finding a schedule with the maximum weight, where the weight of a link is given by

$$\max_d (p_{id} - p_{kd}).$$

This is called the *back-pressure* algorithm.

Let us examine the price-update Equation 26.32. This equation closely resembles the queue dynamics in the queue holding packets for destination d at packet i. However, notice that the arrival rate into the queue includes the term

$$R^d_{in(i)} = \sum_j R^d_{ji},$$

which is not the true arrival rate at the queue, but it is the potential rate at which packets could arrive from other nodes to node i. This rate may not be realizable if some of the other nodes have empty queues. However, note from the back-pressure algorithm (Equation 26.38) that $R^d_{ji} = 0$ if $p_{jd} = 0$. Thus, hop (i, j) is not scheduled if $p_{jd} = 0$. Hence, one can indeed interpret the price as the queue length at node i for packets destined for node d. One can then simply use the lengths of the per-destination queues as the prices, and no price computation is necessary.

Notice that the congestion controller (Equation 26.31) for flow f uses only the ingress queue length to regulate the flow's arrival rate, and not the lengths of the queues in the interior of the network. This may appear counter-intuitive given that congestion can occur anywhere in the interior of the network and not only at the edges. The key component that enables the use of only the ingress queue lengths is the nature of the back-pressure scheduling algorithm (Equation 26.38). In particular, any link that is highly congested within the network will be given priority, effectively preventing other links in its interference range from transmitting. This increases the congestion level of the inactive links, which then cause congestion to other links in their neighborhood. Thus, the congestion at any link in the network gradually spreads to all the links in the network. This is the fundamental reason why it is sufficient for the congestion controller to use only the ingress queue-length values.

26.7.1 Stochastic Channel State and Arrival Processes

Our focus has been on networks operating in continuous-time and composed of deterministic components. However, in reality, there are numerous stochastic factors that arise in their operation, such as random fluctuations in the channel quality between nodes. We discuss some of the important steps in incorporating such effects into the design framework we have discussed so far.

To model channel variations, we assume that the network channel state can be in one of many states belonging to a finite set, say $\mathcal{J}$. We assume that the channel state process is described by a stationary Markov Chain with β_j denoting the stationary probability of the channel state being j. We let Γ_j denote the set of feasible link rates when the current state is $j \in \mathcal{J}$. Thus, if the state of the channel is j at time t, the scheduled link rates $R(t) := (R_{ij}(t))_{(i\,j)\in\mathcal{L}}$ must lie in Γ_j.

In this dynamic scenario, the back-pressure policy need to be modified to conform with the existing capacity constraints. Thus, the scheduler performs the following optimization to determine the link rates at time t:

$$R(t) \in \arg\max_{\{\mu\in\Gamma_j\}} \sum_{\{(n,m)\in\mathcal{L}\}} \mu_{(n,m)} \max_d (p_{nd}(t) - p_{md}(t)).$$

To allow for randomness in the arrival process to model various implementational details, the congestion controller component can be modified in the following manner:

$$E[x_f(t+1) \mid p_{b(f),e(f)}(t)] = \min\{M, x_f(t) + \alpha(KU'_f(x_f(t)) - p_{b(f),e(f)}(t))\},$$

where K is a positive design parameter, $\alpha > 0$ is a small step size parameter, and M is a finite constant that prevents the congestion controller to inject huge amount of traffic into the network in a short duration.

Notice that both the resource allocation and the congestion controller algorithms are inspired by their deterministic, continuous-time counterparts. However, their dynamic and stochastic nature necessitates the analysis of the resulting stochastic system. To that end, we first note that the queue-length process of the network forms a Markov chain. Thus, it is no longer possible to state that the queues and rates will converge to a deterministic set of points. Instead, one needs to revert to stochastic convergence results. Next, we discuss some of the key technical components of the stochastic analysis, while omitting the details of the analysis. We refer the interested reader to [12–21].

The following stochastic version of the Lyapunov stability theorem, namely the Foster's criterion, is extensively used in the proof.

Theorem 26.6: Foster's Criterion

Suppose $\{q(t)\}_{t=1}^{\infty}$ *is an aperiodic and irreducible Markov chain over a countable state space, and let* V *and* W *be nonnegative functions such that* $V(q) \geq W(q)$ *for* $q \in \Omega^c$, *where* Ω *is a finite subset of the state space. If*

$$E[V(q(t+1)) \mid q(t) = q] < \infty, \quad q \in \Omega,$$
$$E[V(q(t+1)) - V(q(t)) \mid q(t) = q] \leq -W(q), \quad q \in \Omega^c,$$

then, the Markov chain is stationary and ergodic, and

$$E[W(q(\infty))] < \infty.$$

The Foster criterion states that when the *mean drift* of a function of the Markov chain is negative outside a finite set Ω, then it has to be stationary, and depending on the strength of the drifts, measured by $W(q)$, one can also bound a function of the first moment of the steady-state distribution. Note that the nature of the statement and the result is parallel to the Lyapunov stability criterion we discussed before. Thus, the Lyapunov functions used in the continuous-time analysis of the cross-layer algorithms can be used as the V function in the Foster's criterion. The stochastic and dynamic components influence the form of the W function and the Ω set. ■

26.8 Summary

In this chapter we have tried to present the main ideas that drive the design of congestion control protocols in the multihop communication systems and, in particular, the Internet. The ideas are close to that of multicommodity flow problems that arise in many supply chain contexts. With our focus on fundamental ideas, we have omitted the details of the actual protocols that are used on the Internet. In reality, the tie between real-life congestion control protocols and our simple control-theoretic models is quite strong, and has been explored in greater detail in works such as [1,2].

References

1. S. Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, Now Publishers, Delft, The Netherlands, 2, 2007.
2. R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, Berlin, Germany, 2004.

3. F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.
4. F. P. Kelly. Models for a self-managed Internet. *Philosophical Transactions of the Royal Society*, A358:2335–2348, 2000.
5. F. P. Kelly. Mathematical modelling of the Internet. In *Mathematics Unlimited—2001 and Beyond* (Eds. B. Engquist and W. Schmid), pp. 685–702, Springer-Verlag, Berlin, 2001.
6. F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
7. J. Mo and J. Walrand. Fair end-to-end window-based congestion control. In *SPIE International Symposium*, Boston, MA, 1998.
8. J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.
9. H. Khalil. *Nonlinear Systems*, 2nd edn. Prentice-Hall, Upper Saddle River, NJ, 1996.
10. S. H. Low and D. E. Lapsley. Optimization flow control, I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 861–875, December 1999.
11. F. Paganini. A global stability result in network flow control. *Systems and Control Letters*, 46(3): 153–163, 2002.
12. A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *Proceedings of IEEE INFOCOM*, Miami, FL, 2005.
13. L. Georgiadis, M. J. Neely, and L. Tassiulas. *Resource Allocation and Cross-Layer Control in Wireless Networks*. Foundations and Trends in Networking, NOW Publishers, Delft, The Netherlands, 2006.
14. M.J. Neely, E. Modiano, and C.E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. *Proceedings of IEEE INFOCOM*, San Francisco, CA, April 2003. To appear.
15. L. Tassiulas. Scheduling and performance limits of networks with constantly varying topology. *IEEE Transactions on Information Theory*, 43(3):1067–1073, May 1997.
16. L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.
17. A. Eryilmaz and R. Srikant. Joint congestion control, routing and MAC for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1514–1524, August 2006.
18. X. Lin, N. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, August 2006.
19. P. Giaccone, B. Prabhakar, and D. Shah. Towards simple, high-performance schedulers for high-aggregate bandwidth switches. In *Proceedings of IEEE INFOCOM*, New York, NY, 2002.
20. M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proceedings of IEEE INFOCOM*, Miami, FL, 2005.
21. A. L. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50:401–457, 2005.

VII

Special Applications

27

Advanced Motion Control Design

Maarten Steinbuch
Eindhoven University of Technology

Roel J. E. Merry
Eindhoven University of Technology

Matthijs L. G. Boerlage
General Electric Global Research

Michael J. C. Ronde
Eindhoven University of Technology

Marinus J. G. van de Molengraft
Eindhoven University of Technology

27.1 Introduction

Mechanical systems with actuators that have the primary function to position a load are called motion systems. The actuator can be either hydraulic, pneumatic, or electric. The trend is toward energy efficient and clean electric (and piezo) drives and motors. Motion systems are different from robotic systems in the sense that the freedom of trajectory planning and motion profile is often limited. A difference with the field of active vibration systems is that a real motion is done. Linear and rotational drives are examples, but also state-of-the art planar 6DOF (degree-of-freedom) motion platforms. A typical feature of motion systems is that the system dynamics can often be approximated well by using linear models, albeit sometimes with significant flexible dynamics, especially in high-performance and high-DOF systems. Cheaper motion systems typically have friction in the guidance. Backlash is often prevented by application of direct drive actuators. The sensors used nowadays are often encoders, which can be used down to extreme resolutions (below nm).

The industrial state-of-the-art control of motion systems can be summarized as follows. Most systems, by design, are either decoupled, or can be decoupled using static I/O (input-output) transformations. Hence, most motion systems and their motion software architecture use single-input-single-output (SISO) control design methods and solutions. Feedback design is mostly done in the frequency domain, using loopshaping techniques. A typical motion controller has a PID (proportional-integral-derivative) structure, with a low pass at high frequencies and one or two notch filters to compensate flexible dynamics [1]. In addition to the feedback controller, a feedforward controller is applied with acceleration, velocity, and

friction feedforward from the reference signal. The setpoint itself is a result of a setpoint generator with jerk limitation profiles [2].

In the case more is known about the disturbances, such as repetitive disturbances or setpoints, more recent developments are the use of Iterative Learning and Repetitive control. Also, in the field of advanced and high precision motion systems, such as wafer scanners, the use of more advanced setpoints, or higher order feedforward is used (snap feedforward [2,3]). Moreover, if the requirements increase, the dynamic coupling between the various DOFs can no longer be neglected and more advanced multiple-input-multiple-output (MIMO) control is required. This might also result is strict requirements for system identification results focussed on robust control design.

In this work we would like to summarize in detail the state-of-the-art control design of motion systems, and introduce a step-by-step procedure to be able to extend the SISO loopshaping method into the MIMO motion field. This step-by-step procedure consists of (1) interaction analysis, (2) decoupling, (3) independent SISO design, (4) sequential SISO design, and finally, (5) norm-based MIMO design. We will illustrate the design features on a 3DOF Atomic Force Microscope (AFM) motion system.

As definitions, we will use the following. Centralized control: the transfer function matrix of the controller is allowed to have any structure. Decentralized control: diagonal controller transfer function, but constant (by frequency) decoupling manipulations of inputs and outputs are allowed. Independent decentralized control: a single loop is designed without taking into account the effect of earlier or later designed loops. Sequential decentralized control: a single loop is designed with taking into account the effect of all earlier closed loops.

In Section 27.2 a brief description of the dynamics of motion systems will be given, followed by feedforward design in Section 27.3. Section 27.4 shows the feedback design of both SISO problems and the new procedure for MIMO motion problems. The application to an AFM is shown in Section 27.5. Finally, the most important conclusions will be summarized in Section 27.6.

27.2 Motion Systems

The (linear) dynamic behavior of motion systems is often dominated by mechanics. Therefore, physical interpretation of the mechanical systems can facilitate transparent multivariable control design and decoupling. In the subsequent derivations, we assume that the current amplifier or electrical part of the actuator is so fast that it can be approximated as a gain. From either finite element modeling, linearized first principles modeling, or reduced order continuous system descriptions, the following finite-dimensional, linear, multiple DOF equations of motion can be derived [4],

$$\begin{aligned} M\ddot{q} + D\dot{q} + Kq &= B_o u \\ y &= C_{oq} q. \end{aligned} \tag{27.1}$$

Herein, M, D, K are the mass matrix, viscous damping matrix and stiffness matrix, respectively. In this model, only position measurements are considered. Extensions to include velocity and acceleration measurements can be found in [5]. We assume that the mass matrix is positive definite and the stiffness matrix is semi-positive definite. The difference with robotic systems is that for motion systems the parameter matrices M, D, and K are contant matrices in most cases. Several assumptions on the properties of D will be discussed shortly. The vector $q \in \mathbb{R}^{n_s}$ represents the displacement of the nodes of the lumped parameter system. From the undamped vibration problem, without input, the real mode shapes ϕ and eigen or natural frequency ω can be determined solving the following generalized eigenvalue problem:

$$K\phi = \omega^2 M\phi, \quad \phi \neq 0. \tag{27.2}$$

The zero-valued eigenfrequencies correspond to the so-called rigid body (RB) modes of the system. With p times multiplicity of eigenfrequencies, there exists a set of p linearly independent eigenmode

shapes. Then, these eigenmode shapes are not unique (multiplicity leads to nonunique eigenvectors, which is typical for RB modes). Let the modal matrix Φ contain columns that span the directions of the mode shapes $\phi_i, i = 1, \ldots, n_s$. Then, the equations of motion (Equation 27.1) can be expressed in *modal* coordinates,

$$\begin{aligned} M_m\ddot{\eta} + D_m\dot{\eta} + K_m\eta &= \Phi^T B_o u \\ y &= C_{oq}\Phi\eta, \end{aligned} \tag{27.3}$$

where $M_m = \Phi^T M\Phi$ and $K_m = \Phi^T K\Phi$ are diagonal. The matrix $D_m = \Phi^T D\Phi$ is only diagonal in special cases. For example, in the case of Rayleigh or proportional damping, where it is assumed that, $D = \alpha M + \beta K$ with α, β being the nonnegative scalars [4, p. 303]. Also with modal or classic damping, D_m is diagonal [4]. These damping models are often justified for structural analysis of lightly damped systems [4,5]. When D_m is diagonal and Equation 27.3 is multiplied from the left with M_m^{-1}, we obtains,

$$\begin{aligned} \ddot{\eta} + 2Z\Omega\dot{\eta} + \Omega^2\eta &= M_m^{-1}\Phi^T B_o u \\ y &= C_{oq}\Phi\eta, \end{aligned} \tag{27.4}$$

where $\Omega^2 = M_m^{-1}K_m$, $Z = \text{diag}\{\zeta_i\}, i = 1, \ldots, n_s$ are diagonal. We define,

$$y(s) = G_p(s)u(s), \tag{27.5}$$

where $G_p(s)$ follows from Equation 27.4 as follows. With the assumption that D_m is diagonal and defining $C_m = C_{oq}\Phi$, $B_m = M_m^{-1}\Phi^T B_o$, with Φ real valued, we write,

$$G_p(s) = C_m G_m(s) B_m, \tag{27.6}$$

with

$$G_m(s) = \text{diag}\{g_{m,i}(s)\}, \quad g_{m,i}(s) = \frac{1}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}, \tag{27.7}$$

for $i = \{1, \ldots, n_s\}$, or alternatively, written in summation form,

$$G_m(s) = \sum_{i=1}^{n_s} \frac{c_{mi} b_{mi}^H}{s^2 + 2\zeta_i\omega_i s + \omega_i^2}. \tag{27.8}$$

with c_{mi}, b_{mi} the columns and rows of C_m, B_m, respectively. This shows that each, multivariable, motion system can be described by a summation of second-order systems. The matrices C_m, B_m determine which input–output combination is connected to a specific mode.

As an example, Figure 27.1 shows a magnitude frequency response function (FRF) and the underlying modes of a system.

The ability to decouple a mechanical system (with modal or proportional damping) depends on the actuator and sensor locations, the number of (dominant) modes and the alignment of mode shapes with the sensor and actuator matrices (dominant modes must be both in $\text{Ker}(C_m)^\perp$ and in $\text{Im}(B_m)$). Methods to place actuators and sensors to control each mode independently are discussed in more detail in [5,6]. High-performance motion systems are often designed light and stiff, with the aim to move flexible mode behavior to frequencies above the intended closed-loop bandwidth. Especially for positioning systems, the plant at low frequencies behaves as a RB.

In each DOF, the plant has RB modes. For six Cartesian degrees of freedom, six eigenfrequencies (12 poles) will be equal to zero in case there are no stiffness connections involved in the actuator, such as, the case with Lorentz force actuators. For the piezo drives applied in the case of the AFM application later in this work, the actuators can be seen as position actuators and the first resonance is the result of the stage mechanics mounted on the fixed world with limited stiffness. In any case, there exists a set of six linear

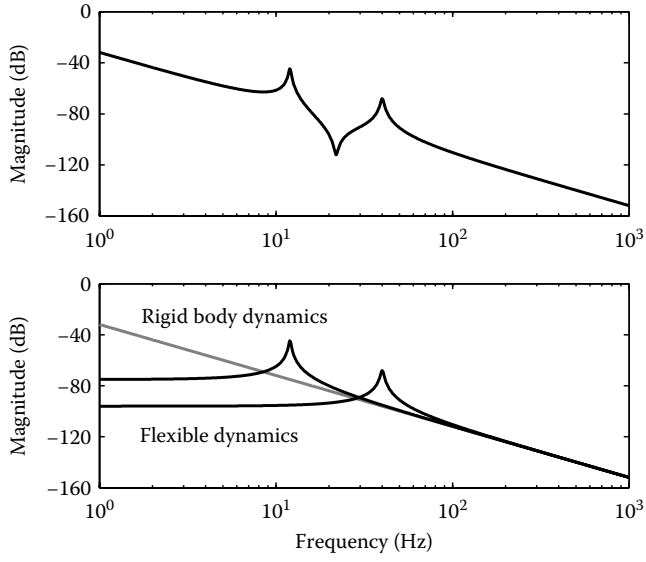

FIGURE 27.1 A frequency response measurement (top) can be interpreted as a summation of flexible and rigid-body mode contributions (bottom).

independent eigenvectors for these RB modes. One may choose any orthogonal base to decouple the RB behavior of the system [7]. Each axis of this base may be aligned with specific performance objectives or a particular disturbance direction. When the number of sensors and actuators exceed the number of RB modes, and C_m, B_m are invertible, input (and output) transformations exist to transform the system into independently controllable directions. This situation is also known as over-actuation, and is a new field of research.

Here, we focus on the control of linear time invariant electromechanical motion systems that have the same number of actuators and sensors as RB modes. Typical applications are high-performance positioning stages used in semiconductor manufacturing [8], electron microscopy, or component placement machines. The dynamics of such systems are often dominated by the mechanics, which are therefore constructed to be light and stiff, so that resonance modes due to flexible dynamics appear only at high frequencies.

$$G_p(s) = \sum_{i=1}^{N_{rb}} \frac{c_i b_i^T}{s^2} + \sum_{i=N_{rb}+1}^{N} \frac{c_i b_i^T}{s^2 + 2\zeta_i \omega_i s + \omega_i^2}. \tag{27.9}$$

Herein, N_{rb} denotes the number of RB modes. The parameters ζ_i, ω_i are the relative damping and the resonance frequency of the flexible modes, respectively. The vectors c_i, b_i span the directions of the ith mode shapes and are constant for all frequencies. The resonance frequencies ω_i are high, hence the plant can be approximately decoupled using static input (and/or output) transformations, T_u, T_y, respectively so that,

$$G_{yu}(s) = T_y G_p(s) T_u = G(s) + G_{flex}(s) \tag{27.10}$$

$$G(s) = \frac{1}{ms^2} I, \tag{27.11}$$

where $m \in \mathbb{R}^1$ (if only translations, otherwise the expression should be extended with the inertias) and $G_{flex}(s)$ contains the flexible dynamics of the plant and is often nondiagonal. In many applications, the frequencies and damping of the resonance modes change in the life cycle of the plant and are sensitive to changes in position. Hence, inversion of these dynamics leads to robustness problems. The objective is to control the RB behavior of the plant with high fidelity.

27.3 Feedforward Control Design

Industrial motion systems are designed to perform step and scanning movements or pick and place operations. Typically, piecewise finite order polynomials are used as reference profile. These profiles have motion phases with constant velocity, acceleration, jerk, the derivative of jerk, and so on. These reference trajectories have mostly low frequency energy, hence resonance dynamics are little excited, especially in light and stiff constructions. If resonance modes are excited, input shaping techniques can be used to reduce the energy in specific frequency bands.

As mentioned before, in modern motion control platforms the use of acceleration feedforward K_{fa}, and also velocity K_{fv} and friction feedforward K_{fc} (for the additive repeatable disturbances) are standard features nowadays, see Figure 27.2. In the rest of this section, we will extend the notion of inversion of the dynamics to the general MIMO case.

Using the model assumptions of the previous section, we can derive the following simplified model for RB systems with high-frequency resonance dynamics. The model contains a constant matrix representing *all* modal contributions in low frequencies added to a RB model, so that

$$\hat{G}(s) = G_{rb}\frac{1}{s^2} + \hat{G}_{flex} \tag{27.12}$$

The objective we focus on here is to follow a given reference profile at all time instances during the motion. We assume that the plant output y is measured at the location where tracking performance is to be achieved. Otherwise we speak of inferential motion control [9]. Therefore, the low-frequency tracking problem can be studied considering the transfer function from reference trajectory r to the servo error e, as shown in Figure 27.2;

$$e = S_o(s)(I - T_y G(s) T_u F(s))r \tag{27.13}$$

with the output sensitivity defined as $S_o(s) = (I + T_y G(s) T_u K(s))^{-1}$. It is common to design a feedforward controller that approximates the inverse of the plant. As many motion systems contain dominant RB behavior, it is in practice to use RB feedforward inversion by means of *acceleration feedforward*, so that

$$F(s) = \tilde{G}_{rb}^{-1} s^2 \tag{27.14}$$

that is, $\tilde{G}_{rb}$ is the MIMO extension of K_{fa}. Using the plant model from Equation 27.12, the transfer function of interest equals

$$\begin{aligned} e &= S_o(s)\left(I - \left(\tilde{G}_{rb}\frac{1}{s^2} + \tilde{G}_{flex}\right)(\tilde{G}_{rb}^{-1}s^2)\right) r \\ &= -S_o(s)\tilde{G}_{flex}\tilde{G}_{rb}^{-1}s^2 r \end{aligned} \tag{27.15}$$

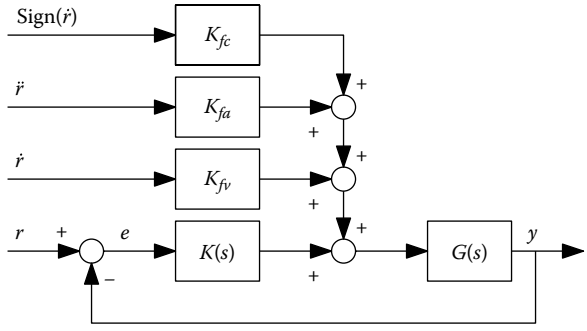

FIGURE 27.2 Feedback control loop with acceleration, velocity, and friction feedforward.

It is clear that there exists a residual transfer function between the acceleration of the reference trajectory and the servo error. The term $\tilde{G}_{flex}\tilde{G}_{rb}^{-1}$ is constant. Hence, when no feedback control is applied ($S_o = I$) the servo error equals the acceleration of the reference profile scaled with the factor $\tilde{G}_{flex}\tilde{G}_{rb}^{-1}$. When feedback control is applied, the output sensitivity function has at least slope +2 at low frequencies, so that the servo error shows peaks during nonzero jerk phases of a motion. The residual transfer function between acceleration and the servo error is responsible for both cross-talk and low frequency tracking of the diagonal terms. This transfer function cannot be reduced using acceleration feedforward.

To increase tracking performance, higher order derivatives of the reference trajectory can be used as well. The *jerk derivative feedforward controller* [2,3] considers the fourth order approximate so that the new MIMO jerk derivative feedforward controller becomes

$$F(s) = F_{acc}s^2 + F_{djerk}s^4 \tag{27.16}$$

$$F_{acc} = \tilde{G}_{rb}^{-1}, \tag{27.17}$$

$$F_{djerk} = -\tilde{G}_{rb}^{-1}\tilde{G}_{flex}\tilde{G}_{rb}^{-1} \tag{27.18}$$

The first term equals conventional acceleration feedforward while the second term equals (multivariable) jerk derivative feedforward (*djerk*). If the RB modes of the plant are decoupled, this jerk derivative feedforward part can compensate for the flexible modes that do not have to be decoupled at low frequencies.

Note that due to this implementation, jerk derivative feedforward can be tuned subsequently to tuning acceleration feedforward control. Therefore, manual tuning is facilitated, gradually increasing the complexity of the feedforward controller.

27.4 Feedback Control Design

27.4.1 System Identification—Obtaining the FRF

Consider the standard unity feedback configuration depicted in Figure 27.3. The first necessary step to perform any design of a controller is to identify the system dynamics. In case the machine has been realized, a measurement is appropriate. For motion systems, typically, a frequency response measurement is done using noise signals, single sine, swept sine, or specialized multisines. It is important to note that since most motion systems do have a RB mode (i.e., a double integrator as basic dynamics), in open-loop they are unstable.

For that reason, it is convenient, even necessary, to stabilize the system by a low bandwidth PD controller $K(s) = Ds + P$ first. Typically, the controller zero (ratio P/D) is taken equal to 1–10 Hz, and if the sign is known of the system, a simple procedure can be used to tune up the gain, while keeping the ratio P/D constant, or first D is increased, and then P, while giving the motion system a modest setpoint (the friction has to be overcome while tuning). Such a procedure is known as time-domain tuning, and can only lead to low bandwidth controllers, which is fine for the first stabilized situation. Once this has been done successfully, an identification experiment can be done. Although many engineers perform this under closed-loop conditions, direct measurement of the plant is done by measuring the output y over

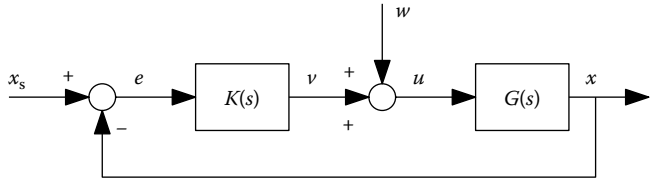

FIGURE 27.3 Standard unity feedback configuration.

the input u. It is well-known that due to correlation, if significant disturbances occur between u and y, a linear combination of plant and inverse controller will be measured [10,11].

For this reason, the instrumental variable method should be used: measure for instance the process sensitivity (y/w), see again Figure 27.3, and the sensitivity (u/w) then divide the two frequency responses. This will give a unbiased estimate of the plant. The same procedure can be followed for MIMO systems, but appropriate matrix transformations should be done for every frequency measurement point.

27.4.2 Loopshaping—The SISO Case

Now a FRF is given, let us start by giving a short summary of loopshaping for SISO systems. As an example consider the measured FRF of an industrial printer in Figure 27.4.

The key idea of loopshaping is the modification of the controller such that the open-loop is made according to specifications. The reason this works so well is that the controller enters linearly into the open-loop transfer $l(j\omega) = g(j\omega)k(j\omega)$, so that it is fast and easy to reason what is to be changed in the controller. However, in practice all specifications are of course given in terms of the final system performance, that is, as closed-loop specifications. So we should convert the closed-loop specifications into specifications on the open-loop. The area of quantitative feedback theory (QFT) [12–14] is based on this idea. Typically we should model the external signals (setpoint, disturbances) and determine the specifications on the error. The knowledge about the setpoint is mostly not relevant for determining the required bandwidth, because that is covered by the feedforward. The modeling of the disturbances is less trivial, especially for the MIMO case [15].

Let us assume we know the spectral contents of the disturbance. Take as an example the simple case of a force disturbance at the plant input being a sinusoid of known amplitude. If we know the specification on the error amplitude we can derive the requirement on the process sensitivity at that frequency. Knowing the process gain at that frequency, the required reduction of the sensitivity can be calculated. Since at low frequencies the sensitivity can be approximated as the inverse of the open-loop, we can translate this into a specification of the open-loop at that frequency. Because we know that the slope of the open-loop of a well tuned motion system will be between -2 and -1 (and certainly around -1 near crossover), we

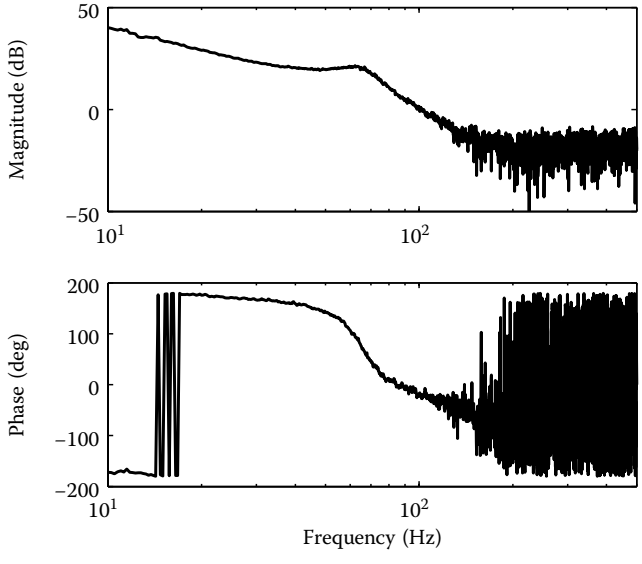

FIGURE 27.4 Measured FRF of an industrial printer.

can estimate the required crossover frequency ω_c. Now, the closed-loop specifications are translated to specifications on the open-loop and the next phase is the loopshaping process itself.

Knowing the required crossover, the FRF of the plant is observed. First, the structure of the controller is chosen such that stabilization is possible, that is, check the plant phase at the crossover ω_c, and add D action if required. Typically, the first motion controller will be a PD with the zero (P/D) at the crossover ω_c. After that, we add integral action, typically with the zero of the integral action $\omega_c/5$. To prevent noise at high frequencies in the loop, a (second order) low pass is added with a cutoff frequency $5 \cdot \omega_c$, and a relative damping of 0.5. If parasitics are apparent in the plant FRF then notch filters can be applied. In case the disturbances contain specific frequency components, additional filters might be useful to improve performance, such as inverse notch filters or even repetitive controllers. In all phases of loopshaping, the tuning is done with good margins, where the modulus margin, that is, the peak of the sensitivity, is the most convenient: it shows precisely the frequency at which the distance to the -1 point in the Nyquist diagram is smallest. Then additional attention is needed in that range to change phase or amplitude.

27.4.3 Loopshaping—The MIMO Case

In MIMO systems it is much less trivial to apply loopshaping. The stability is determined by the closed-loop polynomial, $\det(I + L(s))$, and the characteristic loci can be used for this graphically. The characteristic loci are the eigenvalues of the FRF $L(j\omega)$ of the open-loop transfer function matrix $L(s)$. A system with N inputs and N outputs has N characteristic loci. For stability analysis one can use the *Generalized Nyquist criterium* [16]. With a nonparametric FRF matrix of the open-loop, the characteristic loci can be plotted in the complex plane. If each eigenvalue locus does not encircle the point $(-1, 0)$, the MIMO system is closed-loop stable (for open-loop stable systems). The shaping of these eigenvalue loci is not straightforward if the plant has large off-diagonal elements (interaction). In that case, a single element of the controller will affect more eigenvalue loci. This may lead to many design iterations and loss of intuition. In the work of [17] it is shown that up to some extent, when the open-loop is *diagonal dominant*, one can allow plant interaction and still use frequency response loop shaping design techniques. However, the notion of margins is far from trivial: for example, the phase margin seen in the a plot of the characteristic loci, is valid for a phase change at the same time in all loops!

In the special case that the open-loop transfer function matrix is diagonal, $L(s) = \mathrm{diag}\{l_i(s)\}$, that is, the open-loop is decoupled, it follows that,

$$\det(I + L(s)) = \prod_{i=1}^{n} \det(1 + l_i(s)), \tag{27.19}$$

so that the characteristic loci of the open-loop transfer function matrix are determined by the frequency response of each decoupled open-loop function $l_i(s)$. The MIMO feedback control design complexity reduces to that of SISO feedback control design. Many classical MIMO control design methods aim at decoupling the open-loop function at loop breaking point (e.g., at the inputs or at the outputs of the plant).

The strong nonintuitive aspect of MIMO loopshaping and the fact that SISO loopshaping is used often, are major obstacles in application of modern design tools in industrial motion systems. For that reason, we propose in this work a step-by-step approach in which complexity is only increased if necessary. This recipe for control design for MIMO motion systems consists of the following steps:

1. Interaction analysis
2. Decoupling transformations
3. Independent feedback control design
4. Sequential feedback control design
5. Norm-based control design.

All but the last step can be executed with a nonparametric model of the plant (frequency response). The norm-based control design requires a parametric model of the plant. A crucial step before controller synthesis with operator norms is scaling and conditioning of the plant.

27.4.3.1 Interaction Analysis

The goal of the interaction analysis is to identify two-sided interactions in the plant dynamics. If there is no two-sided interaction, we can choose a diagonal transfer function matrix controller to achieve closed-loop stability as if the open-loop is decoupled. That is, from a stability point, feedback design is just a collection of SISO design problems.

We use two measures for plant interaction: (1) relative gain array (RGA) per frequency, and (2) structured singular value (SSV) of interaction as multiplicative output uncertainty.

Definition 27.1:

The frequency-dependent relative gain array (RGA) [16,18] *is calculated as*

$$RGA(G(j\omega)) = G(j\omega) \times (G(j\omega)^{-1})^T, \tag{27.20}$$

where $\times$ *denotes element-wise multiplication.*

The rows and columns of the RGA sum to 1 for all frequencies ω (rad/s). If $(RGA)(j\omega) = I,\ \forall\omega$, perfect two-sided decoupling is achieved. If not, a further interaction analysis can be done using the next measure.

Definition 27.2:

The structured singular value (SSV) interaction measure: with $E_T(j\omega) = G_{nd}(j\omega)G_d^{-1}(j\omega)$,

$$\mu_D(E_T(j\omega)) < \frac{1}{2}, \forall\omega. \tag{27.21}$$

where μ_D *is the structured singular value,* [19], *with respect to the diagonal(decoupled) structure of the feedback controller.*

If a diagonal transfer function matrix controller is used, controller gains must be small (much smaller than 0 dB) at frequencies where this condition is not met. This gives a rough indication on achievable performance as a result of interaction.

27.4.3.2 Decoupling Transformations

A common method to reduce plant interaction is to redefine the input and output of the plant. One can combine several inputs or outputs to control the system in more decoupled coordinates. For motion systems most of these transformations are found on the basis of kinematic models. Herein, combinations of the actuators are defined so that actuator variables act in independent (orthogonal) directions at the center of gravity. Likewise, combinations of the sensors are defined so that each translation and rotation of the center of gravity can be measured independently. This is basically the inversion of a kinematic model of the plant.

As motion systems are often designed to be light and stiff, kinematic decoupling (RB decoupling) is often sufficient to achieve acceptable decoupling at the crossover (bandwidth) frequencies. Also specific,

modal, dyadic, structures of the plant can be used to decouple flexible mode dynamics (parasitics with low damping): then again a constant I/O transformation is often sufficient to decouple the system.

Some literature describes methods to find dynamic decoupling, or static decoupling at different frequencies (at crossover is the most relevant of course). However, such methods are nontrivial to apply.

The effect of the decoupling transformations can be measured with the interaction measures derived earlier.

27.4.3.3 Independent Loop Closing

For systems where interaction is low, or the decoupling is almost successful, one can design a diagonal controller by closing each control loop independently. We call this independent loop closing. The residual interaction can be accounted for in the analysis.

For this we make use of the following decomposition, see also [20]:

$$\det(I + GK) = \det(I + E_T T_d)\det(I + G_d K), \tag{27.22}$$

with, $T_d = G_d K(I + G_d K)^{-1}$. A typical choice is to take $G_d(s)$ as only the diagonal terms of the plant transfer function matrix. The effect of the nondiagonal terms of the plant $G_{nd}(s) = G(s) - G_d(s)$ is accounted for by $E_T(s)$. Then, the MIMO closed-loop stability assessment can be split up in two assessments; the first for stability of N noninteracting loops, namely $\det(I + G_d(s)K(s))$, the second for stability of $\det(I + E_T(s)T_d(s))$. In the second stability test, T_d is the complementary sensitivity function of the N decoupled loops. If $G(s)$ is stable and $T_d(s)$ is stable, one can use the small gain theorem [16], to find a sufficient condition for stability of $\det(I + E_T T_d)$ as,

$$\rho(E_T(j\omega)T_d(j\omega)) < 1, \forall\omega, \tag{27.23}$$

where ρ is the spectral radius. With introduction of conservatism, the following sufficient condition is

$$\mu_{T_d}(E_T(j\omega)T_d(j\omega)) < 1, \forall\omega \quad \Rightarrow \tag{27.24}$$

$$\overline{\sigma}(T_d(j\omega)) < \mu_{T_d}(E_T(j\omega)^{-1}, \forall\omega \tag{27.25}$$

where μ_{T_d} is the SSV [19], with respect to the diagonal(decoupled) structure of T_d. As $\overline{\sigma}(T_d(j\omega)) = \max_i |T_{d,ii}(j\omega)|$, condition 27.24 implies that a single bound is acting on the worst case (highest gain) loop of T_d.

Each control loop does not take into account other (earlier) tuned control loops, hence the closed-loop remains stable if an arbitrary loop is opened. Due to the fact that a sufficient condition is used, independent loop closing usually leads to conservative (low performance) designs.

27.4.3.4 Sequential Loop Closing

If the interaction is a larger, but still not really significant (depends on the case how this should be quantified) the sequential loop closing (SLC) method is appropriate. Herein, one utilizes information about each controller designed earlier. The control designs are now dependent and we can save some conservatism. The controller is still a diagonal transfer function matrix.

In principle, one starts with the open-loop FRF of the MIMO plant. Then one loop is closed using SISO loopshaping. The controller is taken into the plant description, and a new FRF is obtained with one input and output less. Then, the next loop is designed and so on. We can formalize this as follows. Each SISO controller k_i, from $K = \text{diag}\{k_i\}, i = \{1, \ldots, n\}$, is designed using the property [21],

$$\det(I + GK) = \prod_{i=1}^{n} \det(1 + g^i k_i), \tag{27.26}$$

where for each ith design step, the equivalent plant g^i is defined as the lower fractional transformation (LFT):

$$g^i = \mathcal{F}_l(G, -K^i) \tag{27.27}$$

where $K^i = \text{diag}\{k_j\}, j = \{1, \ldots, n\}, j \neq i$.

The multivariable system is nominally closed-loop stable if in each design step the system is closed-loop stable. The system remains closed-loop stable if the loops are opened in the reverse order as in they were designed. If an arbitrary loop is opened there is no guarantee for closed-loop stability. The robustness margins in each design step do not guarantee robust stability of the final multivariable system [22].

Various sequential design methods are developed in the framework of QFT [12–14]. Also, cascade control design, is often quite similar to sequential design [16, p. 422].

Drawbacks of sequential design are, first, the ordering of the design steps may have great impact on the achievable performance. There is no general approach to determine the best sequence for design. This may lead to many design iterations, especially for large MIMO systems. Second, there are no guarantees that robustness margins in earlier designed loops are preserved. The robustness margins at each design step do not indicate robustness of the final closed-loop system. Third, as each design step usually considers only a single output, the responses in earlier designed loops may degrade, making iterative design necessary.

Nonetheless, sequential loopshaping can be a great method to reduce conservatism of the independent loop closing method while the complexity (SISO loopshaping, FRFs as plant model) is very low.

27.4.3.5 Norm-Based Control Design

If the previous step (SLC) was not successful, the next step is to start norm-based control design. This method requires a parametric (e.g., state-space model) and weighting filters to express the control problem in terms of an operator norm (like H_2 and H_∞). Parametric modeling of plant dynamics is often time consuming, costly, and generally increases controller design complexity. Only recent developments tend to solve the problem also for FRF (data)-based models [23].

In the application we show how parametric model complexity can be build up step-by-step: considering the unmodeled dynamics as (unstructured) uncertainty, up to higher-order models.

Some tricks and tips for norm-based control of motion systems can be found in [1]. As an example, first design a low-performance decentralized controller using one of the tools from the previous steps, mentioned above. Then include this controller in the standard plant and calculate for the given weighting filters all MIMO FRFs and the norm. This gives a good initial setup for (fine)tuning the weights now.

Also the effect of scalings should be addressed carefully. In many existing practical systems, amplification factors of actuators and sensors are chosen arbitrary. As operator norms express all matrix properties in a scalar measure, scaling of the augmented plant (and hence the plant) is of crucial importance in the control problem definition. Therefore, we choose input and output scaling of the plant, so that at the loop breaking point we considered closed-loop functions (our design objectives), the gain of closed-loop functions are less dependent on the direction of the inputs (at this loop breaking point). In other words, the maximum and minimum singular value are closer to each other at each frequency (principal gains). This leads to the following:

1. Rule of thumb 1: scale the plant to 0 dB at the intended crossover frequency
2. Rule of thumb 2: apply input and output scaling, D, D^{-1} where D is diagonal, so that the cross terms from the input to the output (per diagonal term) have approximately equal size.

In the next section the various steps will be illustrated for a motion control design of an AFM.

27.5 Control Design for a Metrological AFM

Metrological AFMs are used to characterize transfer standards for commercial AFMs to be calibrated. The metrological AFM, shown in Figure 27.5, consists of a high-precision three DOF stage, a Topometrix AFM head and a ZYGO interferometer to measure the stage position in all DOFs. The stage, which carries the sample, has to perform a scanning trajectory in the (x, y)-plane with high fidelity. The stage is driven by three piezo stack actuators through a flexure mechanism in a range of 100 μm in the scanning (x, y)-plane and 20 μm in the imaging z-direction. The deflection of the cantilever in the AFM head is measured using a laser and photodetector. The mirrors and lasers of the interferometer are exactly aligned to the tip of the cantilever, thus minimizing Abbe errors. The measurements of the ZYGO interferometer in all DOFs are traceable to the standard of length. The resolution and root-mean-square (rms) values of the standstill noise with decoupled piezo actuators are given for all sensors in Table 27.1.

The metrological AFM is controlled in feedback, with as input the voltages to the piezo stack actuators u_i (V), $i \in \{x, y, z\}$ in all DOFs and as output the ZYGO interferometers in the scanning x, y-directions and the output of the photodetector in the imaging z-direction, as shown in Figure 27.6. The tip of the cantilever is controlled in constant force mode while the sample is moved relative to the cantilever by the stage in all three DOFs. The output of the ZYGO interferometer in z-direction is used to measure the sample topography directly. The MIMO system can be written as

$$\begin{bmatrix} x \\ y \\ z_t \end{bmatrix} = \begin{bmatrix} G_{xx} & G_{xy} & G_{xz} \\ G_{yx} & G_{yy} & G_{yz} \\ G_{zx} & G_{zy} & G_{zz} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}. \tag{27.28}$$

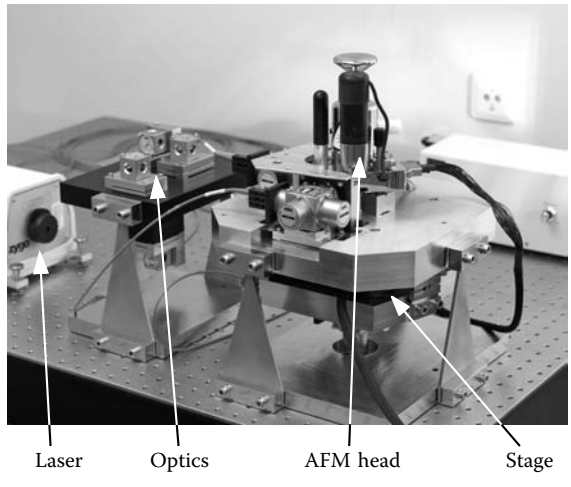

FIGURE 27.5 The metrological AFM.

TABLE 27.1 Resolution and rms Values of the Noise for the Different Sensors

Sensor	Resolution (nm)	Noise rms Value (nm)
ZYGO x	0.15	3.56
ZYGO y	0.15	3.06
ZYGO z	0.15	1.25
Head z_t	0.05	0.14

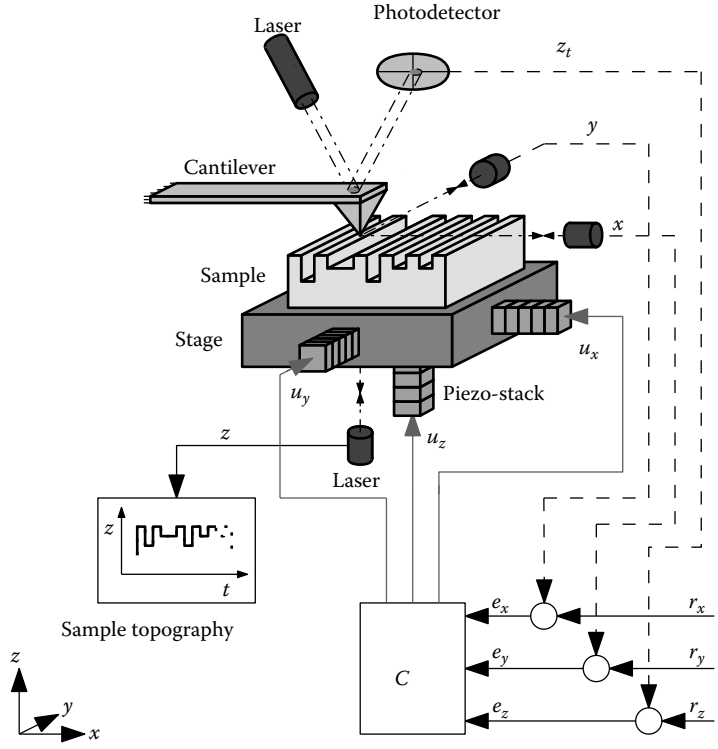

FIGURE 27.6 Schematic representation of the metrological AFM and the feedback control architecture.

27.5.1 Nonparametric Identification

FRFs are key in the design of high-performance motion systems. With a low-fidelity decentralized controller, which tuned in the time domain, the plant can be identified in closed-loop. For this, each input $u = [u_x, u_y, u_z]^T$ is excited independently with a zero-mean white-noise signal. In this way, during each experiment the columns can be filled using the transfer function estimates of the input sensitivity

$$S_i(j\omega) = (I + K(j\omega)G(j\omega))^{-1} \tag{27.29}$$

and the process sensitivity

$$\begin{aligned} PS(j\omega) &= (I + G(j\omega)K(j\omega))^{-1}G(j\omega) \\ &= G(j\omega)(I + K(j\omega)G(j\omega))^{-1}, \end{aligned} \tag{27.30}$$

where the push-through rule [16] is used for proving the equivalence of Equation 27.30. The FRF of the multivariable plant G can be determined as

$$G(j\omega) = PS(j\omega)S_i(j\omega)^{-1}. \tag{27.31}$$

Depending on the quality of the obtained FRF (Equation 27.31), above identification can be repeated with a new controller, designed using the firstly obtained plant FRF. Since the closed-loop function gain with respect to disturbances and noise will be high around the frequencies of the bandwidth, the new identification controller can be designed to obtain a good measurement quality in the frequency range of interest.

27.5.2 Scaling

Scaling and conditioning of the plant are crucial in (norm-based) MIMO controller synthesis. First the plant is scaled to 0 dB at the intended crossover frequencies of 15 Hz in each DOF. With the scaling matrix $T_u = \text{diag}\{1.68 \times 10^{-4}, 2.11 \times 10^{-4}, 1.48 \times 10^{-2}\}$, the scaled system equals $G_T = GT_u$. The scaled system FRF $G_T(j\omega)$ is shown in Figure 27.7 by the dashed gray lines.

The off diagonal terms of the FRF of G_T in Figure 27.7 have different gains in the input and output in z-direction. To improve the conditioning an input–output scaling is applied. Since the system is only of dimension 3×3, the scaling matrix is determined manually as

$$D = \text{diag}\{1, 1, \alpha\}, \tag{27.32}$$

where $\alpha = -35$ dB. For larger systems, the input–output scaling can be obtained using the D-scalings from the SSV calculation [19]. Finally, the scaled system with input–output scaling equals

$$G = DG_T D^{-1}. \tag{27.33}$$

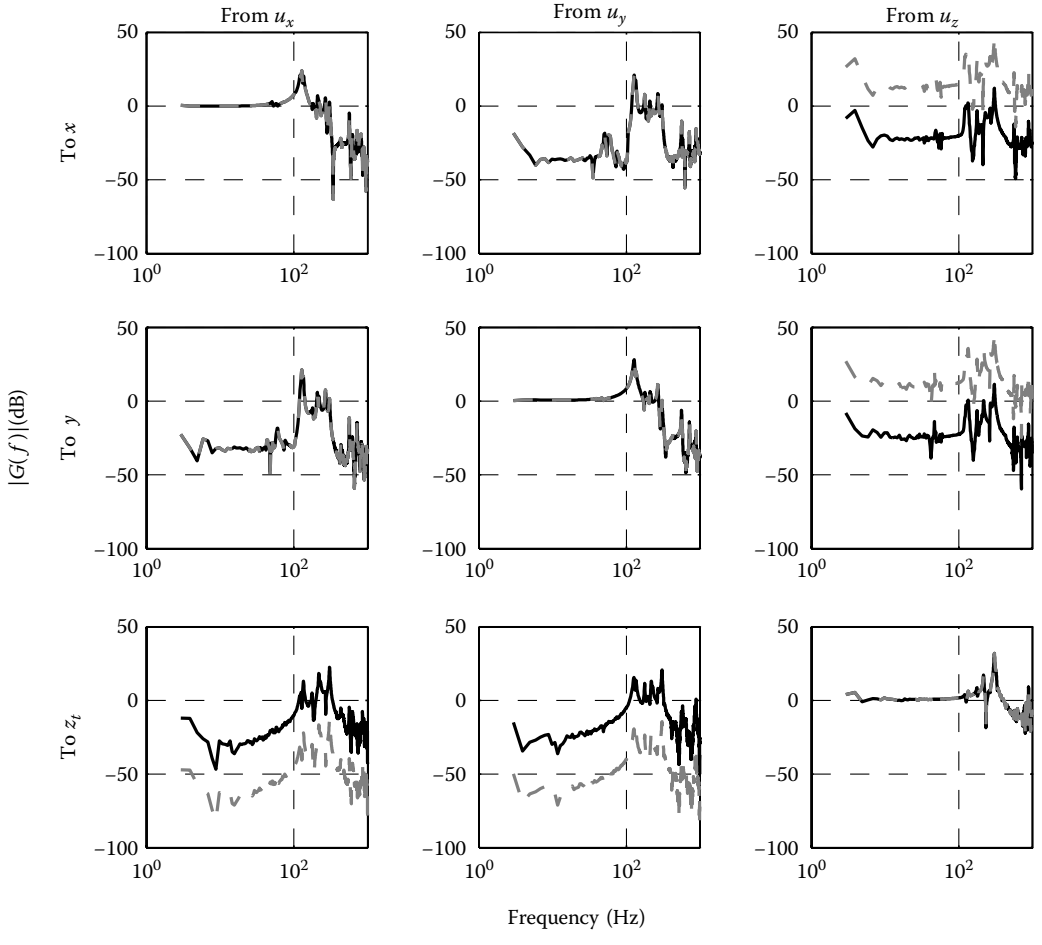

FIGURE 27.7 Frequency response functions of the scaled systems G_T (dashed, gray) and G (solid, black).

The FRF of the system G is shown in Figure 27.7 by the black solid lines. It can be seen that the diagonal terms at the intended bandwidth of 15 Hz have an amplitude of 0 dB and that all off-diagonal terms have approximately equal gains at low frequencies. For the remainder of this example, we define the scaled system with input–output scaling G of Equation 27.33 as the plant to be controlled. The open-loop function at the output of the plant is defined for a controller K as

$$L = GK. \tag{27.34}$$

27.5.3 Interaction Analysis

For the scaling the RGA and SSV interaction measures can be used, as discussed in Section 1.4.3. Since the interaction measures are scaling independent, G and G_T give the same results.

To assess the interaction between the different axes of the metrological AFM, the frequency-dependent RGA [16,18] is calculated as in Equation 27.20. The rows and columns of the RGA sum to 1 for all frequencies f (Hz). If $(\mathrm{RGA})(j\omega) = I$, $\forall\omega$, perfect decoupling is achieved. The RGA for the FRF G of Figure 27.7 is shown in Figure 27.8. For frequencies $f < 100$ Hz, $(f = 2\pi j\omega)$ there is little two-way (bilateral) interaction in the plant.

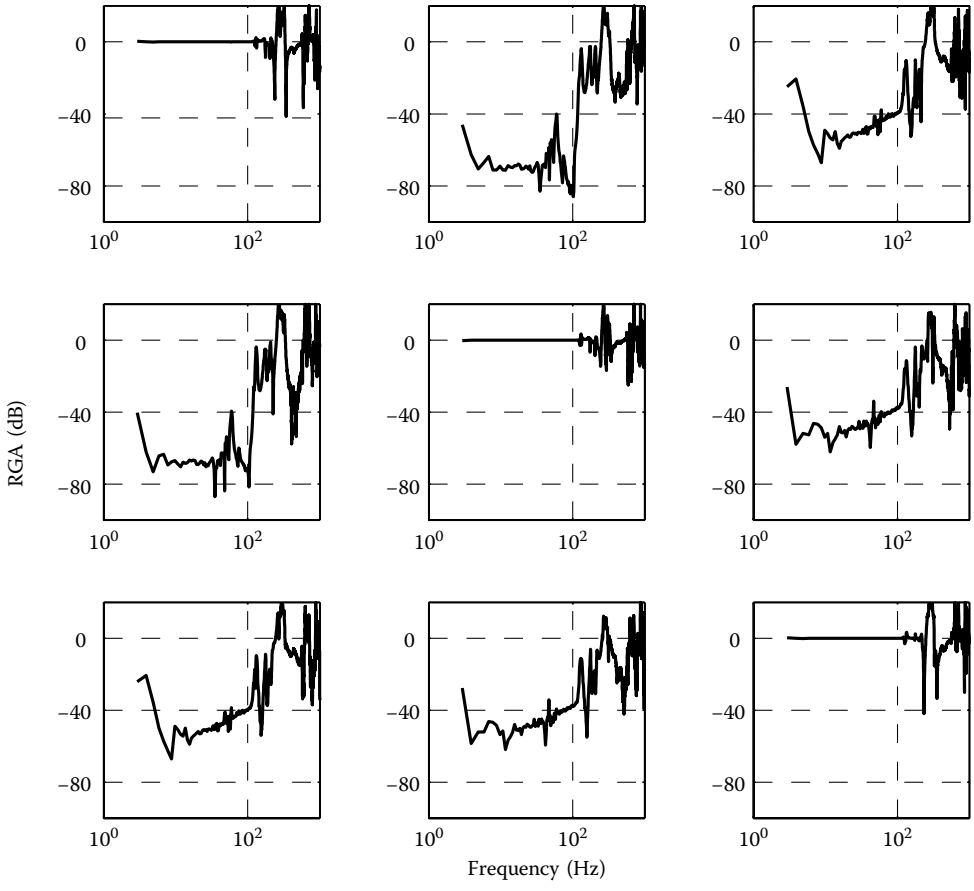

FIGURE 27.8 RGA (Equation 27.20) of the system G of Figure 27.7.

27.5.4 Decoupling Transformations

The system is very well decoupled by design at frequencies up to 50 Hz, as can be seen in the RGA of Figure 27.8. On a physical basis, we expect a transfer function matrix with the structure of

$$\tilde{G}(s) = \sum_{i=1}^{N} \frac{c_i b_i^T}{s^2 + 2\zeta_i \omega_i s + \omega_i^2}.$$

So, at low (zero) frequencies, the system can be modeled as $\sum_{i=1}^{N} c_i b_i^T / \omega_i^2$. The cross terms, up to approximately 50 Hz, can be considered constant. One can use different matrix diagonalization techniques to find decoupling transformations. If we evaluate the interaction measures on the decoupled plant, we see little difference; the bandwidth is still limited by the resonances at 121 Hz in the x, y-directions.

27.5.5 Independent Control Design

Using SISO loopshaping techniques, feedback controllers are designed in each loop independently as

$$K^i = \text{diag}\{K_x^i, K_y^i, K_z^i\}, \tag{27.35}$$

where

$$K_j^i = \underbrace{\frac{k}{s^2}}_{\text{int.}} \underbrace{\frac{1}{\frac{s^2}{(2\pi f_{j,1})^2} + \frac{2\beta_j}{2\pi f_{j,1}} s + 1}}_{\text{2nd order low pass}} \underbrace{\frac{\frac{s}{2\pi f_{j,2}} + 1}{\frac{s}{2\pi f_{j,3}} + 1}}_{\text{Lead}} \tag{27.36}$$

where $j \in \{x, y, z\}$. The parameters of the controllers for the different axes are contained in Table 27.2. To assess the stability of the controlled MIMO system with the independent controller K^i, the SSV with respect to the diagonal structure of the complementary sensitivity T_d is calculated as shown in Figure 27.9a. From SISO loopshaping, it appears that the bandwidth in z-direction can be placed much higher than in x, y-direction. From Figure 27.9a follows that the sufficient condition (Equation 27.24) is not met, that is, stability cannot be guaranteed.

Without loss of generality, we may place a diagonal frequency-dependent weighting filter W in the sufficient condition for independent decentralized control, which has no effect on the spectral radius condition, that is,

$$\rho\left(E_T(j\omega) W(j\omega) W^{-1}(j\omega) T_d(j\omega)\right) < 1, \ \forall \omega, \tag{27.37}$$

see also [24]. Now the closed-loop is stable if for each frequency holds that

$$\overline{\sigma}(W^{-1} T_d) \leq \mu_{T_d}^{-1}(E_T W). \tag{27.38}$$

The stability criterion with the weighting filter is still a sufficient condition. The weighting filter is only used for analysis, it is not included in the controller. If W is chosen diagonal, one can emphasize the

TABLE 27.2 Parameters of the Independent Controller Design K^i

Axis j	k	$f_{j,1}$ (Hz)	β_j	$f_{j,2}$ (Hz)	$f_{j,3}$ (Hz)
x	7943	150	0.6	12.5	100
y	7943	150	0.6	12.5	100
z	26,607	250	0.6	20	200

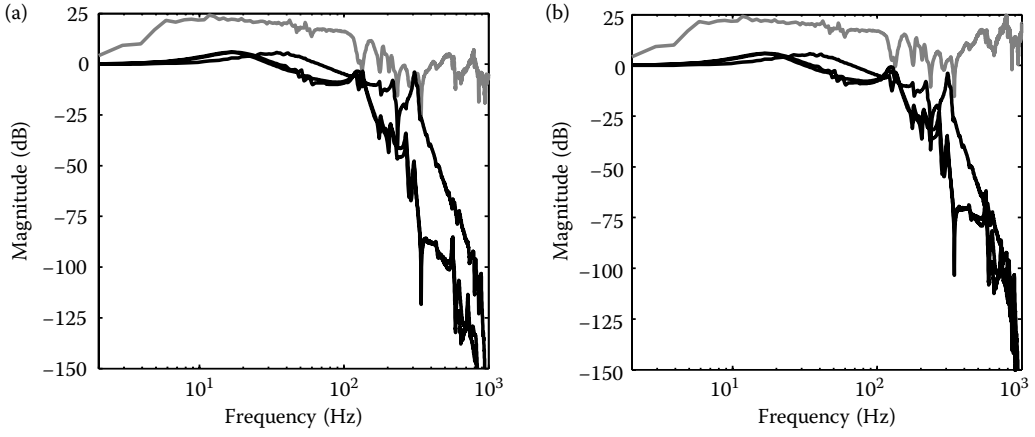

FIGURE 27.9 Sufficient condition for stability with the high-bandwidth decentralized controller. (a) Without scaling: sufficient condition not achieved and (b) with scaling matrix w: stability guaranteed.

contribution of each loop to the maximum singular value of T_d. The complementary sensitivity function in the z-direction rolls off at higher frequencies (50 Hz) than the complementary sensitivity functions of the x, y-axes. Therefore, the contribution of the designs in x,y-directions to $\sigma_i(T_d)$ may be increased at high frequencies. At the same time, the weighting filter W may result in a smaller value of $\mu_{T_d}(E_T W)$. Therefore, for this application, we choose,

$$W(s) = \text{diag}\{w(s), w(s), 1\}, \tag{27.39}$$

$$w(s) = \frac{\omega_w^2}{s^2 + 2\zeta_w \omega_w s + \omega_w^2} \tag{27.40}$$

with $\omega_w = 2\pi 50$ rad/s, $\zeta_w = 0.8$. As shown in Figure 27.9b, the bound due to $\mu_{T_d}(E_T W)$ is reduced at higher frequencies so that the weighted sufficient condition for closed-loop stability is satisfied. The controller design with independent loopshaping is successful, but limited to these crossover frequencies.

27.5.6 Sequential Control Design

The decentralized controller K^i obtained from the independent control design can also be used to analyze stability using sequential loopshaping. We start with the loop where we can achieve the highest bandwidth, at low frequencies, this then has very little influence at the other loops. The next step is to design the x-, respectively, y-direction with the z-loop closed (z, and later x controlled included). All three equivalent open-loops are shown in Figure 27.10. It is visible that each loop is stable and there may be even potential to increase bandwidth slightly iterating the sequential design steps. This illustrates that by only changing the stability analysis, we have now less conservatism with sequential loopshaping. There is still some more design freedom not exploited with independent loopshaping.

27.5.7 Norm-Based Control Design

For the norm-based control design, we formulate the following mixed sensitivity $\mathcal{H}_\infty$ control design:

$$\min_{stab.K} \left\| \begin{bmatrix} W_S S_o \\ W_U K S_o \\ W_E T \end{bmatrix} \right\|_\infty \tag{27.41}$$

For given weighting filters W_S, W_U, and W_E and a parametric model, standard software is available for the controller synthesis. Since the plant FRF is scaled to 0 dB at the intended crossover frequencies, a

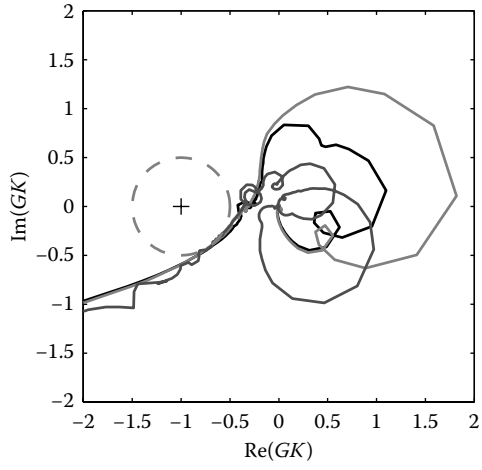

FIGURE 27.10 Equivalent open-loops for the MIMO system with the sequential controller design, x-direction (black), y-direction (light-gray), and z-direction (dark-gray).

(basic) parametric model that approximates the plant up to 50 Hz is

$$G_{model,1}(s) = I. \tag{27.42}$$

The model uncertainty is now defined as

$$E(s) = (G_{model,1}(s) - G(s))G_{model,1}(s)^{-1}. \tag{27.43}$$

As a first step, we can bind the worst case uncertainty per frequency with a scalar weighting filter, $W_E(s)$,

$$W_E(s) > \mu_{Td}(E(s)). \tag{27.44}$$

To assure stability of the true system, this weighting filter is the upper bound of the complementary sensitivity function, according to Equation 27.24.

With this simple model, the crossover frequencies are limited.
An improved model approximates each DOF with a second order model:

$$G_{model,2} = \text{diag}\left\{\frac{1}{s^2 + 2\zeta_i\omega_i + \omega_i^2}\right\}, \quad i = \{1, 2, 3\} \tag{27.45}$$

where ζ_i, ω_i are chosen to match the resonance frequencies of the first modes in all three channels. Although the model is a sixth-order model, we will call it in the sequel the second-order model, because of its basic structure. Again, the uncertainty due to these (new) model simplifications are put in a weighting filter on the complementary sensitivity function. Iterative redesign of the performance weighting filters then results in the sensitivity function shown in Figure 27.11. It is clear that higher crossover frequencies are achievable with the second-order model and the iterative redesign of the performance weighting filters, as shown by the sensitivity functions in Figure 27.11, where the $\mathcal{H}_\infty$ design with the unity model is shown by the light gray line and the $\mathcal{H}_\infty$ design with the second-order model by the black line, respectively.

From here on, more refinements can be made in modeling the (low)-frequency contribution of the cross terms, by compensating/modeling the time delay due to sampling. In addition, a more refined analysis can take into account the structure (direction dependency) of the interaction modeled as uncertainty. A control method that is capable of this is μ-synthesis.

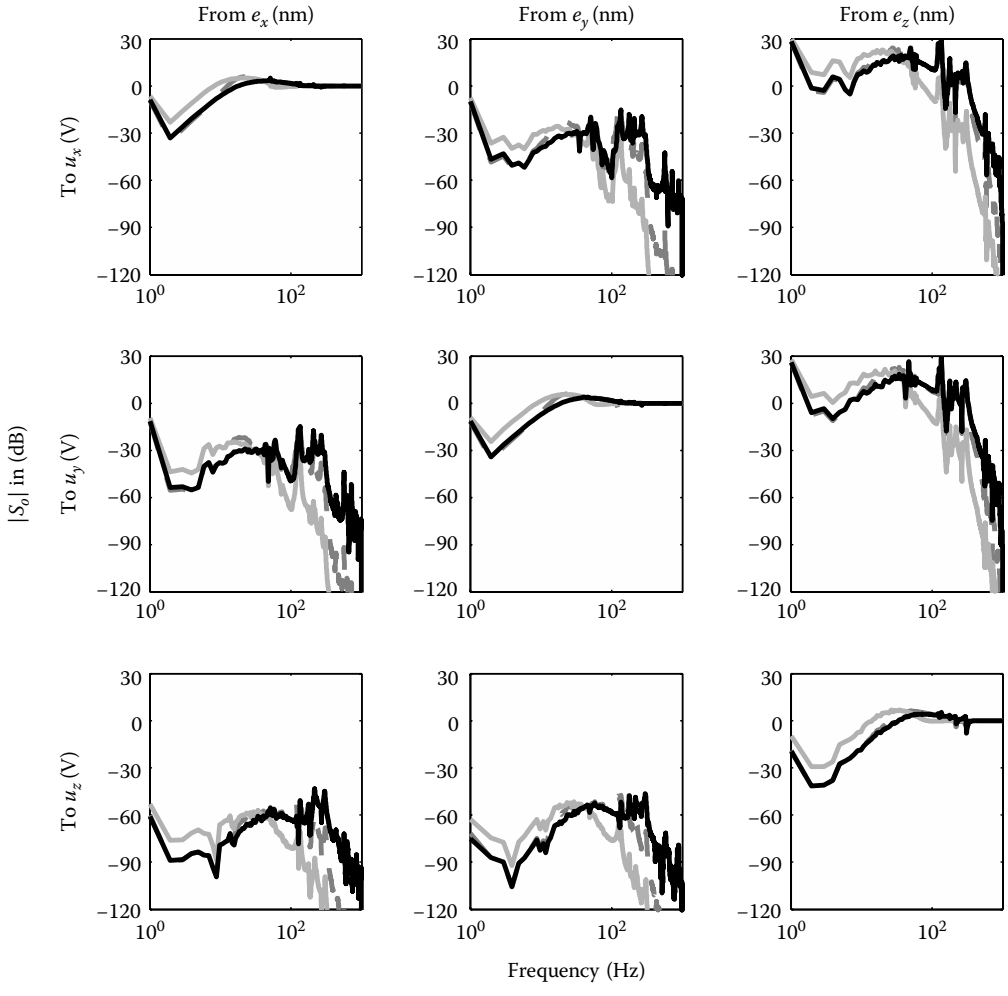

FIGURE 27.11 Output sensitivity $S_o = (I + GK)^{-1}$ obtained with the independent/sequential controller design (dark gray, dashed), the $\mathcal{H}_\infty$ controller design with identity model (light gray, solid), and the $\mathcal{H}_\infty$ controller design with the second-order model (black, solid).

Also, different control design formulations can be used. If the frequency content of disturbances is modeled, these models can be included in the weighting filters. Simple disturbance models consider disturbances in each channel independently, like in SISO control [1]. More advanced models take directions and correlations into account [25]. The complexity of the design process increases gradually with inclusion of each physical model (increase of order of the augmented plant). As a result of the norm-based synthesis techniques, the controller order increases, so that at some point control relevant model reduction techniques become necessary for implementation.

The stability of the metrological AFM with the $\mathcal{H}_\infty$ controllers designed with the unity model and the second-order model is assessed by evaluating the characteristic loci $\lambda(GK^I_{\mathcal{H}_\infty})$ and $\lambda(GK_{\mathcal{H}_\infty})$, respectively. The characteristic loci, shown in Figure 27.12, indicate that both MIMO controllers stabilize the system. Since the controller $K_{\mathcal{H}_\infty}$ obtained with the second-order model has larger crossover frequencies, the high-frequency resonances become more apparent, which is visible by the circles in Figure 27.12b.

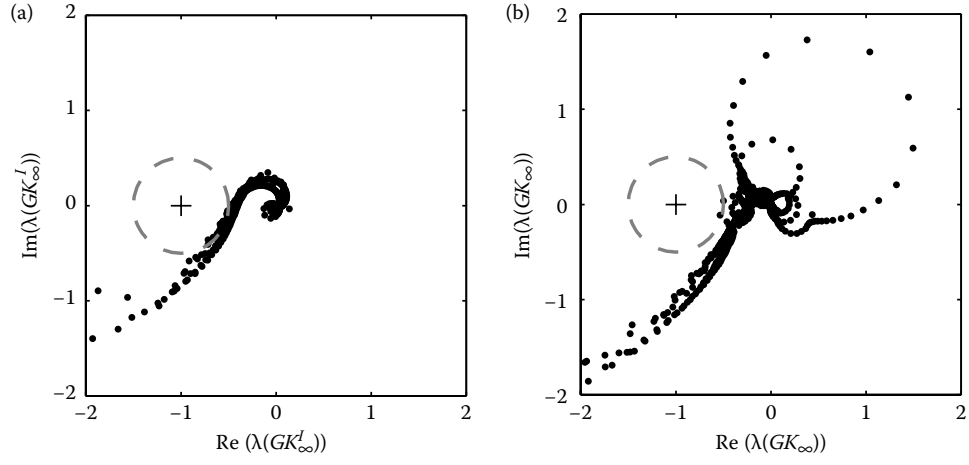

FIGURE 27.12 Characteristic loci of the MIMO $\mathcal{H}_\infty$ control designs with the unity model and the second order model. (a) Unity model and (b) Second-order model.

Finally, for all three designs to be implemented, the scaling and decoupling matrices are placed in the controller as

$$K_{imp}(s) = T_u T_{scale,input} K(s) T_{scale,output}. \tag{27.46}$$

27.5.8 Experimental Results

The output sensitivities S_o obtained with the independent controller design K^i and with the norm-based controller designs, $K^I_{\mathcal{H}_\infty}$ with the unity model and $K_{\mathcal{H}_\infty}$ with the second model, are shown in Figure 27.11. The lowest crossover frequency is obtained with the $\mathcal{H}_\infty$ controller design with the unity model, which allows for simple modeling and controller synthesis steps, but at the cost of reduced achievable crossover frequencies due to the larger model uncertainty. With the $\mathcal{H}_\infty$ controller design with the second order model slightly higher crossover frequencies are achieved than with the independent controller design. At low frequencies the off-diagonal terms in the output sensitivity S_o are reduced by both $\mathcal{H}_\infty$ controllers compared to the independent design. However, at higher frequencies $f > 50$ Hz, a better suppression of the output disturbances is obtained with the independent controller design.

The three designed controllers are tested by means of experiments on the metrological AFM. For the experiments a constant velocity setpoint of 125 nm/s is used in the slow scanning x-direction. In the fast scanning y-direction a triangular shaped setpoint profile over a range of ± 25 μm with a velocity of 25 μm/s is used, that is, with a period-time of 4 s. The z-direction is controlled to a constant tip deflection. The controller sampling frequency for the experiments equals $f_s = 2$ kHz.

The measured sample topographies of the ZYGO interferometer in z-direction for the experiments with all three controllers are shown in Figure 27.13 with the corresponding tracking errors of the control loop in z-direction (using the photodetector). The small time shift between the measured sample topographies is due to the absence of an absolute homing of the stage position. A clear difference between the measured sample topographies by the three controllers can be seen at the time instants where the transitions in the sample topography occur. A larger overshoot is obtained with the controller $K^I_{\mathcal{H}_\infty}$ due to the lower crossover frequencies. The smallest overshoot is obtained with the $\mathcal{H}_\infty$ controller $K_{\mathcal{H}_\infty}$ designed with the second-order model. The root-mean-square (rms) values of the tracking errors from the three experiments equal $\text{rms}(e_{z_{K^i}}) = 9.69$ nm, $\text{rms}(e_{z_{K^I_{\mathcal{H}_\infty}}}) = 19.51$ nm and $\text{rms}(e_{z_{K_{\mathcal{H}_\infty}}}) = 8.40$ nm, respectively. The cumulative power spectral densities (CPSDs), shown in the right axis of Figure 27.13, show the

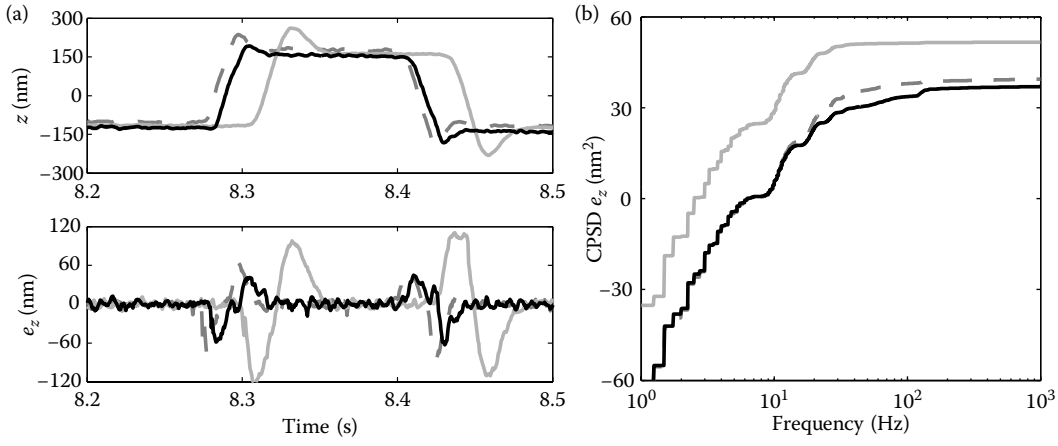

FIGURE 27.13 Measured sample topographies, errors and CPSDs of the errors of the experiments with the independent/sequential controller design (dark gray, dashed), the $\mathcal{H}_\infty$ controller design with identity model (light gray, solid), and the $\mathcal{H}_\infty$ controller design with the second-order model (black, solid).

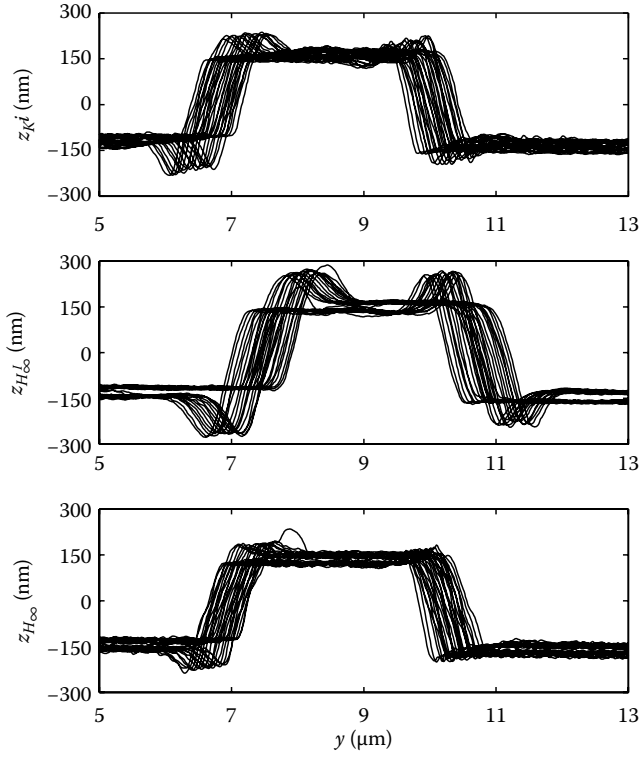

FIGURE 27.14 Measured sample topography with the independent/sequential controller design (top axis), the $\mathcal{H}_\infty$ controller design with identity model (middle axis), and the $\mathcal{H}_\infty$ controller design with the second-order model (bottom axis).

differences in the tracking errors and the slight improvement obtained with $K_{\mathcal{H}_\infty}$ compared to K^i. For frequency $f \to \infty$, the CPSDs converge to the squared rms values of the tracking errors.

The measured sample topographies of the experiments with the different controllers are plotted versus the measured y position in Figure 27.14. All three experiments show a sample topography with a line of approximately 1.5 μm wide and 300 nm high. However, the differences in overshoot by the different controllers is clearly visible in the measured topographies. The best performance is obtained with the norm-based controller $K_{\mathcal{H}_\infty}$. The phase shift between the transitions in the sample topographies indicates that the sample is slightly rotated under the AFM with respect to the scanning y direction, that is, the sample is rotated on the stage around the z-axis [26].

Figure 27.15 shows the tracking errors of all three axes as obtained during the experiments with all three controllers. At approximately 4.3 s a transition in the sample topography occurs. At this time instant a clear oscillation in the tracking error e_z in z-direction can be seen. The largest error e_z is obtained with the $\mathcal{H}_\infty$ controller $K^I_{\mathcal{H}_\infty}$ (unity model) due to the lower crossover frequencies. The controllers K^i and $K_{\mathcal{H}_\infty}$ reduce the error e_z significantly. A slightly better performance is obtained with the norm-based controller design with the second-order model $K_{\mathcal{H}_\infty}$ compared to the independent controller design K^i. At the time instants of the transition in the sample topography a clear disturbance in the tracking error e_x, that is, in the slow scanning x-direction, is visible and also slightly in e_y of the fast scanning y-direction. These errors are caused by the coupling between the axes.

At 5 s a turnaround point is present in the fast scanning y-direction, which leads to an increase in the corresponding tracking error e_y as can be seen in Figure 27.15. The smallest error is obtained with

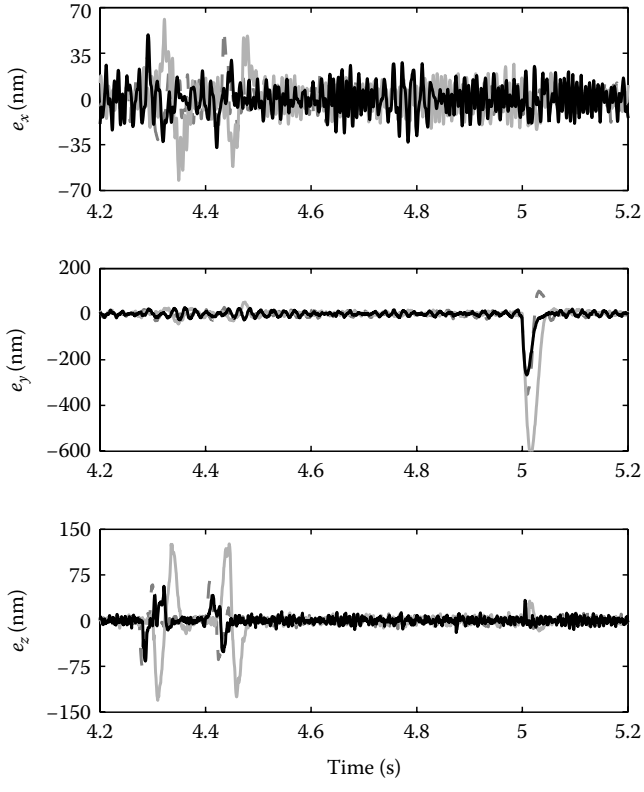

FIGURE 27.15 Errors in all DOFs of the metrological AFM of the experiments with the independent/sequential controller design (dark gray, dashed), the $\mathcal{H}_\infty$ controller design with identity model (light gray, solid), and the $\mathcal{H}_\infty$ controller design with the second-order model (black, solid).

TABLE 27.3 Errors of all Axes for the Different Controller Designs

Error	K^i	$K^I_{\mathcal{H}_\infty}$	$K_{\mathcal{H}_\infty}$
rms(e_x) (nm)	9.10	11.23	10.06
rms(e_y) (nm)	29.00	57.03	22.63
rms(e_z) (nm)	9.69	19.51	8.40

the norm-based controller $K_{\mathcal{H}_\infty}$, followed by the independent design K^i and finally the largest error is obtained with the norm-based controller $K^I_{\mathcal{H}_\infty}$. The differences in error are caused by the differences in the achieved crossover frequencies with the different controllers. At the time instant of the turnaround point (5 s) also a small error increase is visible in e_z, which is caused by coupling between the axes. In x-direction the coupling effects are not visible since the error increase does not exceed the noise bound of this axis.

The rms values of the errors of all axes are shown in Table 27.3 for all controller designs.

The CPSDs of the different errors of Figure 27.15 are shown in Figure 27.16. For frequency $f \to \infty$ the different CPSDs converge to the squared rms values of the error. It can be seen that in all three axes the largest error is obtained with the norm-based design with the unity model, that is, with $K^I_{\mathcal{H}_\infty}$. In x-direction a slightly larger tracking error is obtained with the $\mathcal{H}_\infty$ design $K_{\mathcal{H}_\infty}$ compared to the independently designed K^i. For the y- and z-direction the norm-based controller design with the second-order model $K_{\mathcal{H}_\infty}$ outperforms the independent designed K^i.

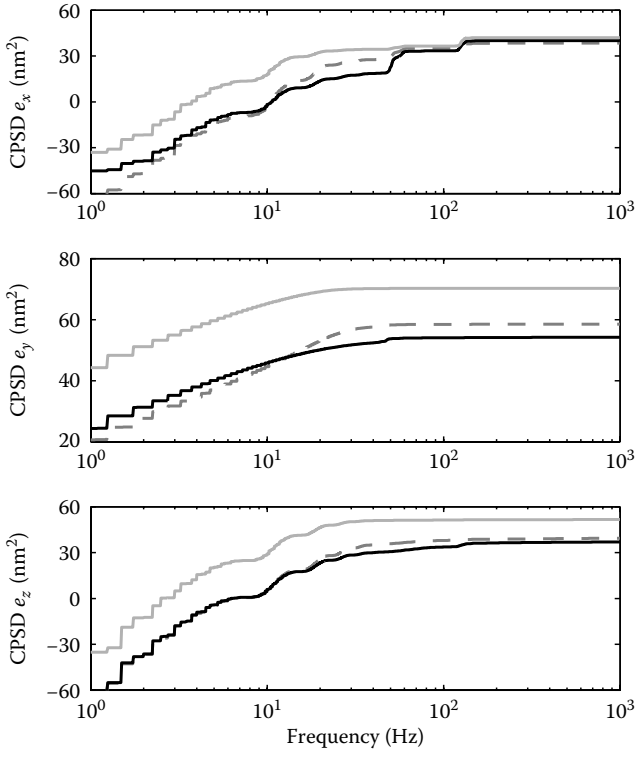

FIGURE 27.16 PSDs of the errors of Figure 27.15; independent/sequential controller design (dark gray, dashed), $\mathcal{H}_\infty$ controller design with identity model (light gray, solid), and $\mathcal{H}_\infty$ controller design with the second-order model (black, solid).

27.6 Conclusion

In this work we have shown typical control design tools, methods and industrial tips and tricks for motion systems. A practical step-by-step procedure is proposed in order to make use of well-known industrial practice of SISO loopshaping, while accounting for possible interaction in MIMO motion systems. In addition to the feedback design, for motion the use of feedforward is crucial to realize the performance.

The fast amount of literature on model-based design hardly provides the means for engineers in industry to successfully apply modern tools in practice. This chapter hopefully, bridges a gap between classical loopshaping and MIMO control. Further developments are foreseen in the area of disturbance modeling, because the notion of directionality is hardly used in MIMO shaping, as well as data-based control, in which the FRF can directly be used to do norm-based control. Finally, the area of robust control oriented closed-loop identification is a necessary way to go if high-performance motion systems emerge and in particular if nonsquare plant are used, such as is the case with overactuation.

References

1. M. Steinbuch and M. L. Norg, Advanced motion control: An industrial perspective, *European Journal of Control*, vol. 4, no. 4, pp. 278–293, 1998.
2. P. Lambrechts, M. Boerlage, and M. Steinbuch, Trajectory planning and feedforward design for electromechanical motion systems, *Control Engineering Practice*, vol. 13, no. 2, pp. 145–157, 2005.
3. M. Boerlage, R. Tousain, and M. Steinbuch, Jerk derivative feedforward control for motion systems, in *Proceedings of the American Control Conference*, pp. 4843–4848, 2004.
4. R. Craig and A. Kurdila, *Fundamentals of Structral Dynamics*, 2nd ed. John Wiley & Sons, New York, 2006.
5. W. Gawronski, *Advanced Structral Dynamics and Active Control of Structures*. Springer-Verlag, Berlin, 2004.
6. S. Moheimani, D. Halim, and A. Fleming, *Spatial Control of Vibration: Theory and Experiments*. World Scientific, Singapore, 2003.
7. M. Boerlage, An exploratory study on multivariable control for motion systems, Master's thesis, Eindhoven University of Technology, 2004.
8. M. van de Wal, G. van Baars, F. Sperling, and O. Bosgra, Multivariable $\mathcal{H}_\infty, \mu$ feedback control design for high-precision wafer stage motion, *Control Engineering Practice*, vol. 10, no. 7, pp. 739–755, 2002.
9. T. Oomen, O. Bosgra, and M. Van de Wal, Identification for robust inferential control, in *Proceedings of the Conference on Decision and Control*, pp. 2581–2586, 2009.
10. L. Ljung, *System Identification: Theory for the User*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 1999.
11. R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach*. IEEE Press, New York, 2001.
12. I. Horowitz, Survey of quantitative feedback theory (QFT), *International Journal of Control*, vol. 53, pp. 255–291, 1991.
13. O. Yaniv, *Quantitative Feedback Design of Linear and Nonlinear Control Systems*. The Springer International Series in Engineering and Computer Science, vol. 509, 1999.
14. M. Garcia-Sanz and I. Egaña, Quantitative non-diagonal controller design for multivariable systems with uncertainty, *International Journal of Robust Nonlinear Control*, vol. 12, pp. 321–333, 2002.
15. M. L. G. Boerlage, A. G. De Jager, and M. Steinbuch, Control relevant blind identification of disturbances, *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 393–404, 2009.
16. S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control, Analysis and Design*, 2nd ed. John Wiley & Sons, New York, 2005.
17. H. Rosenbrock, *Computer-Aided Control System Design*. Academic Press, Orlando, 1974.
18. E. Bristol, On a new measure of interaction for multivariable process control, *IEEE Transactions on Automatic Control*, vol. 11, no. 1, pp. 133–134, 1966.
19. K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice-Hall, Englewood Cliffs, NJ, 1996.
20. P. Grosdidier and M. Morari, Interaction measures for systems under decentralized control, *Automatica*, vol. 22, pp. 309–319, 1986.

21. D. Mayne, Sequential design of linear multivariable systems, in *Proceedings of the IEE*, vol. 126, pp. 568–572, 1979.
22. J. Freudenberg and D. Looze, *Frequency Domain Properties of Scalar and Multivariable Feedback Systems*, Lecture Notes in Control and Information Sciences 104, M. Thoma and A. Wyner, Eds. Springer-Verlag, Berlin, 1988.
23. A. Den Hamer, S. Weiland, and M. Steinbuch, Model-free norm-based fixed structure controller synthesis, in *Proceedings of the Conference on Decision and Control*, pp. 4030–4035, 2009.
24. M. Morari and E. Zafiriou, *Robust Process Control*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
25. M. L. G. Boerlage, Rejection of disturbances in multivariable motion systems, PhD thesis, Eindhoven University of Technology, September 2008.
26. R. J. E. Merry, Performance-driven model-based control for nano-motion systems, PhD thesis, Eindhoven University of Technology, November 2009.

28

Color Controls: An Advanced Feedback System

Lalit K. Mestha
Xerox Research Center

Alvaro E. Gil
Xerox Research Center

28.1 Introduction

While there is a large body of literature [1] written on the application of controls to continous feed offset printing, in this chapter we focus on cut-sheet digital production printers. For color digital production printers, the control challenges are several orders of magnitude larger than those of black and white, because they have to compete in the traditional continuous feed offset market, which demands high quality at low cost. This chapter is written to give a glimpse of control systems used in these color systems. We cover the design of complex algorithms by focusing our attention on system-level color control loops and state-of-the-art algorithms, such as multi-input multi-output (MIMO) state feedback (SF) with pole placement design, optimal controls, model predictive controls, and cooperative controls. These algorithms operate on both digital and process actuators to produce high-quality prints, help to reduce service cost, and provide a quick turn-around time. At the end of this chapter, we introduce some opportunities for controls to further advance the system-level design.

28.1.1 Color Control Needs

The main differences between the digital and traditional offset printing technologies are the setup cost and the ability to print variable data on demand; the former representing the main economic gap between these technologies. The market decides between digital and offset technologies, based on variety of factors, namely the performance in color accuracy and consistency, and the complexity and efficiency of the associated workflow.

Workflow is a commonly used term that describes the various steps required all the way from receiving the orders in a typical print shop to the production of a print job in finished form. It includes not only the document creation and actual production steps, but also all the necessary supporting tasks. Within the workflow, key steps include the document creation, viewing, and rendering, which are frequently called on for any number of devices (e.g., displays, printers, scanners, and digital cameras) with similar or differing technologies. This makes the job of color management and control more complex. At a fundamental level, the color of the prints from any device should match the color that the user requested in their documents. There are numerous phrases used to describe the requested color according to industry parlance. They are target/desired/aim color and so on. To be able to reproduce the requested color, devices should be capable of rendering color documents accurately so that the difference between the requested and reproduced color is within the limits of human visual perception called just noticeable difference (JND). Even if the process can produce stable color, the device may not be able to render color accurately. All the color management and control technologies associated with manipulation of the image and the digital values (at the system level) are aimed at producing accurate color. At a device level, there are process control technologies that try to maintain the color on a print-to-print, job-to-job, machine-to-machine, or device-to-device basis. Often, the color management and control technologies are blended to achieve a common goal—high color accuracy and consistency. Thus, the purpose of color control is clear: to make the color accurate and consistent across multiple prints and multiple devices, across a wide variety of distributed and interconnected workflows.

28.2 System Overview

The printing stage normally involves: (1) a digital front end (DFE), and (2) a print engine. A detailed description of the system can be found in [1]. Unlike the workstation, where processing by the user may be independent of the print engine, a DFE or a network of DFEs, which may be from multiple vendors, are used to convert the electronic job to *CMYK* (cyan, magenta, yellow, and black) form through a series of image processing applications such as trapping, segmentation, rasterization, color management and control, image resolution enhancement, and antialiasing. The resulting *CMYK* is specifically designed and optimized for the associated color digital printing system. Multidimensional, industry standard source profiles are used to transform images to device-independent form, for example, $L^*a^*b^*$, or SWOP (Standard Web Offset Printing) files. L^* defines lightness, a^* corresponds to the red/green value, and b^* denotes the amount of yellow/blue, which corresponds to the way the human eye perceives color. A neutral color is a color for which $a^* = b^* = 0$. In some cases, the transformation may be directly between device-specific forms to printer-specific form. As such, the input document is transformed from its PDL (Page Description Language) format such as PS (Post Script), PDF (Portable Document Format), or TIFF (Tagged Image File Format) to *CMYK* color separations to be printed by the engine. For PS images, this is done by first utilizing an interpreter, for example, a PS interpreter, to identify the commands found in the PDL. An imaging module then generates a rasterized format of the PDL document at the correct print engine resolution, for example, 600 dpi. The above is usually referred to as raster image processing (RIP). During the RIP, color profiles, for example, ICC (International Color Consortium) comprising of multidimensional lookup tables (LUT) are applied, which transform the color from *RGB* to *CMYK* separations with $L^*a^*b^*$ as an internal device-independent space, which the user does not see. Control functions required to optimize the color accuracy are typically applied inside the DFE, in particular inside the multidimensional profiles.

When it comes to the print engine, in a typical EP (electro-photographic) printing process, the material state, that is, state of toner and EP process, affects the print quality and stability of color. Color is also affected by the type of media stocks (e.g., coated, uncoated, textured, smooth, and speciality), sheet-to-sheet differences, temperature and humidity of the environment, process and material aging, and wear in drives etc. Print engine architectures used for imaging four color separations [2], and the

basic steps of the process, for example, in the EP process, charge/recharge, expose, develop, transfer, fuse, and clean, contribute a varying degree of complexity to controlling the stability of color. Tighter requirements for image registration between separations for simplex and duplex printing and varying paper motions at various regions in the paper path make the process controls even more difficult.

28.2.1 Process Physics

The EP printing process is a unique discipline that incorporates domain-specific concepts from the physical sciences and engineering. There are at least six fundamental steps to monochrome digital printing based on EP process that are available in the literature. Color digital printing, which may be carried out in several different architectures, involves different combinations of these steps. One of those architectures is illustrated in Figure 28.1. As shown, the photoconductor is charged in the charging station, and the electrostatic latent image is formed on the photoconductor in the exposure station. Then, in the next step, the latent image is rendered into a real visible image on the photoconductor in the development station using an electrostatically charged toner cloud. The developed latent image is transferred to the media in a transfer station. The toner particles of the transferred image are fused to the media via heat and pressure in the fusing station. Finally, the residual toner on the photoconductor is removed in the cleaning station. While each of the six stations of the EP process is critical to digital printing, we generally ignore the cleaning station when modeling the process physics for control purposes.

28.2.2 Models

Developing a complete model of the printer for the purpose of color control is very difficult. There are two fundamental approaches to color modeling: (1) empirical or interpolation-based approaches that treat the device as a black box and (2) analytical or first principle approaches that attempt to characterize the device color response using analytical functions that have physical connection to the process. Both approaches are capable of predicting the color response of the device for a variety of input images.

Empirical or interpolation-based methods are generally measurement intensive and require the use of a large set of experimentally generated input and output data. These models may contain LUTs or parameterized analytical functions that fit the data.

Accurate first-principle models (so-called "white box" models) are not available for all kinds of imaging devices. The complexity of the models, errors in capturing the actual physical process in the presence of

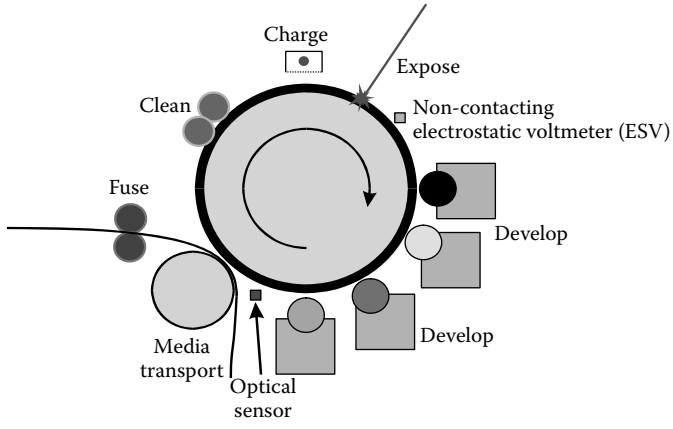

FIGURE 28.1 A typical electrophotographic print engine with a drum photoconductor.

device drift over time, light-scattering effects, and many other uncertainties associated with the physical device and media make it impossible to accurately model the device in a reasonable period of time. In [1] (Chapter 10), a more elaborate, parameterized nonlinear spectral model of the printing system that incorporates reasonable abstractions of the process is described. This can help us inject meaningful timevarying effects into the system. The five key EP process steps are modeled as nonlinear localized transfer functions (LTF) with the actuators as the inputs, and sensed or measured parameters as the outputs. The LTF models capture only the local aspect of the color "dot printer." The spatial aspects (i.e., dot growth, dot spread, edge enhancement) of these processes are then captured using modulation transfer function (MTF) models. With this sequential approach, we first develop the underlying physics of the processes used for modeling a colored dot that is fundamental to the creation of digital images for color process. The process models are then cascaded in sequence, where the output of one process model becomes the input to the next one. The dot spread is then modeled using a halftoning strategy and MTFs of the key segments of the EP process. These models are used for designing feedback controllers for each of the major subsystems, such as controllers for generating multidimensional profiles, to understand their interactions, manage the complexity of the system through careful design of control loops, and achieve the overall system objective.

Controllers for digital color printing are typically designed using an empirical approach. Since our focus is on the design of control algorithms, we have limited the use of process models for the validation of the controller performance on a nonlinear printing system.

28.3 Color Controls: A Modern Feedback Approach

To understand the control approach for reproducing color accurately and consistently, let us take a simple case of spot color reproduction.

28.3.1 Spot Color Control

A spot color is any color generated by the *CMYK* colorants when printed on paper. Spot colors can purely be one separation or mixed (known as spot color emulation). Marketing collaterals, direct mail, catalogs, business cards, and design documents are applications where spot colors are used.

Some printer companies that produce spot colors use a manual approach to adjust the device *CMYK* separations prior to RIP. For example, the document creator may select a Pantone® color for application in specific areas through a user interface on a printing device or computer monitor. The Pantone color can be described in a device-independent coordinate space like $L^*a^*b^*$ or the Pantone-provided *CMYK* recipe for a selected printer. A hardcopy sample of the chosen Pantone spot color would normally be available for comparision with a printed spot color. Prior to processing the document in the DFE with the right recipe, the operator has the option of using a spot color editor to manually enter an alternative *CMYK* recipe. The editing function may need to be repeated if the first print does not meet the accuracy goals. It can take several iterations of the print-view-adjust process to find the right *CMYK*.

This kind of manual work process presents various problems, which include operator errors associated with the manual adjustments of the *CMYK* combinations. Accordingly, modifications to the *CMYK* values may result in more variability in the output color. Correct *CMYK* combinations may never be found even after repeated attempts by skilled operators. Below, we present how a control theory approach can be very effective for this type of work process assuming that an inline or offline color measurement device is available.

An automatic color feedback control approach can be more effective than the manual one. In an automatic approach, for the first iteration, the sensor returns a measurement of printed color, that is, $L^*a^*b^*$ values, for some nominal *CMYK* recipe. This recipe can be obtained from a device model or from the Pantone supplied recipe if the spot color is from the Pantone color library. Control algorithms will

compute the next *CMYK* recipe by comparing the measured $L^*a^*b^*$ values to the desired $L^*a^*b^*$ values. These iterations can be continued until a minimum error is reached between the measured and desired color values. There is always a possibility that the iterative control loop will become unstable and the output color will move away from the desired color in successive iterations. However, as we describe next, the iterative process can be made stable via a properly designed control algorithm. For validation, the iterations can be carried out directly on a printer or on the printer model when it accurately represents the printer.

28.3.1.1 State Model

The *CMYK* to $L^*a^*b^*$ printing process is a MIMO dynamic system. For the purpose of spot color control, the dynamics are captured in the print-to-print cycle. Here we present a linear state variable model of the color printing process that can be used for designing stable feedback controllers. Process variation generally occurs at many print cycles, hence, it is incorporated as uncertainities in the system.

For the purpose of controlling a single spot color, let the color system be represented by a black box to show inputs and outputs as shown in Figure 28.2. The vector V represents small deviations in *CMYK*, which is used during iterations. The sum of the nominal *CMYK* values and the vector V corresponds to the spot color recipe used for printing at any given iteration cycle. Measured $L^*a^*b^*$ values are shown by the vector x. As mentioned above, nominal *CMYK* values can be obtained from one of the following methods: (1) using the inverse of a coarse printer model, (2) based on skills and understanding for the desired reference $L^*a^*b^*$ values, or (3) using a previously determined spot color recipe (from a multidimensional profile table or the Pantone color library). Once the nominal *CMYK* values are chosen, the problem of finding the correct *CMYK* values becomes an iterative search for the best V vector by printing, measuring $L^*a^*b^*$ values, comparing the measured with the reference $L^*a^*b^*$ values, and processing the error in a feedback controller to generate the next V.

We need an appropriate error-processing algorithm (or a controller) so that the iterations converge and common closed-loop performance criteria (i.e., fast convergence time—one to two iterations, zero/minimal steady-state errors, no transient overshoot response, low sensitivity to changes in system, large stability bounds etc.) is met.

The design of the error-processing algorithm requires theoretical knowledge of MIMO control systems [3]. Considering the printer input–output characteristics as linear, we first develop a state model for the open-loop system of Figure 28.2. The input–output characteristic is generally true at the nominal *CMYK* values (see Figure 28.3 for the case where we only consider *CMY* values for simplicity). At the nominal *CMYK* inputs, a first-order finite-difference equation (with dependence on the print number) represents the open-loop color system. If k is the print number (more appropriately called the iteration number), then the open-loop system equation for a single spot color is written in terms of the Jacobian—the first-order sensitivity matrix relating the output and input values—which is given by:

$$x(k+1) = BV(k) + x_o \tag{28.1}$$

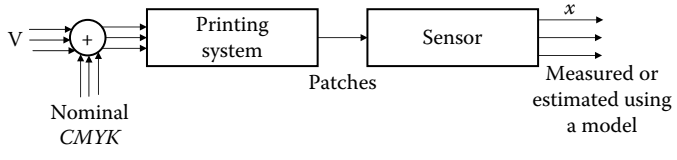

FIGURE 28.2 Diagram representing an open-loop system for a four-color *CMYK* printing system.

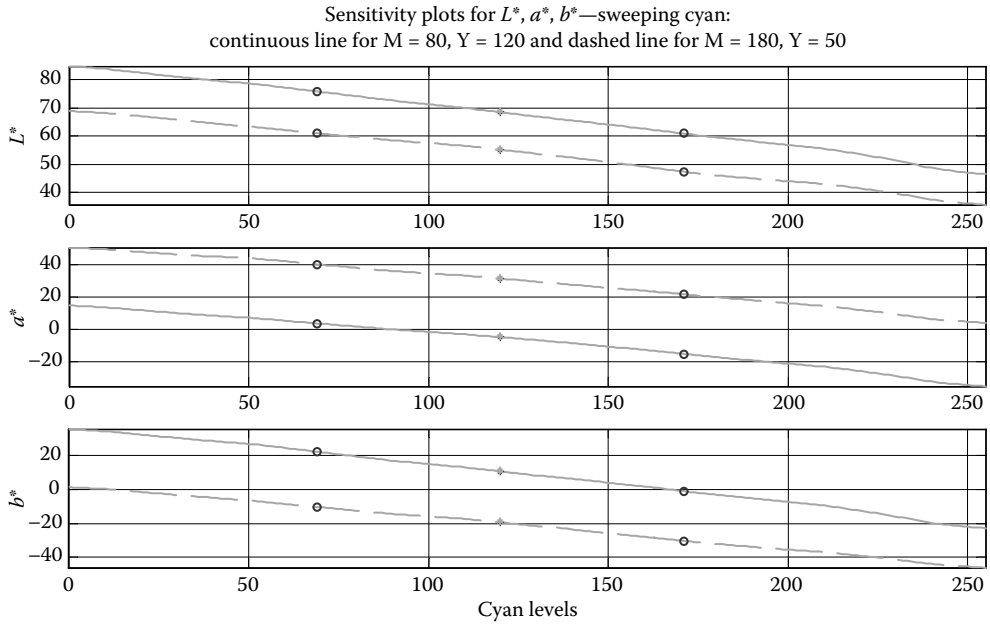

FIGURE 28.3 Diagram representing $L^*a^*b^*$ values when C is varied at constant M and Y with $K = 0$.

where $x = \begin{bmatrix} L^* \\ a^* \\ b^* \end{bmatrix}$, $V = \begin{bmatrix} \Delta C \\ \Delta M \\ \Delta Y \\ \Delta K \end{bmatrix}$, $B = \begin{bmatrix} \frac{\Delta L^*}{\Delta C} & \frac{\Delta L^*}{\Delta M} & \frac{\Delta L^*}{\Delta Y} & \frac{\Delta L^*}{\Delta K} \\ \frac{\Delta a^*}{\Delta C} & \frac{\Delta a^*}{\Delta M} & \frac{\Delta a^*}{\Delta Y} & \frac{\Delta a^*}{\Delta K} \\ \frac{\Delta b^*}{\Delta C} & \frac{\partial b^*}{\partial M} & \frac{\Delta b^*}{\Delta Y} & \frac{\Delta b^*}{\Delta K} \end{bmatrix}$, and $x_o = \begin{bmatrix} L_o^* \\ a_o^* \\ b_o^* \end{bmatrix}$ values for nominal *CMYK*.

When the open-loop system of Figure 28.2 is closed with a controller (see Figure 28.4), a state-space model for the closed-loop system is obtained. Here, the gain matrix K and the integrator comprise the controller for the iterative loop. The input to the integrator is denoted by the vector $u(k)$.

Using this formulation, the integrator can be modeled as follows:

$$V(k) = V(k-1) + u(k) \tag{28.2}$$

Substituting Equation 28.2 into Equation 28.1, the open-loop equation becomes,

$$x(k+1) = B[V(k-1) + u(k)] + x_o \tag{28.3}$$

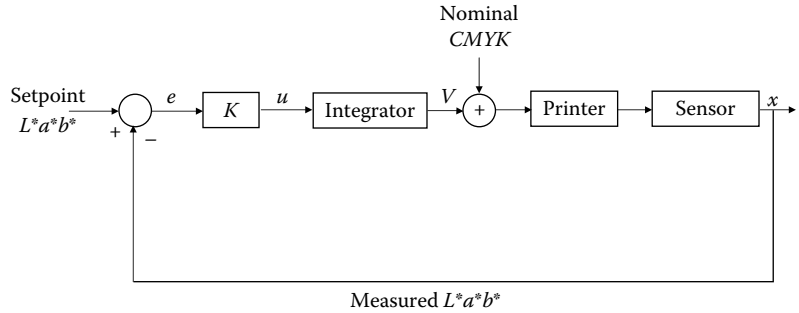

FIGURE 28.4 Closed loop with a gain and integrator as the error-processing controller.

Now we go through some algebraic simplification to derive an augmented open-loop state equation with an explicitly introduced integrator. Consider the representation of Equation 28.1 for the k^{th} print, which is shown below:

$$x(k) = B\,V(k-1) + x_o \tag{28.4}$$

If the Jacobian matrix is invertible, which is not always true at the gamut boundaries, Equation 28.4 can be written as follows:

$$V(k-1) = B^{-1}\,x(k) - B^{-1}\,x_o \tag{28.5}$$

Substituting Equation 28.5 in Equation 28.3 we obtain a state-space representation

$$x(k+1) = B[B^{-1}x(k) - B^{-1}x_o + u(k)] + x_o \tag{28.6}$$

Further reducing Equation 28.6 leads to the standard state-space representation

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{28.7}$$

where the system matrix A and the output matrix C are both identity matrices. The output Equation 28.7 is same as the states. Clearly, the output values for nominal *CMYK* inputs are not present in the final-state equation, due to cancellations. If the printer drifts during the time between calibration prints, then x_o will not be the same, resulting in no cancellations. On the other hand, we can still lump the drifts as uncertainties in the model, because the first approximation of the system with the Jacobian matrix, B, captures major system input–output characteristic needed to stabilize the feedback loop.

It is important to note here that the model described by Equation 28.7 is only applicable to controlling a single desired color as the Jacobian matrix is different for different colors. However, the Jacobian may not differ much between different printers, because, on the whole, the gradient of output colors with respect to the primaries tends to behave consistently between systems.

Even though the spot color control model is simple, a practical algorithm must be used to achieve high-quality color reproduction for colors inside and outside the gamut. The following steps represent one such algorithm:

1. Determine whether the spot color targets are inside or outside the device gamut.
2. If they are outside the gamut, then map the out-of-gamut colors to printable colors that are on the surface or within the gamut using an appropriate gamut-mapping algorithm and determine new target values for the mapped colors.
3. Select an appropriate gray component replacement (GCR), that is, black addition (more on this can be found in Chapter 7 of [1]), for the target values.
4. Apply a closed-loop control algorithm for the GCR constrained set in step 3.
5. Select the best recipe out of the multiple iteration steps.

A tricolor GCR is preferred for spot colors, since it improves color accuracy by holding one of the separations at zero. This approach enables a unique recipe for each spot color, which results in the best visual match across devices. For a tricolor GCR, the four-color printer gamut is represented as a composite of the tricolor gamut subclasses, wherein each gamut subclass consists of a subset of three-color gamuts. Selected spot color targets are assigned to one of the gamut subclasses to calculate the *CMYK* recipe for a given spot color target. An example of a tricolor gamut class consists of four gamut classes, $CMY{-}L^*a^*b^*$, $CYK{-}L^*a^*b^*$, $CMK{-}L^*a^*b^*$, $MYK{-}L^*a^*b^*$.

Two major types of closed-loop algorithms are: (1) a gradient-based and (2) SF-based. These algorithms can be used to find the recipe once a suitable tricolor gamut class has been chosen and, if required, after an appropriate gamut-mapping algorithm is applied when the spot color targets are outside the device gamut. The iteratively clustered interpolation (ICI) algorithm [4] is an example of a gradient-based optimization method, where the initial points are generated through an iterative technique. In this section, we focus

on spot color control using the SF controller. The SF controller can be applied to spot color reproduction with or without GCR constraints.

28.3.1.2 State Feedback with Pole-Placement Design

In this section, we show how the state-space model shown in Equation 28.7 can be used for reproducing spot color recipes in terms of the *CMYK* colorants as determined by control systems. The error between the measured $L^*a^*b^*$ values and the corresponding set-point colors is multiplied by the gain matrix, K (note: this K is different from the primary K in the four-color *CMYK* colorants), to produce a small correction to the nominal *CMYK* values. The gain matrix can be designed using MIMO pole-placement or MIMO Optimal Control methods described in [1]. The integrator integrates the weighted error between the desired and the measured $L^*a^*b^*$ values. Accordingly, from Figure 28.4, the control vector can be written as:

$$u(k) = Ke(k) \tag{28.8}$$

The error vector $e(k)$, which is the difference between the target $L^*a^*b^*$ values and the measured $L^*a^*b^*$ values for a single color. Estimated $L^*a^*b^*$ values are used when iterations are carried out on a model instead of the printer. Iterations can occur until the error is less than a predetermined value. Iterations are stopped after achieving the accuracy limits. Sometimes the *CMYK* values from a previous set of iterations (also called the "best actuators" depending on the algorithm) are used when determining the best colorant recipe for a given spot color. If a pole-placement algorithm is used to calculate the gain matrix K, the number of required iterations can be adjusted via the pole values. For pole locations $\begin{bmatrix}0 & 0 & 0\end{bmatrix}$, satisfactory error can be reached in one iteration (dead beat control), assuming that the print engine has not drifted too far away from the state it was in at the time the printer Jacobian was characterized.

28.3.1.3 Linear Quadratic Regulator Design

A MIMO gain matrix can also be computed using the linear quadratic regulator (LQR) design. The LQR design offers additional freedom to constrain the *CMYK* values, thus offering additional GCR capability. For example, if a spot color can be reproduced only with *CMY* separations, and we have not assigned the gamut classes, then a constraint can be injected to suppress the K separation. The Linear Quadratic Controller, for such a GCR constrained operation, minimizes a selected quadratic objective function for single color, that is, a node color, over the iteration length, N, which is shown below:

$$J = \frac{1}{2}\sum_{k=0}^{N-1}\left[x^{\top}(k)Qx(k) + u^{\top}(k)Ru(k)\right] \tag{28.9}$$

where $x(k)$ is the state vector containing the $L^*a^*b^*$ values and $u(k)$ is the input or actuator vector for a four-color system. In this problem, the single color is modeled using the state-space formulation described above using the printer model (or printer) with 4 inputs and 3 outputs. A 3×4 Jacobian matrix, B, characterizes the system, which is used in conjunction with Equation 28.9 to derive the gain matrix for each spot color.

For example, if the goal is to minimize the error vector between the target vector containing node $L^*a^*b^*$ values and the vector formed by $L^*a^*b^*$ values from the printer, then the objective function could be formed with the sums of the squares of the weighted error vector. Additionally, the sums of the squares of the *CMYK* values (actuators) can be included in the objective function to appropriately weigh the desired actuator. Next, we shall describe the application of LQR design with the intention to suppress

black (i.e., the K separation). The Q and R matrices can be designed as follows:

$$\begin{aligned} Q &= diag\begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \\ R &= diag\begin{bmatrix} r_1 & r_2 & r_3 & \alpha \end{bmatrix} \\ \alpha &= wr + \varepsilon \end{aligned} \tag{28.10}$$

Note that the R matrix includes α, the weight used to suppress black. This factor is a function of the variable w which is designed *a priori* based on the spot color and the device gamut. The constant r is a scale parameter and ε is chosen small, for example, equal to 0.22, to ensure that the R matrix is always positive definite for all spot colors. For some colors, the weight profile, w, is equal to zero. For those colors, there is a risk of violating the positive definiteness condition if nonzero ε is not used. We use fixed values for the other variables r_1, r_2, r_3; however, these values can be varied to adjust K suppression. For example, when the user finds excessive black in the neutrals, they can change the values for the variable r_i (anywhere between 0 and 100). The gain matrix equation is obtained using the approach described in [1 (Chapter 7),5], and the principle of optimality over the iteration interval 1 to N and can be found in [6]. We show the final equations below.

The gain matrix equation is described by:

$$K(k) = \left[R + B^{\top}P(k+1)B\right]^{-1}B^{\top}P(k+1)A \tag{28.11}$$

The recursive equation is defined as:

$$P(k) = A^{\top}P(k+1)A - A^{\top}P(k+1)B\left[R + B^{\top}P(k+1)B\right]^{-1}B^{\top}P(k+1)A + Q$$

and the boundary condition is given by $P(N) = 0$ and $K(N) = 0$. It turns out that the state-space model for each spot color has an A matrix that is equal to the identity matrix.

28.3.1.4 Model Predictive Controller (MPC)

To achieve the best visual match for spot colors, it is important to minimize the perceptual color-difference function, that is, $\Delta E2000$—a color-difference formula defined by CIE ([1]: Appendix A), between the target and measured spot color $L^*a^*b^*$ values for a preset planning or a predictive horizon over the iteration length. The SF algorithm with Pole-placement or LQR design minimizes the Euclidean Norm (ΔE) between the target and measured spot color $L^*a^*b^*$ values. At each iteration, the MPC selects the best gain matrix minimizing $\Delta E2000$ color differences. The best gain matrix then becomes the gain matrix actively used during iteration. This approach has many advantages when compared to the methods described in Sections 28.3.1.2 and 28.3.1.3. Some of the advantages are as follows:

1. It defines better the performance (cost) function regarding the minimization. For example, the minimization of the $\Delta E2000$ criteria during an iteration is important for producing the best perceptual matching for a spot color particularly one outside the gamut but not too far away from the gamut surface. Also, for spot colors on the device gamut, we can include the minimization of control energy (i.e., the energy used during each iteration) in a nonlinear performance function as opposed to a quadratic function in the LQR design, in addition to including the $\Delta E2000$ criteria.
2. It enables the use of predictive/planning horizon during iterations. At each iteration in the spot color algorithm, we can further discretize the computation of the gain matrices to find the one that minimizes the performance criteria, thus creating a convergence behavior. This control is often needed for a printer with halftone noise and drift [7].
3. It enables computation of the printer Jacobian and hence the gain matrices during each iteration, which makes the system more adaptive. Note that the use of more accurate printer models, for example, determined inline with sensors, can further improve the adaptation performance [8].

4. Trade-offs between control energy and $\Delta E2000$ criteria can be easily obtained by adjusting the weighting factors. This is particularly useful for the control of in-gamut spot colors to on-gamut or near boundary spot colors, where on-gamut boundary spot colors can be set to use minimal control energy without leading to instabilities.
5. It allows faster convergence to the desired $\Delta E2000$ values when compared with other methods.
6. It enables accurate spot color control for arbitrary spot colors as in matching hardcopy proofs with a nonstandard spot color library (i.e., custom-defined spot color library by print shop owners).
7. It provides an improved convergence rate (i.e., the number of iterations to reach the $\Delta E2000$ criteria) during iteration, which can reduce the number of iterations and printed spot color samples.
8. Potential opportunities exist to include additional components in the performance criteria. For example, optimization for reduced toner usage. This becomes significant when the number of spot colors increases and the area coverage used by some of the spot colors is high.

The MPC algorithms can be used to obtain the minimization of (1) the $\Delta E2000$ color-difference values, (2) the minimization of the control energy of the actuators, or (3) a compromise between (1) and (2). By selecting the optimization criterion in (1), the designer seeks to minimize the error between the desired color and the measured color. When the criterion (2) is used, the error between the desired color and the measured one is still reduced but at the expense of minimizing the control energy. The implementation of (3) seeks to balance the goals of (1) and (2). It is important to point out that the minimization of the error as well as the control energy results in an "optimal" controller.

Here ideas from [9] are taken to define our nomenclature and the problem we want to solve. A survey of MPC can be found in [10]. Let k denote the iteration number. Let

$$y(k+1) = f(x(k), u(k), d(k)) = \begin{bmatrix} L^*_{k+1} & a^*_{k+1} & b^*_{k+1} \end{bmatrix}^\top \in \Re^3$$

be the outputs (the $L^*a^*b^*$ measurements at iteration $k+1$) obtained by a sensor or printer model, where f is a smooth function of the states $x(k) \in \Re^4$ (i.e., any combination of colors taken from $CMYK$), $u(k)$ is the control input, and $d(k)$ is a white-noise signal. Let $r \in \Re^3$ be the reference $\begin{bmatrix} L^*_r & a^*_r & b^*_r \end{bmatrix}^\top$ values at iteration k. We define the tracking error as

$$e(k+j) = \begin{bmatrix} L^*_r & a^*_r & b^*_r \end{bmatrix}^\top - y(k+j)$$

Our goal is to develop a planning strategy that generates a sequence of control inputs that minimize $e(k)$ for all k. Denote

$$u^i[k, N] = u^i(k, 0), u^i(k, 1), u^i(k, 2), \ldots, u^i(k, N-1)$$

as the sequence of control inputs of the ith plan of length N. Each plan i is formed by a set of control inputs generated by a SF controller computed for a specific printer Jacobian and set of pole locations. The printer Jacobian is computed online using a stored printer model. Pole locations are assigned to each iteration using pole-placement design.

We use the discrete model

$$y_m(j+1) = f_m(x_m(j), u(j))$$

for $j = 0, 1, \ldots, N-1$ (j is the estimation iteration index for plan i) in our simulations and let $y^i_m(k, j)$ be the j^{th} estimated output value generated at time k using the control input $u^i[k, N]$. To see how the control input $u^i[k, N]$ of plan i affects our system, we project the behavior of the system output at iteration k for $j = 0, 1, \ldots, N-1$, that is,

$$y^i_m(k, j+1) = f_m(x_m(k, j), u^i(k, j))$$

as well as the system states

$$x^i_m(k, j+1) = I x^i_m(k, j) + K^i(j) e^i(k, j) \tag{28.12}$$

where $x^i_m(k, j)$ is the jth estimated state values of plan i at iteration k, $I \in \Re^4$ is the identity matrix, $K^i(j)$ is the ith gain matrix used for the entire projection, and $e^i(k, j)$ is the jth estimated tracking error of plan i

at iteration k. It is to be noted here that $x_m^i(k,j)$, are the *CMYK* estimated values, whereas $y_m^i(k,j)$ are the estimated $L^*a^*b^*$ values from the printer model. To evaluate the performance of each plan i, we define the cost function as

$$J(u^i[k,N]) = w_1 \sum_{j=0}^{N-1} (E^i(k+j))^2 + w_2 \sum_{j=0}^{N-1} \left\| u^i(k,j) \right\|^2 \tag{28.13}$$

where

$$E^i(k+j) = \Delta E2000\left(\begin{bmatrix} L_r^* & a_r^* & b_r^* \end{bmatrix}^\top, y_m^i(k+j)\right)$$
$$u^i(k,j) = K^i(j)\left(\begin{bmatrix} L_r^* & a_r^* & b_r^* \end{bmatrix}^\top - y_m^i(k+j)\right)$$

and $\|a\|$ is the 2-norm of a vector a. The variables w_1 and w_2 are positive constants that scale the color-difference formula and the control energy, respectively, and they can be used to put more emphasis on (1) the color difference, (2) the control energy of the actuators used to track the color difference, or (3) achieving a compromise between (1) and (2).

To select the best plan, we compute

$$i^* = arg\ min_i\, J(u^i[k,N]) \tag{28.14}$$

at each iteration k. Then, the control input $u(k) = u^{i^*}(k,0)$ is applied to the system (i.e., the first input from the "best" control input).

Next, we show the performance of the MIMO MPC defined here. We compare these results to the ones generated by a MIMO SF controller. For each color, we consider that one delta value of the Jacobian equals 0.2 and one pole location equals 0.3 for the design of the SF controller. These are common values used in the prior art design of this type of controller. On the other hand, for the MIMO MPC method, we consider 20 delta values linearly spaced in the range [0.02 0.2] and 25 pole locations linearly spaced in the range [0 0.8]. All combinations of the delta values and the pole locations are used to compute both the printer Jacobians and the gain matrices (via pole-placement). Notice that the MPC considers the behavior of 500 controllers (gain matrices) at each iteration k. Furthermore, the MPC projects the performance of each controller over a horizon of length N. Then, the MPC chooses $u(k)$ according to the result from Equation 28.14. For all the simulations below, we let $N = 10$, $w_1 = 1$, and $w_2 = 0$ so that we only focus on minimizing the cost of the color-difference formula.

The reference spot color target values are:

$$\begin{bmatrix} L_r^* & a_r^* & b_r^* \end{bmatrix}^\top = \begin{bmatrix} 79.61 & -8.92 & -22.06 \end{bmatrix}$$

To evaluate the controller performance, we perturb the initial *CMYK* values to $\begin{bmatrix} 105 & 5 & 38 & 0 \end{bmatrix}$, which corresponds to a $\Delta E2000$ spot color error of 7.21. It is worthwhile to mention that we use a numerical printer model in our simulation to obtain $\begin{bmatrix} L^* & a^* & b^* \end{bmatrix}$ values given by *CMYK* at any iteration k. Figure 28.5 shows the $\begin{bmatrix} L^* & a^* & b^* \end{bmatrix}$ values obtained by both the SF and MPC for 10 iterations. Note that the MPC locks in the reference $\begin{bmatrix} L_r^* & a_r^* & b_r^* \end{bmatrix}$ values by its second iteration, whereas the SF controller requires four iterations. The corresponding $\Delta E2000$ values are shown in Figure 28.6. MPC performance surpasses SF (see Figure 28.7) by dynamically selecting (for each iteration) the "best" gain matrix from the available gain matrices pool. Note that the selected controller may not be the optimal one, because we are not considering all the possible combinations of the delta *CMYK* and poles location values. However, it is not practical to consider all possible combinations for every digital count, since such an approach would computationally be very expensive, compromising the time required to generate a control input. Figures 28.8 and 28.9 show the shapes of the cost function defined in Equation 28.13 for the first and second iterations, respectively. Note how the best gain matrix is determined based on the available costs and observe the difference between the cost function values obtained from the MPC and SF. Also, notice

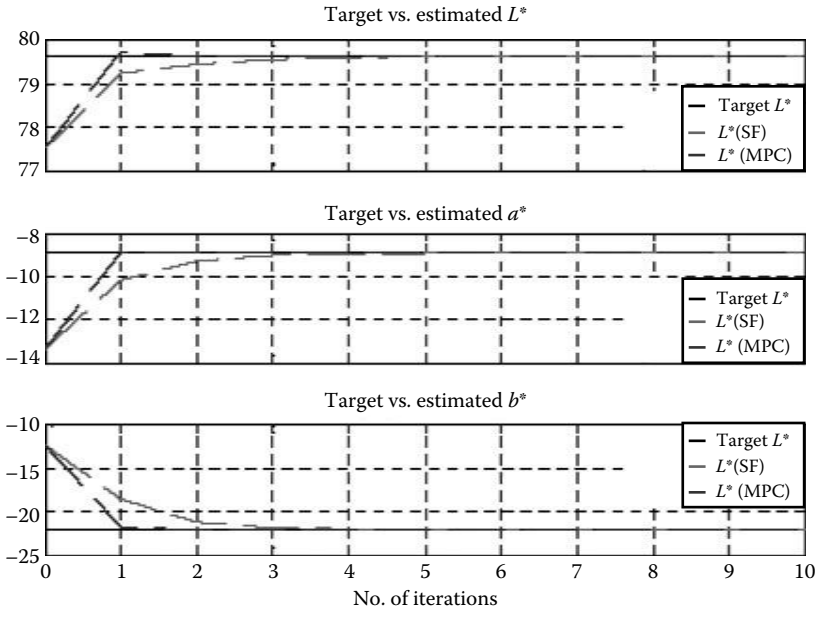

FIGURE 28.5 Reference $\begin{bmatrix} L_r^* & a_r^* & b_r^* \end{bmatrix}$ and $\begin{bmatrix} L^* & a^* & b^* \end{bmatrix}$ values obtained by SF controller and MPC ($w_1 = 1, w_2 = 0$).

how the cost function values decreased considerably by the second iteration as a consequence of applying an "optimal" control input during the first iteration.

Figures 28.8 and 28.9 also provide useful information which can be used to decide the ranges and resolutions for the delta *CMYK* and poles location values. For instance, if the MIMO MPC method is only used for the spot color considered in this case study, then it would be more beneficial to consider less than 20 delta values in the range $\begin{bmatrix} 0.001 & 0.18 \end{bmatrix}$ and less than 25 poles location values within the range

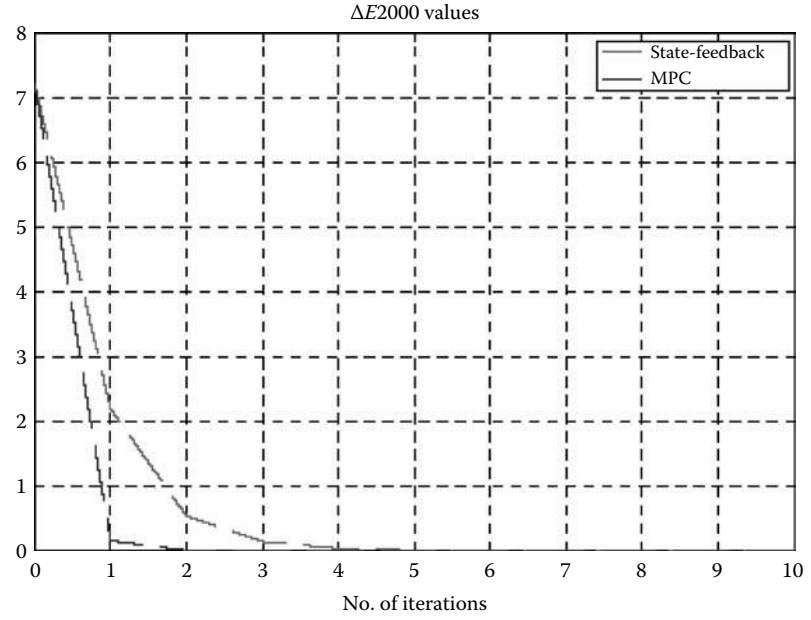

FIGURE 28.6 ΔE2000 values obtained by SF and MPC ($w_1 = 1$, $w_2 = 0$).

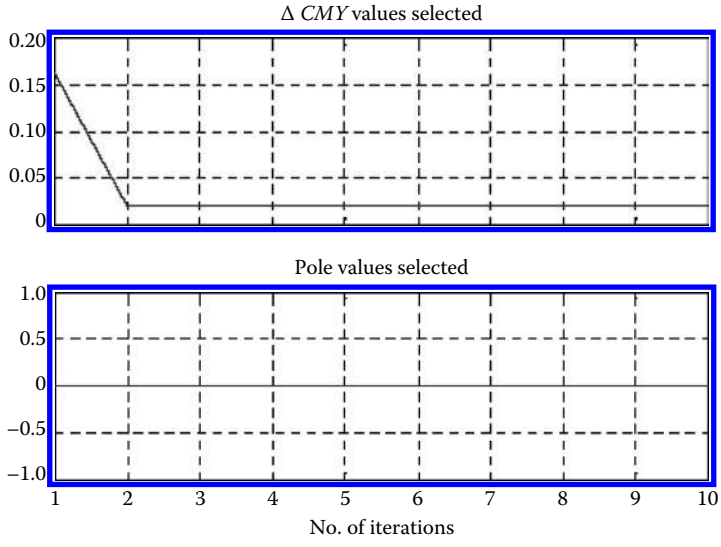

FIGURE 28.7 Jacobian delta *CMYK* and pole locations values selected by the MPC ($w_1 = 1$, $w_2 = 0$).

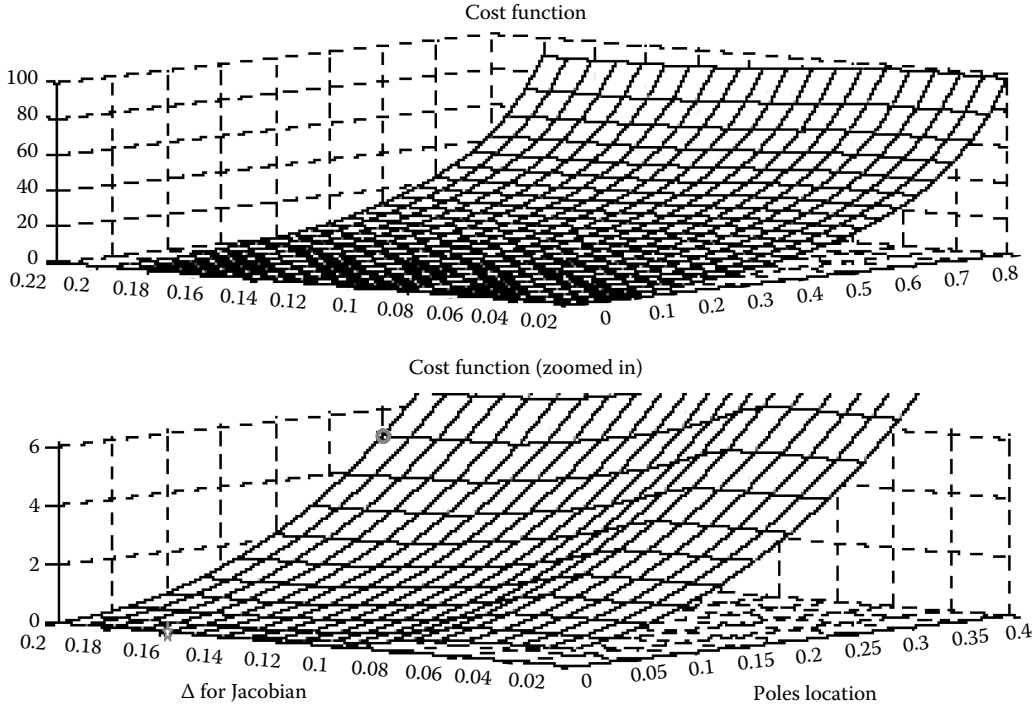

FIGURE 28.8 Cost function values ($w_1 = 1$, $w_2 = 0$) for each Jacobian delta and pole location at the first iteration. The gray "*" at the lower left side of the bottom-plot represents the cost for the MPC method, whereas the gray 'o' at the upper left side represents the cost for the SF one.

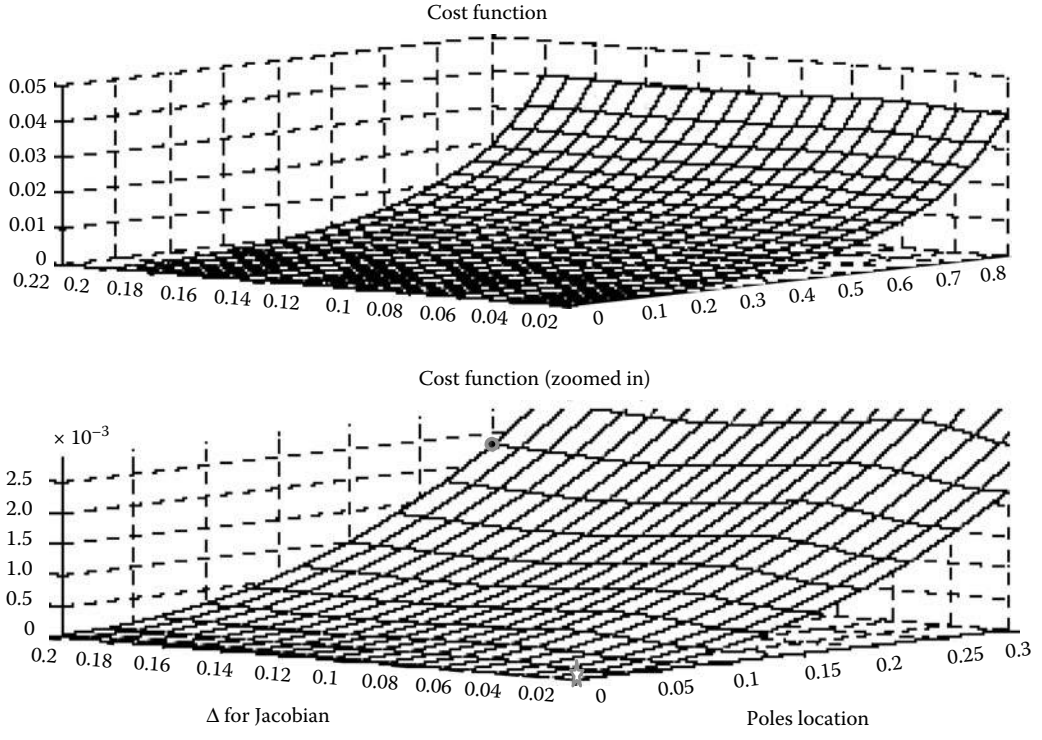

FIGURE 28.9 Cost function values ($w_1 = 1, w_2 = 0$) for each Jacobian delta and pole location at the second iteration. The gray "*" at the lower right side of the bottom-plot represents the cost for the MPC method whereas the gray 'o' at the upper left side represents the cost for the SF one.

$[0 \quad 0.1]$. However, when controlling colors are scattered along the entire gamut, it is probably more convenient to use the delta *CMYK* and poles location ranges listed above. Therefore, both the ranges and the resolutions of the delta *CMYK* and poles location values could be conveniently used as design parameters for the colors to be controlled via the MIMO MPC method.

Now we show the effect of considering the control energy, by setting the weight variable $w_2 = 300$. This time the MPC also emphasizes on the control energy and hence tries to avoid abrupt changes in the $u(k)$ values. This can be seen in Figure 28.10, which shows that the MPC increases the selected pole locations until a final steady value is reached. By increasing the poles location values, the MPC tries to be less aggressive by avoiding abrupt changes in the $u(k)$ values. Comparing the results obtained in Figure 28.10 with the ones obtained in Figure 28.7; the consideration of control energy ($w_2 = 300$) is evident. Furthermore, notice that the decay rate of the $\Delta E2000$ values in Figure 28.6 is lower than the one in Figure 28.11 since the MPC does not allow abrupt changes in the control inputs. It should be clear that weight parameters can be optimized in order to achieve a desired controller performance during iteration, particularly for spot colors near or on the boundary. In Figures 28.12 and 28.13 additional simulation results are shown for four spot colors with

$$
\begin{aligned}
[L_r^* \quad a_r^* \quad b_r^*]^\top &= [90 \quad 0 \quad 96], \\
[L_r^* \quad a_r^* \quad b_r^*]^\top &= [22 \quad -2 \quad -31], \\
[L_r^* \quad a_r^* \quad b_r^*]^\top &= [24 \quad -20 \quad -22], \qquad \text{and} \\
[L_r^* \quad a_r^* \quad b_r^*]^\top &= [54 \quad -64 \quad 28]
\end{aligned}
$$

Clearly, the multigain algorithm presented in this section outperforms the single controller SF method.

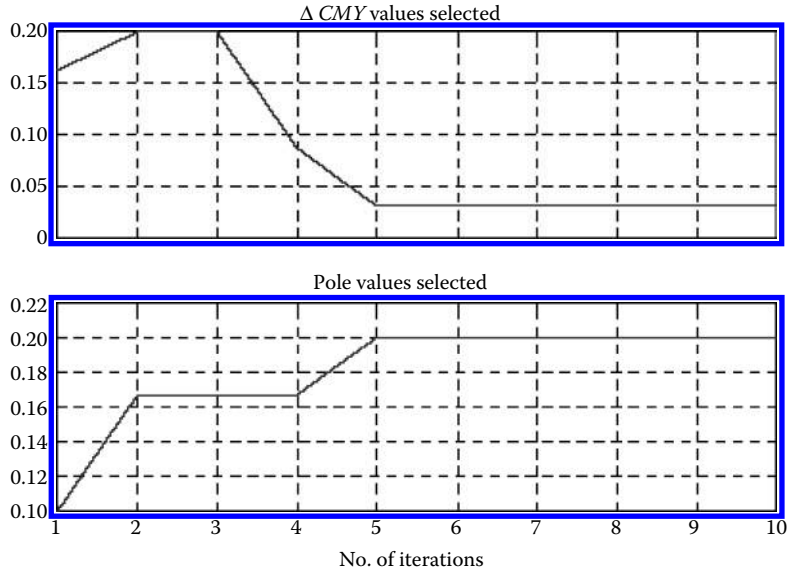

FIGURE 28.10 Jacobian delta and poles location values selected by the MPC when $w_1 = 1$, $w_2 = 300$.

28.3.2 1D, 2D, 3D Control: For Rendering High Quality Images

28.3.2.1 Introduction

The control approaches applied to achieve accurate spot color can be extended to create a one-dimensional (1D) gray balance or a single separation tone reproduction curve (TRC), as well as two or three-dimensional (2D or 3D) transforms to enable accurate rendering of images. 1D tone curves or 2D/3D LUTs are generally implemented in the DFE, which applies a transformation to the image pixels to match

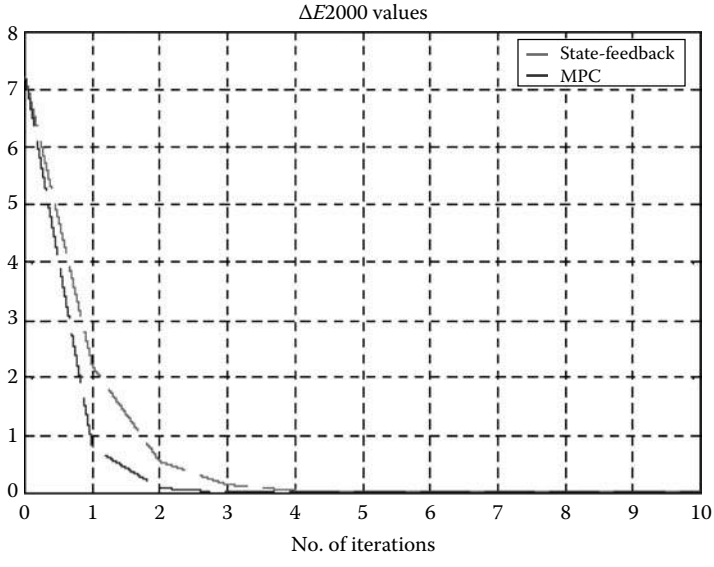

FIGURE 28.11 $\Delta E2000$ values obtained by SF and MPC for $w_1 = 1$, $w_2 = 300$.

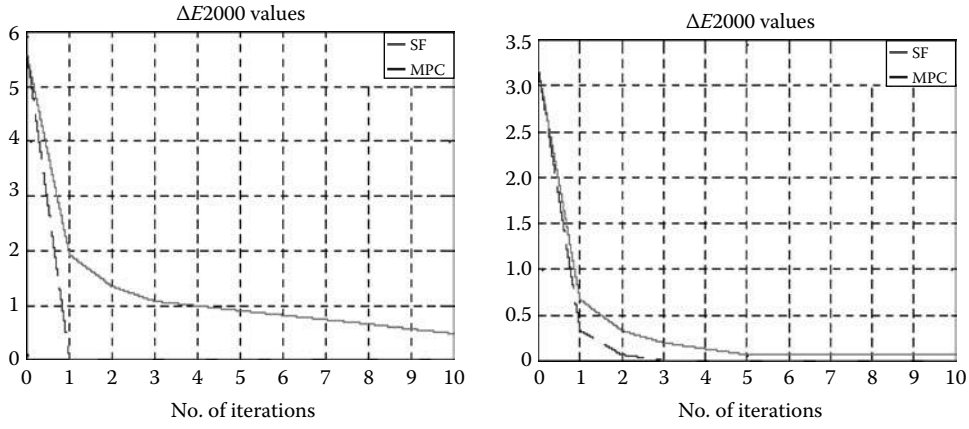

FIGURE 28.12 $\Delta E2000$ values obtained by SF and MPC for $w_1 = 1$, $w_2 = 300$ for two additional spot colors [Notice the steady state error is lower in MPC when compared to SF].

them to electronic input or to a printed document from a reference device, such as an offset printer or a proofer. Below, we provide a brief description of the 1D, 2D, and 3D transforms, and describe some new algorithms that are not disclosed in [1].

28.3.2.2 1D Channel-Wise Linearization

In the channel independent 1D calibration, each primary color (called channel, e.g., $C, M, Y,$) is independently linearized. Through linearization, we would like to make the ΔE obtained by measuring color from paper to be linear with respect to area coverage. The calibration begins by printing each primary at different digital counts between the minimum (0) and maximum (255) values with the other primaries held at zero. With 1D linearization, the channel-linearized printer can emulate an ideal printer which has the characteristic of linearized ΔE from paper. The ΔE from paper is the Euclidean norm between target color and the paper white in the device-independent color space ($L^*a^*b^*$). The $\Delta E2000$ metric is another potential metric for the paper-based 1D channel-wise calibration. The 1D channel-wise linearization

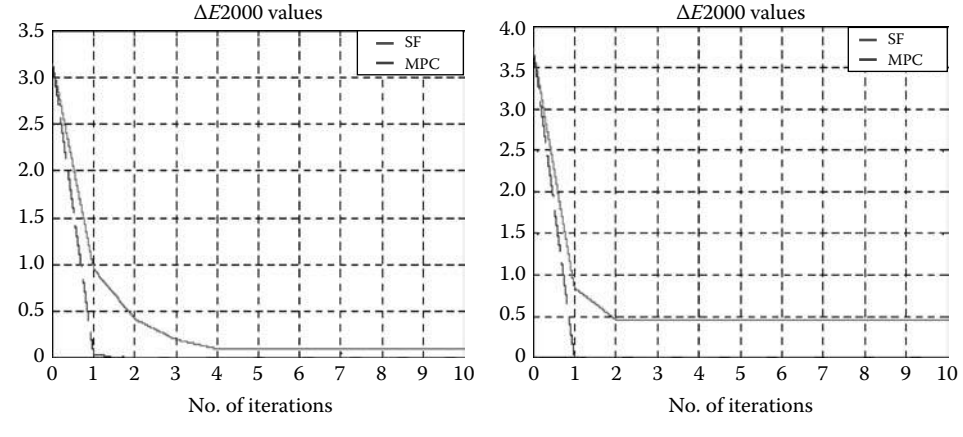

FIGURE 28.13 $\Delta E2000$ values obtained by SF and MPC for $w_1 = 1$, $w_2 = 300$ for another two additional spot colors. [Notice the steady-state error is lower in MPC when compared to SF].

creates four 1D tone curves; $C' = f_c(C)$, $M' = f_m(M)$, $Y' = f_y(Y)$, $K' = f_k(K)$, that map the input *CMYK* to the device $C'M'Y'K'$ for every digital count.

28.3.2.3 1D Gray Balance Calibration

Depending on the device, the 1D channel-wise linearization may not give sufficient color linearization when primaries are mixed at different proportions. Hence, another 1D gray balance calibration procedure is introduced. In some devices 1D channel-wise linearization is followed by 1D gray balance calibration to further linearize the device to color. A 1D gray balance calibration is a process by which 1D TRCs are produced for each color separation (e.g., *CMY* with K held equal to 0) using a measurement of output color near neutral.

There are many definitions of gray balance. We use the equivalent neutral gray balance to introduce the approach. This means, for example, when the input digital count of $C = M = Y = 40$ (in a 8-bit digital imaging system) is used to print a color, the printed color is a gray color with $L^* = 100 - (100/255) * 40 = 84.3$ and $a^* = b^* = 0$. The *K* separation does not come into this definition. For a device to be considered as "gray balanced," the three 1D transformations, $C' = g_c(C)$, $M' = g_m(M)$, and $Y' = g_y(Y)$, are required. These functions map the input *CMY* to the device $C'M'Y'$ for every digital count. These are called gray balanced TRCs. For the *K* separation, a 1D mapping function $K' = f_k(K)$, similar to 1D channel-wise linearization is introduced, which effectivly linearizes the *K* separation channel for ΔE with respect to paper.

28.3.2.4 2D Calibration

The 1D calibration is limited to color balancing the printer to one of the color axes (e.g., equivalent neutral gray) or can be used to achieve a linearized response with respect to paper as in 1D channel-wise linearization. In 2D calibration, three 2D TRCs are constructed as: $C' = f_1(C, M + Y)$, $M' = f_1(M, C + Y)$, $Y' = f_1(Y, C + M)$. For example, the Cyan output C' is a function of input Cyan and sum of input Magenta and Yellow. The advantage of 2D calibration over 1D is that we obtain good control over five-color axes per primary. By controlling the gray axis, we can achieve good gray balance; controlling the Cyan primary axis, we can achieve channel independent linearization. By controlling other axes such as primary to black, we can linearize the printer along those axes, achieving better control of the printer gamut. The process of linearization along these axes is similar to the channel-independent linearization.

28.3.2.5 3D or Multidimensional Profiles

A 3D or multidimensional profile LUT is a destination profile, which is a GCR constrained, gamut mapped LUT, which transforms the device-independent color $L^*a^*b^*$ or *XYZ* to device-specific *CMYK* space. Generally, most photographic quality images are created in *RGB* space. To reproduce the *RGB* image accurately on a four-color printer, the *RGB* triplets for every pixel must be transformed to the device *CMYK* space. Normally, this is done using LUTs. A source profile LUT is a three-to-three transformation from *RGB* color space to device independent $L^*a^*b^*$ color space. If *CMYK* images are to be rendered, then the source profile is a four-to-three transformation from the source *CMYK* space to a device-independent $L^*a^*b^*$ space. Accordingly, a combination of source and destination LUTs are used to tranform the images to the device *CMYK* space.

Multidimensional profile LUTs provide the capability and architecture to develop good transformations to match the printed color to a proofing device for all the colors, assuming the colors are within the common intersection gamut. A multidimensional profile LUT has finite nodes for which device *CMYK* values are calculated during the profile creation stage. Input image files with color pixels not on the nodes are interpolated. In an ICC image path architecture, the profiles provide simple linear interpolation with limited resolution for processing pixels at high speed. Color pixels not on the ICC profile node are linearly interpolated to obtain their values.

To build a good multidimensional LUT, one must find the right *CMYK* formulation for each color (or node) to produce a satisfying printout of the target image/document. To be successful, one must

correctly choose: (1) the number of nodes per LUT, (2) the formulations (called GCRs or constraints on *CMYK* choice), (3) the gamut mapping algorithms for out-of-gamut colors, (4) the inversion algorithms, and (5) the printer models. For the inversion process, each node of the profile LUT can be regarded as a spot color with the desired values specified in the device-independent color $L^*a^*b^*$ or *XYZ*. A GCR constrained inverse can be carried out using the spot color control approach for each of the node colors by iterating on a printer model or by iterating on a printer. Detailed description of the underlying algorithms are provided in Chapter 7 of [1]. Below, we share additional algorithms applied to GCR constrained inversion and gamut-mapping functions.

28.3.2.6 GCR Constrained Inverse—Cooperative Control Strategy

For high-quality printing, 3D profile LUTs are generally of dimensionality $33 \times 33 \times 33$ or smaller, and typically involve a transformation from $L^*a^*b^*$ to a *CMYK* device space with values available at every node. This means 35937 nodes must be populated with device *CMYK* values. Multidimensional interpolation methods are used for finding the device values that are not on the nodes. Image files often have millions of pixels and a large number of those pixels are not on a node. Hence, the interpolation methods have to be simple and fast.

The nodes are selected based on a strategy to maximize the inversion accuracy for the interpolated colors. A GCR strategy is required to obtain pleasing color with an appropriate choice of separations for lighter colors. GCR provides a method to substitute *K* (black) for *CMY* mixtures when rendering a given color. This substitution results in an extension of the darker region of the gamut by changing lightness (or darkness) when compared to printers without a *K* separation. It helps to reproduce shadows, gray areas, and muted tones in images. Other benefits of introducing *K* are the resulting toner savings, color stability, impact on smoothness of images, and so on. However, if high levels of *K* are used throughout the gamut, a dirty/grainy appearance can arise in flesh tones, sky tones, and other important colors. Smoothness and gamut coverage must also be taken into account when rendering pixels with black; it is a critical element in the inversion process, since it introduces redundant solutions. The GCR choice requires a delicate trade-off among these competing requirements. Generally, print vendors finely tune the addition of black intelligently either by using complex algorithms or by using carefully designed experiments. Experiments are often done with many iterations to get the right amount of *K*. Once the tuning is done, the GCR is included as a part of a multidimensional LUT. At a more basic level, a GCR strategy involves suitably combining *CMYK* to provide a pleasing color output, an optimal gamut, constraints on the area coverages of neighboring nodes, and so on [11–13].

Components of profile LUTs are described in [1] (see Figure 28.14 for a schematic diagram for a single node of a profile LUT). An inverse printer model (P^{-1}) is a mapping from a sampled device-independent color space like $L^*a^*b^*$ to a device-dependent color space. This is defined mathematically as P^{-1}: $L^*a^*b^* \rightarrow CMYK$. Out-of-gamut $L^*a^*b^*$ values are mapped to the boundary nodes or the nodes inside the printer's gamut using appropriate gamut-mapping algorithms. We can apply a state space, LQR, or MPC-based design to find a suitable *CMYK* recipe for each of the node color target $L^*a^*b^*$

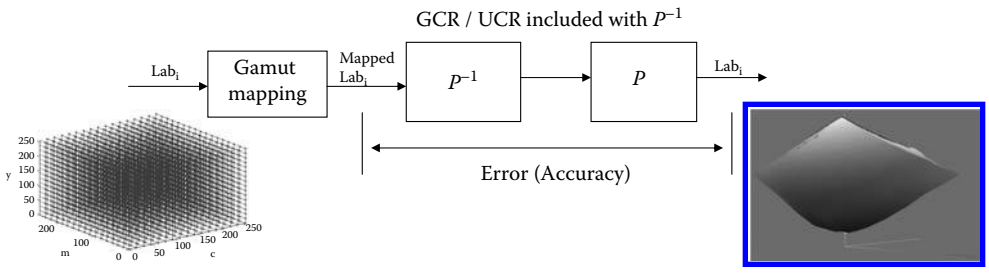

FIGURE 28.14 Forward and inverse printer is shown schematically with gamut mapping.

values without knowing the recipes of the neighboring nodes. The problem with this approach is that, although each node may have a recipe that is good for producing a particular color, when all the recipes are put together in a profile LUT, the images may be rendered with separation noise/contours (i.e., sharp transition in colors) that creates a visible image quality defect. For example, a smooth transition near the shoulder of a human being might be rendered with jumps even though each LUT node has an accurate color recipe. This is because the whole image is rendered by interpolating pixels using recipes from the nodes in its neighborhood. As a result, smoothness in target images may not result in the same smoothness in the rendered image, due to formulation jumps between nodes in the separations. These jumps are caused by ambiguity in the choice of *CMYK* separations since nodes that are in the neighborhood in the $L^*a^*b^*$ color space could be rendered using *CMYK* recipes that are far apart from each other. Unfortunately, it is not possible to soften the jumps via iterative smoothing during the inversion process without sacrificing accuracy. As a result, this problem is ideal for the application of the cooperative control theory.

The work presented below has been motivated by techniques developed for cooperative control, a topic that has received a great deal of attention by the control community in the last decade. Strategies for controlling groups of different types of autonomous air vehicles (AAVs) (connected via a communication network to implement a "vehicle network"), may hold the potential to greatly expand operational capabilities at a lower cost. Cooperative control for the navigation of such vehicle groups involves a coordination of activities and so the agents work together and achieve a common goal. There is a significant amount of current research focused on the cooperative control of AAVs. Solutions to general cooperative control problems can be obtained via solutions to vehicle route planning (VRP) problems [14]. Additional VRP-related work focusing on cooperative search and coordinated sequencing of tasks is in [15–18]. Other cooperative control methods include gradient algorithms [19–21], multisensor fusion [22], surrogate optimization [23], and receding horizon control [15,16,24]. Applying these approaches, significant mission performance benefits can be realized via cooperation in some situations, most notably when there is a high level of certainity.

In a four-color printer, a specific node color ($L^*a^*b^*$) can be achieved by several different *CMYK* recipes, while this may be okay for rendering spot colors, it is not suitable for rendering images with color gradients/sweeps. Here we show how to obtain an $L^*a^*b^*$ to *CMYK* transformation with smooth transition between every neighborhood node in the LUT of the *CMYK* space starting from a known specification. The smoothness is preserved by using two novel algorithms; (1) a MIMO control algorithm and (2) a neighbor detection algorithm in $L^*a^*b^*$ space, where neighboring pairs cooperate mutually during control iterations by exchanging information in order to guarantee a smooth transition between them in the *CMYK* space. This work differs from other approaches in that both accuracy and smoothness is achieved in one step.

Let us consider that the nodes (i.e., $L^*a^*b^*$ values) have been classified as inside and outside the device gamut. The *CMYK* values for out-of-gamut nodes are found using gamut-mapping algorithms. We shall describe one type of gamut-mapping algorithm in Section 28.3.2.7. In the present section, we are interested in finding the *CMYK* values for in-gamut nodes so that the accuracy and smoothness requirements are met. However, the *CMYK* values for all the nodes contained in the LUT could be obtained in the same manner once the out-of-gamut colors are mapped to the surface of the printer. It is understood that the device gamut is measured when a printer is characterized using a set of printed patches with appropriate *CMYK*. Let $CMYK_A$ be the values of a node A which is assumed to be known. This assumption is reasonable, since a multidimensional LUT designer typically specifies the desired value of K (i.e., GCR) along the neutral axis. Once the K separation is known, there is a unique CMY, which can be found using a spot color control method (Section 28.3.1). Given these assumptions, it is possible to estimate the *CMYK* values for the "closest" node B ($CMYK_B$). Suitable distance metrics in $L^*a^*b^*$ space (e.g., perceptual distance $\Delta E2000$ or Euclidean distance ΔE) can be used to define the "closest" node. The node B may have numerous possible combinations of *CMYK*. Among them, one *CMYK* recipe is more suited than the rest to maintain the smoothness with the neighboring node.

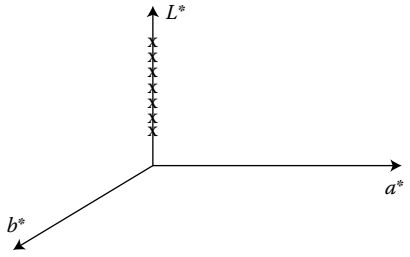

FIGURE 28.15 Nodes contained in recruiting set (shown in Xs).

To create a smooth profile LUT and still maintain the GCR constraints, we define two groups of node colors from the in-gamut set that are specified by their $L^*a^*b^*$ values. The first group is called the "recruiting set" with one or more nodes whose *CMYK* values are determined *a priori*. These *CMYK* values can come from the GCR specifications along the neutral axis. The second group is called the "candidate set" with one or more nodes whose *CMYK* values are to be determined. Conceptually, the purpose of the recruiting set is used to determine potential nodes from the candidate set that could become part of the recruiting set. Similarly, the goal of the candidate set is to market themselves before the recruiting set in order to be recruited. Below, we present a step-by-step algorithm to produce a smooth profile LUT using this kind of cooperative control strategy:

1. Define a recruiting set $R = \{1, 2, \ldots, N\}$ that contains $N \geq 1$ nodes with $L^*a^*b^*$ values. The location of these nodes in the $L^*a^*b^*$ space may be decided by the designer. For example, the recruiting set could come from all the nodes along the neutral axis (i.e., $L^* \neq 0, a^* = 0 = b^*$). Other options are also possible. The motivation of selecting colors along the neutral axis is to inherit the *CMYK* values from a known GCR table such as from a well-tuned profile LUT or a manually specified GCR curve. Thus, we are forcing these colors to behave the way we want. Figure 28.15 shows the location of the nodes from an example recruiting set.
2. Use a desired GCR to compute the *CMYK* values of any node in the recruiting set. One option is to use the *K* restricted function [25] to derive the *CMYK* recipes for all the $L^*a^*b^*$ values contained in the recruiting set, however, other methods could also be used. Figure 28.16 shows one possible selection of the *CMYK* response for colors located on the neutral axis obtained from 33 *RGB* levels.
3. Define a candidate set $C_a = \{1, 2, \ldots, M\}$ that contains M number of nodes with $L^*a^*b^*$ values. This list comes from all the in-gamut nodes in the LUT, but again could also contain the in-gamut plus the out-of-gamut colors that have been mapped to the surface of the printer.
4. Compute the distance between each node $i \in R$ and $j \in C_a$ (see Figure 28.17). One metric that could be used here is the $\Delta E2000$ formula. Another choice is the Euclidean distance, ΔE.
5. For each candidate node, determine the minimum distance, $\min_{ij} \Delta E2000$, between a recruiting and candidate node. The closest candidate node is denoted as j^*.
6. Compute the *CMYK* values of each candidate node by running the MIMO spot color control algorithm and using the *CMYK* value of a node in the recruiting set as the nominal value. The recruiting process is neighbor-driven since it always selects the nodes with the minimum distance between any recruiting and candidate nodes. Once a pair of nodes has been identified, the cooperation takes place, since the *CMYK* values of the recruiting set is shared with the candidate set. The candidate node uses a MIMO controller to iterate several times and converge to a new *CMYK* value that is close to its closest neighbor.
7. The closest node identified in step 5 now becomes part of the recruiting team, that is, $R = R \bigcap \{j^*\}$, and no longer belongs to the candidate set, that is, $C_a = C_a \setminus \{j^*\}$.
8. Repeat steps 5–7 until set C_a is empty, that is, all candidates have been recruited.

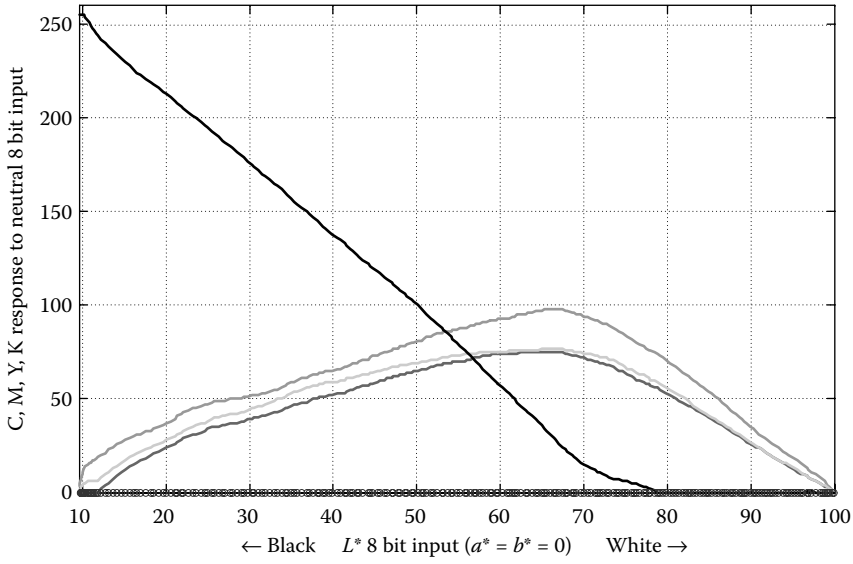

FIGURE 28.16 *CMYK* response of GCR on the neutral axis. Circles in the x-axis represent all the L^* levels considered for the GCR.

This strategy processes the nodes contained in the profile LUT from the neutral axis to the boundary of the printer's gamut. Figure 28.18 shows several snapshots of the chroma plane where the order in which the nodes are processed is depicted to some degree. The sharing of information with individual node control loops is used to implement a tracking (or iterative) algorithm to compute the closest *CMYK* of the selected color in the candidate set to the *CMYK* of the color in the recruiting set. We design MIMO SF controllers [3] to update the *CMYK* recipe that, when converged, will accurately find the recipe to reproduce the desired $L^*a^*b^*$ value.

The system in Figure 28.4 can be expressed as Equation 28.7. Once the above process is finished, we have all the information needed to build the profile LUT. Next, we provide an example of how this technique will be implemented. Suppose that we start having 24 recruiting nodes along the neutral axis with values from $\left[L^*\ a^*\ b^*\right]^\top = \left[15\ 0\ 0\right]$ to $\left[L^*\ a^*\ b^*\right]^\top = \left[100\ 0\ 0\right]$. Note that the L^* values for the recruiting nodes are uniformly spaced at 5 units increments. On the other hand, we only select two colors in the recruiting set to support our claim, that is, $\left[L^*\ a^*\ b^*\right]^\top = \left[56.65\ 6.42\ 6.5\right]$ (Color #1) and $\left[L^*\ a^*\ b^*\right]^\top = \left[68.23\ 6.43\ 6.49\right]$ (Color #2). Color #1 is first selected, because the algorithm determines that it is the closest node (minimum ΔE2000 distance) to the node in the recruiting set with $\left[L^*\ a^*\ b^*\right]^\top = \left[65\ 0\ 0\right]$ and $CMYK = \left[128\ 97\ 101\ 0\right]$. Sensitivity plots for these *CMYK* values are shown in Figure 28.19. Notice that in order to get the $\left[L^*\ a^*\ b^*\right]^\top = \left[56.65\ 6.42\ 6.5\right]$ values of color

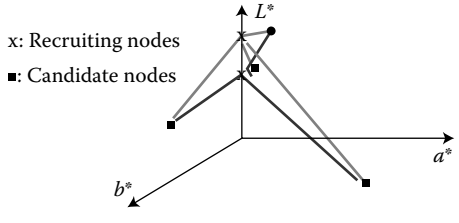

FIGURE 28.17 Computation of metric between any node in the recruiting set and any node in the candidate set.

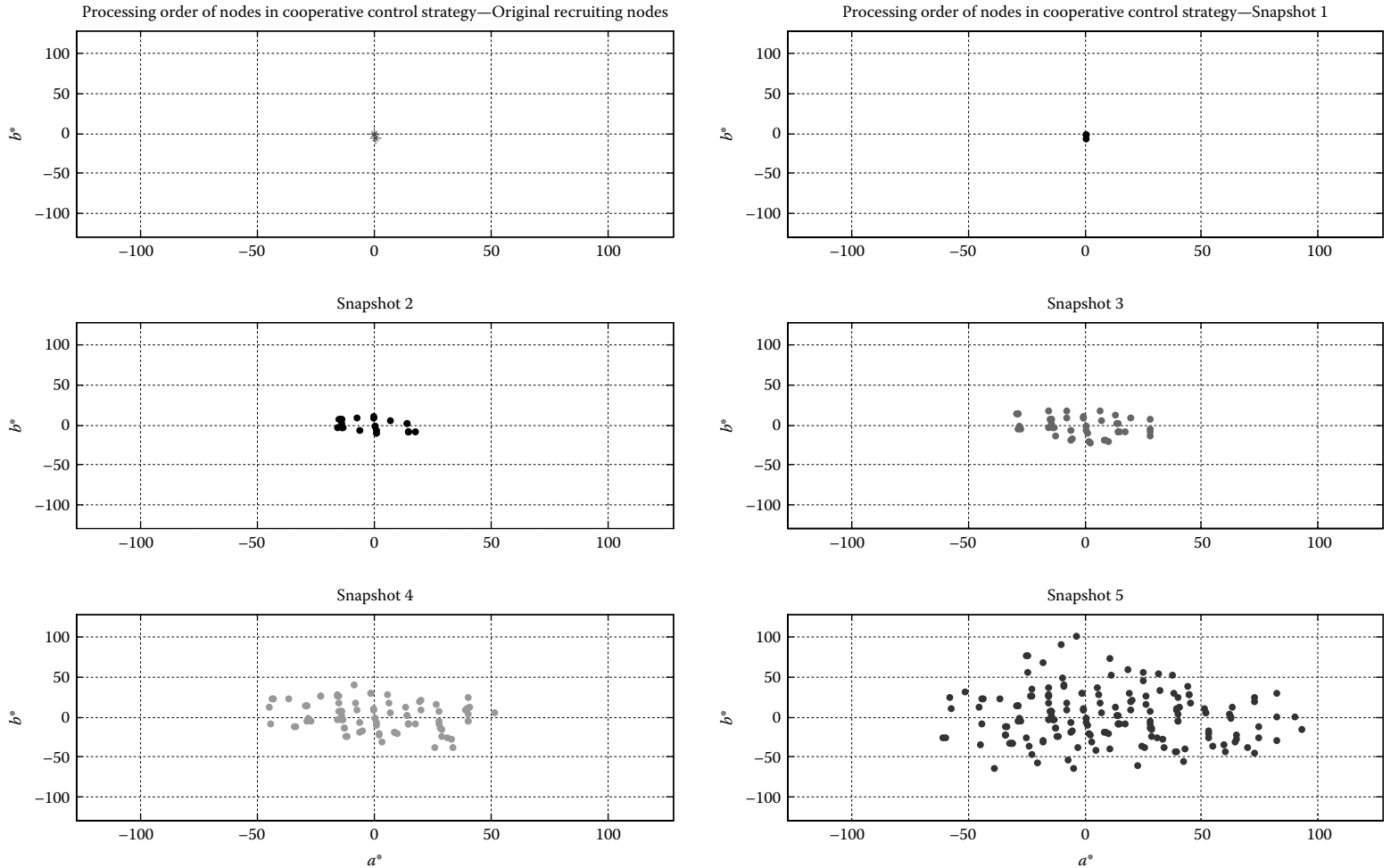

FIGURE 28.18 Snapshots of processing of nodes in the profile LUT using a cooperative control strategy.

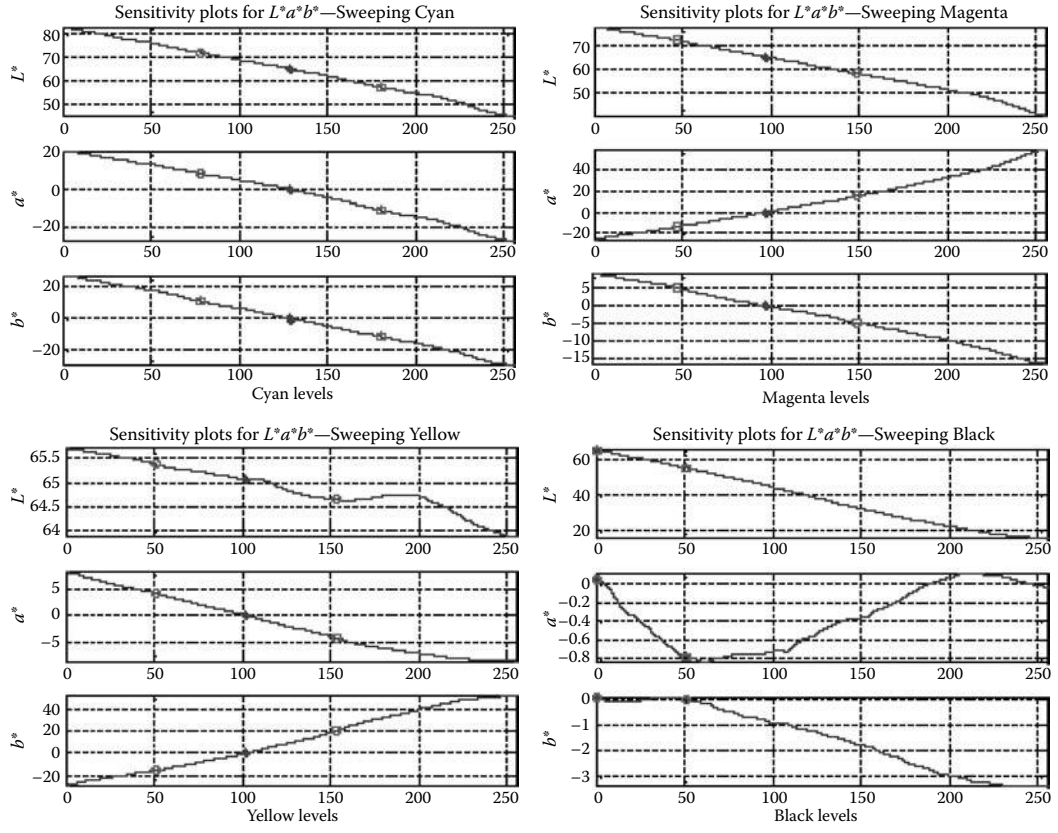

FIGURE 28.19 Sensitivity plots for $CMYK = \begin{bmatrix}128 & 97 & 101 & 0\end{bmatrix}$. The stars indicate the nominal values, whereas the circles indicate the points used to compute the Jacobian around the nominal point.

#1, the controller starts at the nominal *CMYK* from the closest recruiting node and then has to track the sensitivity plots shown in Figure 28.19. This means the controller will iterate the *CMYK* values until the desired $L^*a^*b^*$ values are reached. Smoothness is achieved by starting at the appropriate nominal value. By following the trajectories provided by the sensitivity plots, we guarantee that there is a unique *CMYK* solution for any candidate color; an important attribute when neighboring colors are located in nonlinear regions of the printer's gamut. The approximate $CMYK = \begin{bmatrix}128 & 97 & 101 & 0\end{bmatrix}$ values for color #1 could be inferred by the sensitivity plots; however, this will result in some inaccuracy, since the plots do not account for any interactions between colors. The final *CMYK* values obtained using this approach are $CMYK = \begin{bmatrix}111 & 111 & 112 & 1\end{bmatrix}$. Thus, this node, color #1, is now part of the recruiting set.

Next, color #2 is processed and the algorithm detects that it is close to the node that has $\begin{bmatrix}L^* & a^* & b^*\end{bmatrix}^\top = \begin{bmatrix}56.65 & 6.42 & 6.5\end{bmatrix}$ and $CMYK = \begin{bmatrix}111 & 111 & 112 & 1\end{bmatrix}$, which is color #1. Sensitivity plots for color #1 are shown in Figure 28.20. Notice that in order to get the $\begin{bmatrix}L^* & a^* & b^*\end{bmatrix}^\top = \begin{bmatrix}68.23 & 6.43 & 6.49\end{bmatrix}$ values of the second color, the controller will iteratively modify the *CMYK* values until the desired $L^*a^*b^*$ values are reached. For this case, the approximate $CMYK = \begin{bmatrix}88 & 113 & 112 & 0\end{bmatrix}$ values for color #2 could be inferred by the sensitivity plots; however, this will again result in some inaccuracy, since the plots do not account for any interactions between colors. The final *CMYK* values obtained using this approach are $CMYK = \begin{bmatrix}100 & 103 & 104 & 0\end{bmatrix}$. The two cases mentioned above show how a controller can be used to track the trajectories of neighbor nodes in such a way that the obtained new *CMYK* values are closest to the selected neighbor.

It is important to point out that the Jacobian matrix *B* and gain matrices of the closest candidate node set is computed using local information of the recruiting node. These values remain fixed during all control

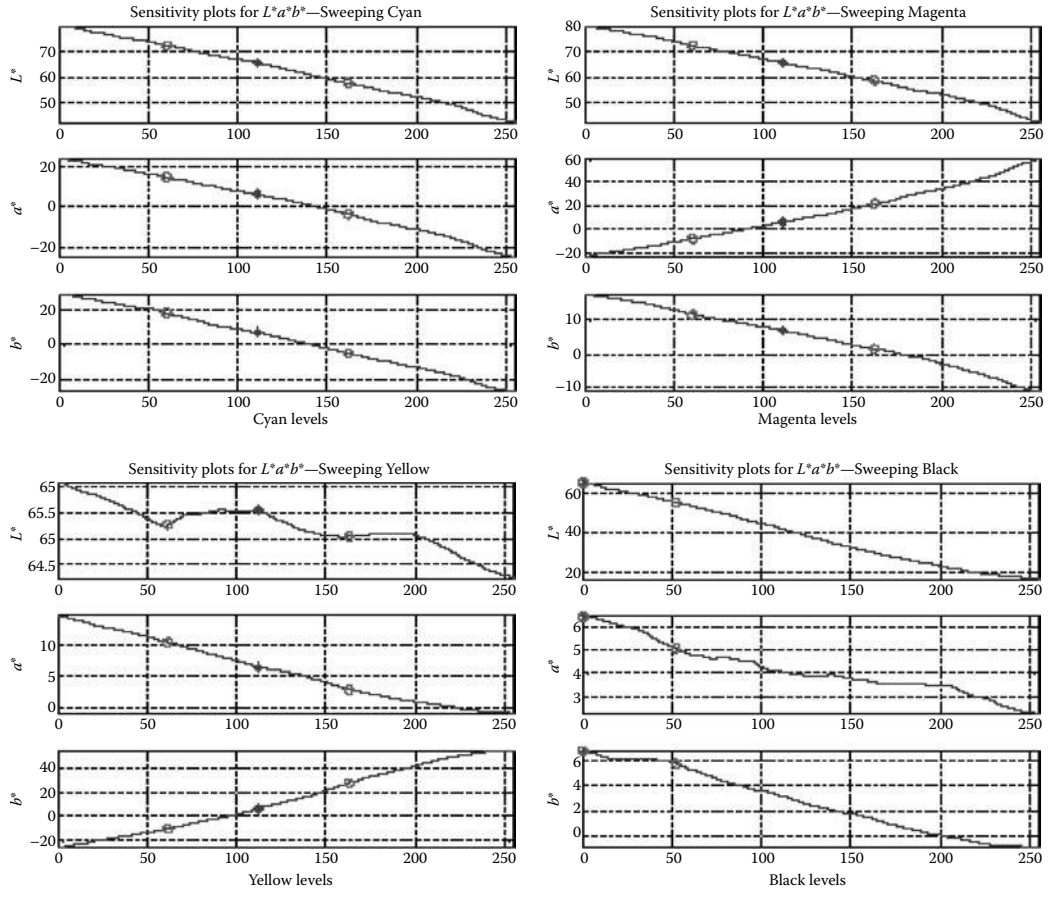

FIGURE 28.20 Sensitivity plots for $CMYK = \begin{bmatrix} 111 & 111 & 112 & 1 \end{bmatrix}$. The stars indicate the nominal values, whereas the circles indicate the points used to compute the Jacobian around the nominal point.

iterations. It is suggested to compute the Jacobian and gain matrices at each iteration, since this could better capture the nonlinearities present in the printer. This requires that B is replaced by $B(k)$ and K is replaced by $K(k)$ in Equations 28.7 and 28.8, respectively. This option will guarantee convergence to the closest $CMYK$ value. This variant is important to consider when the candidate set is sparsely populated as in a less dense profile LUT (e.g., 12^3 LUT as opposed to 33^3 LUT).

28.3.2.7 Gamut Mapping

Gamut mapping is a mapping of out-of-gamut colors to colors that are in-gamut. It is a tone scale modification used to preserve the original color and appearance as much as possible when a device has a smaller gamut than that required to perfectly reproduce all colors in an image. Without gamut mapping, a device like a printer, would be forced to clip out-of-gramut colors. Accordingly, gamut mapping is a key feature used in every color reproduction device.

There exists a considerable amount of literature written about gamut mapping algorithms [26]. There is no unique gamut-mapping method that can satisfy all the requirements, which include pleasing color, contrast, lightness, chroma, hue, etc. Some gamut-mapping algorithms offer feature enhancements in one region of the gamut and others are more favorable elsewhere in the gamut. As such, device designers generally employ a blend of gamut-mapping functions in their color management systems.

Techniques used for dealing with out-of-gamut colors include gamut clipping and gamut compression. In gamut clipping, all out-of-gamut colors are mapped to a color on the gamut "surface" in some way that minimizes the degradation of the resultant output while in-gamut colors are left unaltered. A common form of clipping involves a ray-based approach, wherein a ray is drawn from a desired out-of-gamut color to a point on the neutral axis. The location or point where the ray penetrates the gamut surface becomes the gamut-mapped color. Such a strategy is implemented to preserve hue through the gamut-mapping operation. In gamut compression, both in-gamut and out-of-gamut colors are altered to map the entire range of image colors to the printer gamut. Gamut-mapping methods, both ray based and gamut compression, can be done efficiently using control-based approaches [1, Chapter 7].

More than 90 algorithms are available in [27]. The control-based approach is described in detail in [1]. A merit-based feedback system is a control-based approach that offers a way to automatically select the gamut-mapping algorithms from a library in order to optimize a merit function. In this method, all out-of-gamut colors are clustered in different regions of interest and then each cluster is associated with at least one candidate gamut-mapping function. Here, it is important to choose and assign the best gamut-mapping strategy to a particular cluster and tune the right parameters within each cluster based on their merits and to blend the best local gamut-mapping algorithms to create a final mapping that, when used, can preserve the benefits of each of the algorithms. Hence, this approach is called "merit-based gamut mapping." In this section, we present the algorithmic details on how to build the intelligence using of the control theory to implement the best gamut-mapping strategy for high-quality color printing. Next we discuss some key concepts of the merit-based gamut-mapping algorithm.

Figure 28.21 shows an example of a cusp gamut-mapping algorithm. However, it is easy to envision how this could be extended to other gamut-mapping techniques containing different parameters that require tuning. For a typical cusp gamut-mapping algorithm there are two parameters to tune (hue range and range values). Designers have to go through a trial-and-error process of printing test images with different values for each parameter and then visually decide which values to use. This could also be done using proofing devices such as a display or a separate proofing printer. The first block labeled T represents a transformation from RGB to $L^*a^*b^*$, which is same as the source profile LUT mentioned above. The second block shows the use of an algorithm to map out-of-gamut colors to the boundaries of the printer's gamut. The mapped $L^*a^*b^*$ values can then be used to find the $CMYK$ values using control approaches described in earlier sections.

In Figure 28.22, tuning based on a closed-loop approach is shown for cusp gamut-mapping technique. The same technique can be extended to the tuning of parameters of other gamut-mapping methods for each of the clusters of interest. The block labeled T_2 is a function that maps mapped $L^*a^*b^*$ values of each color into a 1D function. This function could be the $\Delta E2000$ formula, for example, but other options could also be considered. The $\Delta E2000$ formula calculates perceptual distances between two colors, here, the out-of-gamut values and the mapped values for each node color of the current cluster. The block labeled "merit-function" takes all the 1D values available in each cluster and converts them to a single real value that quantitatively represents the "merit" of the gamut-mapping parameters. For example, a mean-squared error function can be used as the merit function. In this case, the merit function determines the mean-squared error of the values calculated using the $\Delta E2000$ function for each node color belonging to the cluster of interest that is to be mapped. The last block, named "Optimization algorithm," serves the

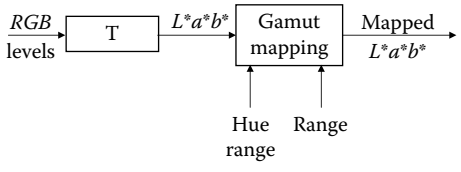

FIGURE 28.21 Open-loop approach to optimize gamut-mapping parameters.

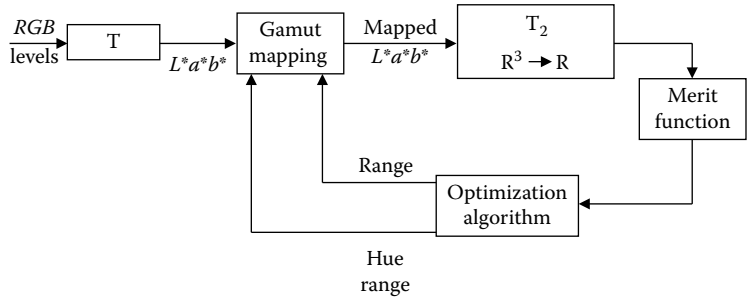

FIGURE 28.22 Optimization of the gamut-mapping algorithm using a closed-loop approach.

purpose of manipulating the gamut-mapping parameters to come up with the "best" merit function for each cluster. The selection of both the mapping function T_2 and the merit function is obviously key; it has to reflect what the designers want to achieve during the optimization process. A unique combination of T_2 and a merit function could be used for all the clusters, but there is also the possibility of selecting different combinations of T_2 and merit functions for different regions of the color space. This has to be designed based on the goals established before running the optimization program. Below, we show an example of T_2 and the merit function.

Suppose the out-of-gamut colors have been partitioned into 3 clusters. The designer is then interested in optimizing the mapping parameters of colors of cluster 1 based on the minimal distance that captures the perceived color differences. For cluster 2, the designer wants to minimize the average of the difference of only lightness, that is, ΔL^*, whereas the goal for cluster 3 is to minimize the mean-squared error of hue preservation of colors. Thus, for cluster 1, T_2 and the merit function are both the $\Delta E2000$ formula. For cluster 2, T_2 is the ΔL^* and the merit function is the minimum of the average of all ΔL^* values. Finally, cluster 3 will use a hue preservation mapping technique for T_2 and the merit function will be the mean-squared error of all the hue preservation values of the colors contained in cluster 3. Next, we show the steps that need to be implemented to create a global gamut-mapping strategy for all the clusters. These can be summarized as follows:

1. Define the number of clusters, N_c, that will be used to group out-of-gamut colors.
2. Run an algorithm (e.g., K-means algorithm in $L^*a^*b^*$ space for a total of 33^3 nodes) that groups the out-of-gamut colors into N_c clusters. The clustering can also be done by specifying gamut regions (e.g., red, blue, yellow, dark etc., regions) without using K-means algorithm.
3. Start working with first cluster, $i = 1$.
4. Run the optimization algorithm for cluster i and determine the best value of the ith merit function.
5. If $i \leq N_c$, then $i = i + 1$ and go to step 4. Otherwise, continue with step 6.
6. Blend all the local results into a global gamut-mapping strategy [28].

Next, we show how to use the multidirectional search method [29] to determine the best value of the ith merit function in step 4 above for a set of M gamut-mapping methods that are available to use from a library. Applications of this algorithm can be found in [9]. First, assume that $G \in 1, 2, \ldots, M$ denotes the set of gamut-mapping algorithms available for any cluster. Any gamut-mapping $j \in G$ could have one or more parameters that could be optimized. For the case when there is a gamut-mapping algorithm without any parameters to tune, the value of the merit function is obtained directly without the need to apply any optimization. If the gamut-mapping algorithms have at least one parameter to tune, then the following process is implemented. Let $\theta_{ij}(k) \in \Re^{p(j)}$ denote the parameters of the jth gamut-mapping algorithm for cluster i at iteration k that could be tuned. Assume that $J(\theta_{ij}(k))$ is continuous in $\theta_{ij}(k)$ and that the gradient $\nabla J(\theta_{i,j}(k))$ exists. There are a set of candidate solutions for each cluster i, denoted as $P^l_{ij}(k) = \{\theta^1_{ij}(k), \theta^2_{i,j}(k), \ldots, \theta^{p(j)+1}_{i,j}(k)\} \subset \Re^{p(j)+1}$, and the method iterates on these candidate solutions

to minimize $J(\theta_{ij}(k))$. The steps to implement this algorithm are described as follows:

1. Define the expansion factor $\gamma_e \in (1, \infty)$ and a contraction factor $\gamma_c = \frac{1}{\gamma_e}$.
2. Start with the first cluster $i = 1$.
3. Start with the first gamut-mapping method $j = 1$.
4. Compute $J(\theta^l_{i,j}(0))$ for $l = 1, 2, \ldots, p(j) + 1$. Proceed with the following steps until a stopping criterion is met.
5. Find the best new vertex of $P^l_{ij}(k)$ using

$$l^* = \arg\ \min\{J(\theta^l_{i,j}(k)) : l = 1, 2, \ldots, p(j) + 1\}$$

   and swap $\theta^1_{i,j}(k)$ and $\theta^{l^*}_{ij}(k)$. Check for the stopping criterion.
6. Rotation step: Compute

$$\theta^l_{rot\,ij}(k) = \theta^1_{i,j}(k) - (\theta^l_{ij}(k) - \theta^1_{i,j}(k))$$

   for $l = 1, 2, \ldots, p(j) + 1$ and $J(\theta^l_{rot\,i,j}(k))$. Continue with 7.
7. Expansion step: If

$$\min\{J(\theta^l_{rot\,ij}(k)) : l = 1, 2, \ldots, p(j) + 1\} < J(\theta^1_{i,j}(k))$$

   then compute

$$\theta^l_{exp\,i,j}(k) = \theta^1_{i,j}(k) - \gamma_e(\theta^l_{ij}(k) - \theta^1_{i,j}(k))$$

   for $l = 1, 2, \ldots, p(j) + 1$ and $J(\theta^l_{exp\,ij}(k))$. Now decide if a new simplex will be formed either with the expansion or rotation and determine the new candidate solution $P^l_{ij}(k + 1)$. If

$$\min\{J(\theta^l_{exp\,ij}(k)) : l = 1, 2, \ldots, p(j) + 1\} < \min\{J(\theta^l_{rot\,i,j}(k)) : l = 1, 2, \ldots, p(j) + 1\}$$

   then expansion is selected and

$$\theta^1_{ij}(k + 1) = \theta^1_{i,j}(k)$$

   and

$$\theta^l_{ij}(k + 1) = \theta^l_{exp\,i,j}(k), l = 1, 2, \ldots, p(j) + 1$$

   Otherwise, let

$$\theta^1_{ij}(k + 1) = \theta^1_{i,j}(k)$$

   and

$$\theta^l_{i,j}(k + 1) = \theta^l_{rot\,ij}(k), l = 1, 2, \ldots, p(j) + 1$$

   Go to step 5.
8. Contraction step: if

$$\min\left\{J(\theta^l_{rot\,i,j}(k)) : l = 1, 2, \ldots, p(j) + 1\right\} \geqslant J(\theta^1_{i,j}(k))$$

   then compute

$$\theta^l_{cont\,ij}(k) = \theta^1_{i,j}(k) + \gamma_c\left(\theta^l_{ij}(k) - \theta^1_{i,j}(k)\right)$$

for $l = 1, 2, \ldots, p(j) + 1$ and $J(\theta^l_{cont\ i,j}(k))$. If

$$\min\left\{J(\theta^l_{cont\ i,j}(k)) : l = 1, 2, \ldots, p(j) + 1\right\} < J(\theta^1_{i,j}(k))$$

form $P^l_{i,j}(k+1)$ by letting

$$\theta^1_{i\,j}(k+1) = \theta^1_{i,j}(k)$$

and

$$\theta^l_{i,j}(k+1) = \theta^l_{cont\ i,j}(k), l = 1, 2, \ldots, p(j) + 1$$

and go to step 5. However, if

$$\min\left\{J(\theta^l_{cont\ i,j}(k)) : l = 1, 2, \ldots, p(j) + 1\right\} \geqslant J(\theta^1_{i,j}(k))$$

then $\theta^l_{i\,j}(k) = \theta^l_{cont\ i,j}(k), l = 1, 2, \ldots, p(j) + 1$ and then go to step 6.

9. If the stopping criterion is satisfied and there are more gamut-mapping methods available, then $j = j + 1$. If there are no more gamut-mapping methods available for cluster i, then let $J^{i,j^*} = \arg\min_j J(\theta^1_{i\,j}(k))$ and check whether there are more clusters available. If so, then let $i = i + 1$ and go to step 3. Otherwise, stop the algorithm.

Note that for each cluster i, the selected gamut mapping to be used for this particular cluster is $j^*$and its corresponding parameters are $\theta^1_{i,j^*}(k)$.

28.3.2.8 Image Simulation

Many factors, quantitative and visual, have to be taken into account when evaluating inversion and profile performance. Profile accuracy, gamut utilization, smoothness of *CMYK* formulations in the multidimensional LUT between nodes, neutral response, are just some of the attributes to consider. Visual evaluation is often subjective but can be very constructive if a comparison with a printed proof can be made. Figures 28.23 and 28.24 show *RGB* images when the color is separated by simulating through two different multidimensional profiles. They each use different *CMYK* separations per pixel to produce the same color. Within each figure we show the image and the four channels after proofing with the ICC profile.

The neutral rendering should contain a smaller amount of black, since the black separation can make rendering of neutrals less smooth. The same could be true for some of the memory colors, for example, sky colors. In the example provided above, the color separated images in Figures 28.23 and 28.24 show differences in the *CMYK* composition and the one with less black utilization may lead to a preferred rendering based on the state of the print engine. Such subtle differences cannot be simulated, but can show as rendering defects. Also, we can sometimes see color contours if the *CMYK* formulations of the profile LUTs are not smooth.

To properly control the choice of *CMYK* separations, one can use the cooperative control strategy described in Section 28.3.2.6 to carefully design the amount of black or other separations along the neutral axis or first at specific regions of the color space (see Figure 28.16) and then propagate the cooperation to other nodes to preserve the smoothness.

28.4 Process Controls

28.4.1 Introduction

In the printing industry, consistency is normally viewed as the function of the internal process, although the 1D, 2D and 3D LUTs (also called control functions) can be updated with measurements on paper at a

FIGURE 28.23 Image simulation for Profile #1 with constrained GCR inverse LUT.

FIGURE 28.24 Image simulatioin for Profile #2 with constrained GCR inverse LUT.

higher update frequency with some kind of time hierarchy. If all 1D, 2D, and 3D loops are implemented in a printing system, then 1D control functions have to be updated more frequently than 2D functions, and 2D more frequently than 3D functions. Generally, the LUTs that manipulate the device *CMYK* values are created with measurements on the output media. The level of performance required for consistency is so high that the update intervals for the 1D loop could approach on a per print basis, which makes it impractical for many high end digital printers. Hence, complex SISO and MIMO control loops are designed to stabilize the internal process in which, these loops maintain background, solid area development, and TRCs of the individual primaries by adjusting various process and digital (image) actuators based on measurements from the output (e.g., toner mass measurements obtained using optical sensors measuring toner state on the photoreceptor instead of the paper) [30]. The loops normally operate at a varying frequency, with some approaching every print. To keep this simple, below we present a level 2 process control design with gain scheduling an algorithm based on the Model Predictive Control theory.

28.4.2 Model Predictive Controller

The process actuators used to control the internal states such as toner mass at different area coverages (e.g., low, mid, and high), toner concentration, etc., have hard limits. For example, in EP printing, photoreceptor charge (both fully exposed and unexposed) cannot exceed certain limits due to cost limitations and image quality considerations. A linear SF controller (with a single MIMO gain matrix) may not satisfy the system-level constraints. This is because when applied to a nonlinear system, a linear controller has the tendency to generate higher actuator values than required. This can lead to undesirable stability problems, particularly when the print engines are operating near their limits.

The level 2 process control system with single MIMO gain matrix typically employs three process actuators (e.g., in iGen3®, the unexposed and exposed voltages of the photoreceptor and the magentic roll bias voltage, and in iGen4®, raster exposure intensity, cleaning voltage, and magnetic roll bias voltage) to track the toner mass on the photoreceptor at low, mid, and high area coverages. The transfer function (i.e., multidimensional input–output characterization data) of the process control system is generally nonlinear. Hence, calculating a new gain matrix for every actuator combination is considered more optimal. This calls for an optimal, adaptive controller design, which can be complex and difficult to implement. Alternatively, we can partition the input–output characterization map into few linear maps, and then schedule a suitable gain matrix during each control action based on some strategy. A pool of gain matrices can be calculated *a priori* from the input–output characterization data.

Below we describe a model predictive control algorithm based on minimization of the Euclidean norm of the tracking error (i.e., the difference between the target DMA [developed mass per unit area] and the measured DMA values). We use the notation introduced in Section 28.3.1.4 and show the modifications needed to implement this controller for the process control loops. Let

$$y(k+1) = \begin{bmatrix} D^l_{k+1} & D^m_{k+1} & D^h_{k+1} \end{bmatrix}^\top \in \Re^3$$

be the measured outputs (DMA measurements of low, mid, and high tones at iteration $k+1$) obtained by an optical sensor, $u(k)$ is the control input $\begin{bmatrix} V_g & V_l & V_b \end{bmatrix}^\top$, which is comprised of the photoreceptor grid voltage, ROS (raster output switch) laser intensity and development bias voltage. Let $r \in \Re^3$ be the reference values $\begin{bmatrix} D^l_r & D^m_r & D^h_r \end{bmatrix}^\top$ at iteration k. We define the tracking error as

$$e(k+j) = \left(\begin{bmatrix} D^l_r & D^m_r & D^h_r \end{bmatrix}^\top - y(k+j) \right)$$

and the system states as described in Equation 28.12. The cost function used here is the same one defined in Equation 28.13 where

$$E^i(k+j) = \left\| \begin{bmatrix} D^l_r & D^m_r & D^h_r \end{bmatrix}^\top - y^i_m(k+j) \right\|$$

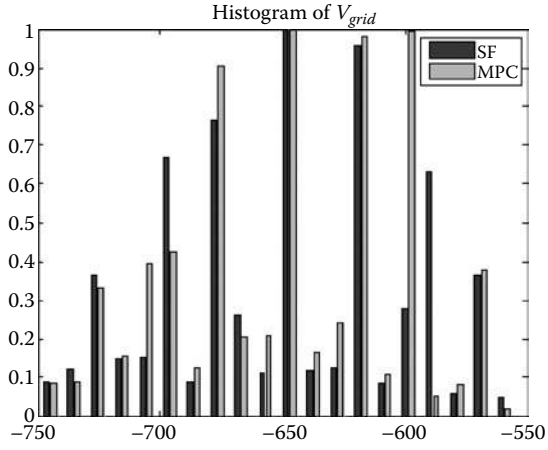

FIGURE 28.25 Histogram of PR grid voltages for SF and MPC level 2 system.

and

$$u^i(k,j) = K^i(j)\left(\begin{bmatrix} D_r^l & D_r^m & D_r^h \end{bmatrix}^\top - y_m^i(k+j)\right)$$

where $K^i(j)$ is the gain matrix for jth iteration and ith plan. The selection of the best plan is computed using Equation 28.14 at each iteration, k. Then, the control input $u(k) = u^i(k,0)$ is applied to the system. Next, we present some simulation results.

Due to computational limits, we determine the gain matrices for all the combinations of poles and Jacobian matrices, first at the nominal actuator input and then use the best gain matrix from this set depending on the path to be taken. We construct a total of 144 gain matrices to chose from. Gain matrices are calculated for different combinations of actuator values and with a MIMO pole placement design. We choose the horizon of $N = 15$. We address the limited excursion of the actuators by using MPC rather than a SF control loop. Figure 28.25 shows a comparison of the grid voltages used by the MPC method and a single gain matrix-based SF controller during numerous simulated transient states (a total of 355 distinct transients). We use weights of $w_1 = 1, w_2 = 0$ to emphasize the minimization of the error between

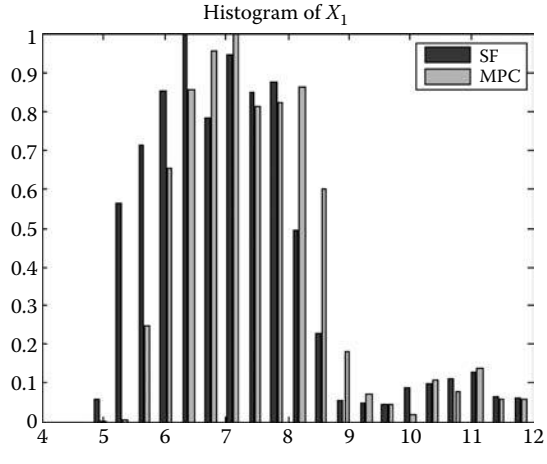

FIGURE 28.26 Histogram of ROS laser intensity for SF and MPC level 2 system.

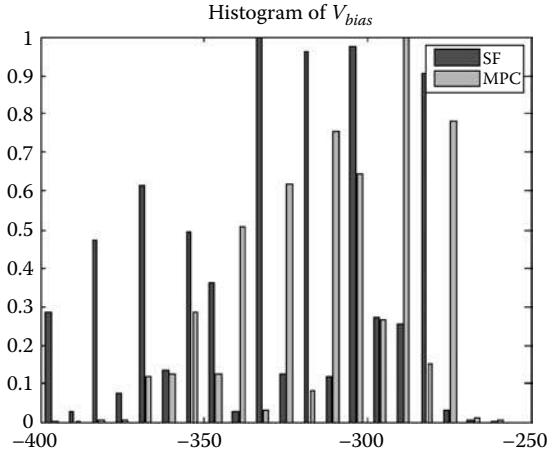

FIGURE 28.27 Histogram of development bias voltage for SF and MPC level 2 system.

the targets and the measurements. Figure 28.26 shows a comparison of the ROS laser intensity values and Figure 28.27 shows a comparison of development bias voltage used by the two methods.

It can be seen in these figures that the grid voltage excursion is not very different from the two approaches. However, there is improvement in the limits used for laser intensity and bias voltages. Since these actuators are used in combination with each other to maintain toner mass at the desired targets, the MPC approach always tries to maintain the process at the "sweet spot," and hence the control loop can be more robust to process excursions. To make this point more apparent, we have plotted the 3D excursion plot of actuators for a DMA target setpoint of [0.0789, 0.2546, 0.4525] in Figure 28.28. Notice that the MPC approach takes a more direct path toward the final actuator values required to meet the DMA targets than SF. Figures 28.29, 28.30, and 28.31 show the histograms of the three actuators for this setpoint. We can clearly see that there is less excursion of actuators, due to gain scheduling using MPC than due to SF. Figure 28.32 shows that error plots for MPC and SF control loop for this setpoint.

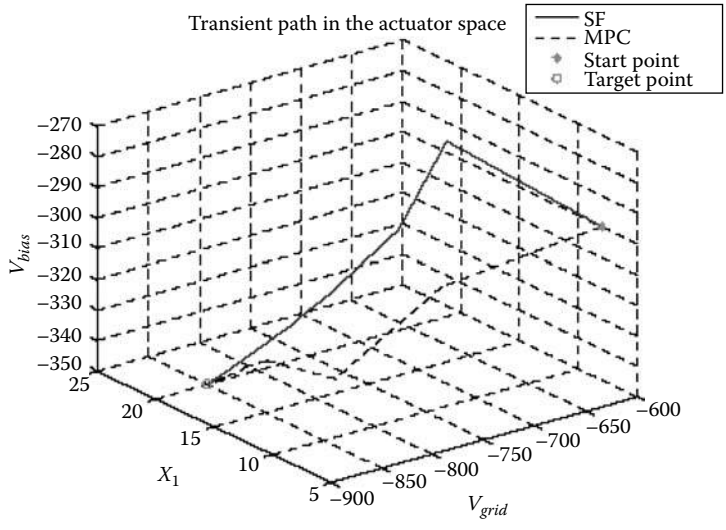

FIGURE 28.28 Transient actuator states for SF and MPC.

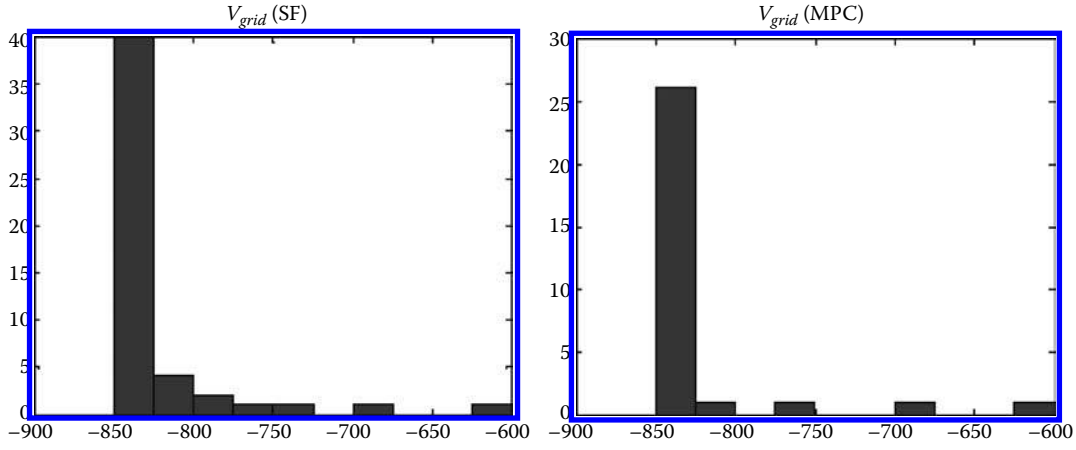

FIGURE 28.29 Histogram of grid voltages for a single DMA setpoint.

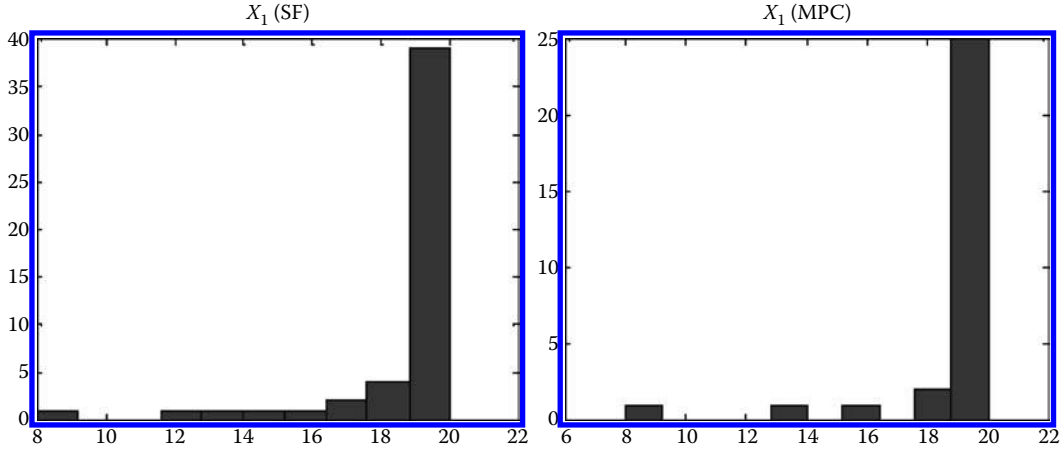

FIGURE 28.30 Histogram of laser intensity for a single DMA setpoint.

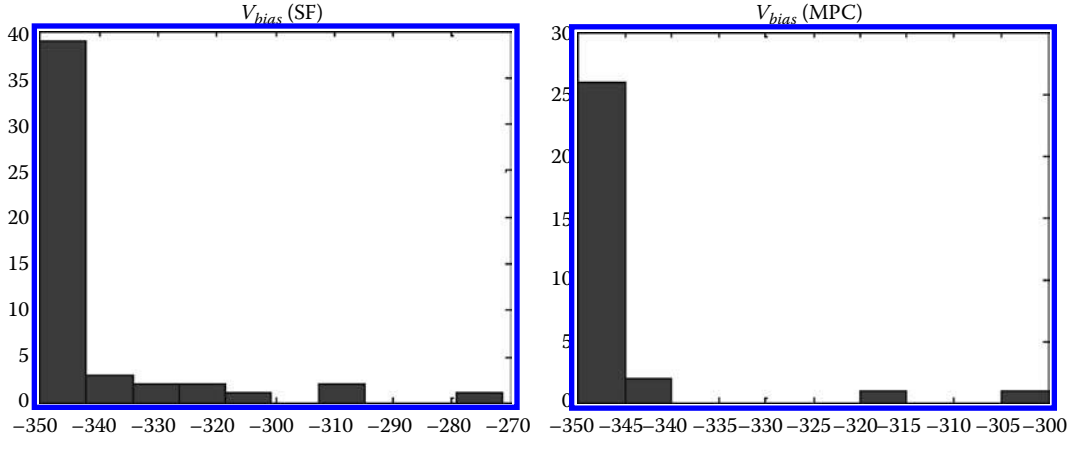

FIGURE 28.31 Histogram of bias voltage for a single DMA setpoint.

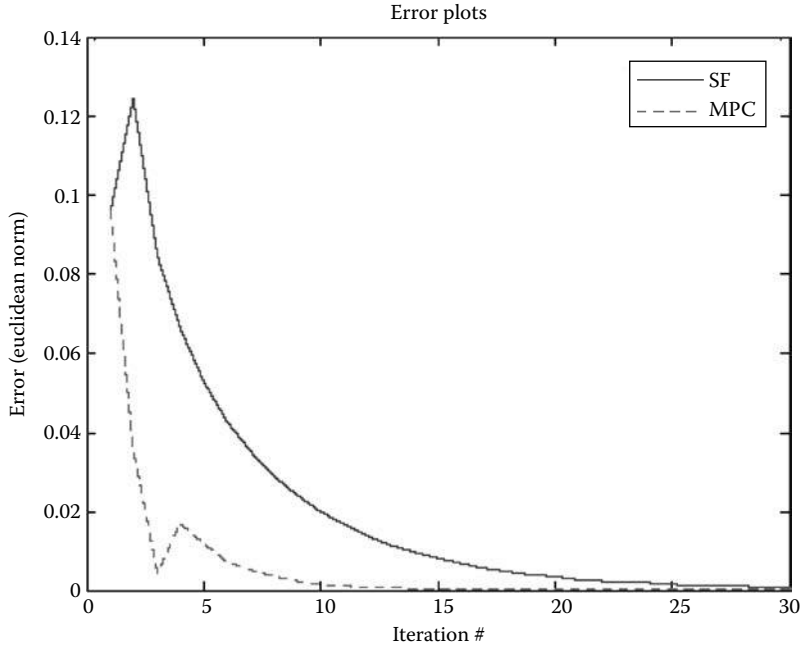

FIGURE 28.32 Error plots for a single DMA setpoint.

In summary, since the printing process is nonlinear, the dynamic range of the actuators becomes large with a single gain matrix solution. A multiple gain solution coupled with a systematic way to automatically switch between them during the closed-loop control action (i.e., measurement-processing-actuation cycle) gives an improved performance for actuators with a reduced dynamic range. Switching between gain matrices happens inside the MPC in a smooth fashion provided the number of gain matrices is reasonably large. This can help to solve many undesirable stability problems, actuator overshoots, etc., particularly when the print engines are operating near their limits. Using this methodology with automatic gain scheduling, when setpoints change (e.g., when the media changes, or when DMA targets change due to actions from a higher level loops [31]), the closed-loop system can still perform with improved robustness without going unstable.

28.5 Conclusion

Although the digital print process is remarkably challenging because it involves many new actuators (e.g., process and image actuators) these actuators provide the opportunity to apply a wide range of control techniques. Also, many new challenges in production color (e.g., to deliver near/better than offset quality at low running cost and high productivity) opened the door for us to inject new control theory such as MIMO SF, pole-placement design, LQR, model predictive controls, and cooperative controls. Everyday, numerous Xerox® printing systems (iGen3, iGen4, Docucolor 7002, Docucolor 8002, Docucolor 5000, Docucolor 8000, iGen4 220, Perfecting Press, Xerox Color 800/1000 etc.) produce high-quality prints with these control methods to generate several billion dollars revenue.

In this chapter, we have emphasized how to achieve color consistency using advanced algorithms. There are even more opportunities that exist for improving the color quality; images are comprised of fine dots with a mixture of four or more primaries and digital printing offers the abilities to write and erase them as we please. Darkness levels are created by changing the size of the dots from nothing

(background) to completely covering a given area (solid). For low-density images (highlights), the dots are small and distinct. For high-density images (shadows) the untoned areas are cusp-shaped, representing the untoned spaces between the dots. In an 8-bit imaging system, a mixture of primaries is created in the DFE by changing the digital levels for each between 0 to 255 and higher for high-resolution imaging. The image path then halftones these levels, converting them into various dot patterns. Thus, a correct mixture of primaries with varying types of dots can not only create accurate and high-quality color, but also can open up opportunities to create spectrally matched output to enable high-quality device, illumination, and observer-independent color.

Other opportunities for digital printing include: print shop automation and optimal system design with control theory, networked controls, high performance paper path controls with state synchronization, and state-based scheduling, active diagnosis and defect analysis, and multiprint matching in a multimachine print environment. Thus, modern controls continue to play a significant role in improving the performance and the economics of the imaging systems.

References

1. L. K. Mestha and S. Dianat, *Control of Color Imaging System*. CRC Press, Boca Raton, Florida, 2009.
2. H. Kipphan, *Handbook of Print Media*. Springer, Berlin, 2001.
3. K. Ogata, *Discrete-Time Control Systems*. Prentice-Hall, New Jersey, 1987.
4. D. Viassolo, S. A. Dianat, L. K. Mestha, Y. R. Wang, Practical algorithm for the inversion of an experimental input-output color map for color correction, Optical Engineering 42(3), pp. 625–631.
5. L. K. Mestha, R. E. Viturro, Y. R. Wang, and S. A. Dianat, Gray balance control loop for digital color printing systems, in *Proceedings of the 2000 IS&T Conference on Digital Printing Technologies (NIP21)*, September 2005.
6. A. E. Bryson, *Dynamic Optimization*. Addison-Wesley, 1999.
7. A. Gil and L. K. Mestha, Spot color controls and method, US Patent Application No. 20080043264, February 2008.
8. L. K. Mestha and O. Y. Ramirez, On-line model prediction and calibration system for a dynamically varying color reproduction device, US Patent No. 6809837, October 2004.
9. K. Passino, *Biomimicry for Optimization, Control, and Automation*. Springer-Verlag, London, 2004.
10. C. E. Garcia, D. M. Prett, and M. Morari, Model predictive control: Theory and practice—a survey, *Automatica*, vol. 25(3), pp. 335–348, 1989.
11. G. Sharma, *Digital Color Imaging Handbook*. New York: CRC Press, 2003.
12. R. Balasubramanian and R. Eschbach, Design of UCR and GCR strategies to reduce moire in color printing, in *IS&T PICS Conference,* 1999, pp. 390–393.
13. R. Balasubramanian and R. Eschbach, Reducing multi-separation color moire via a variable undercolor removal and gray-component replacement strategy, *Journal of Imaging Science and Technology*, vol. 45(2), pp. 152–160, March/April 2001.
14. G. Laporte, The vehicle routing problem: An overview of the exact and approximate algorithms, *European Journal of Operational Research*, vol. 59, pp. 345–358, 1992.
15. J. Bellingham, A. Richards, and J. How, Receding horizon control of autonomous aerial vehicles, in *Proceedings of the ACC*, Anchorage, Alaska, May 2002, pp. 3741–3746.
16. W. Li and C. G. Cassandras, Stability properties of a receding horizon controller for cooperating UAVs, in *43rd IEEE CDC*, Paradise Island, Bahamas, December 2004, pp. 2905–2910.
17. R. Beard, T. McLain, and M. Goodrich, Coordinated target assignment and intercept for unmanned air vehicles, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 2581–2586.
18. M. G. Earl and R. D'Andrea, Iterative MILP methods for vehicle control problems, in *43rd IEEE CDC*, Paradise Island, Bahamas, December 2004, pp. 4369–4374.
19. T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, On a decentralized active sensing strategy using mobile sensor platforms in a network, in *43rd IEEE CDC*, Paradise Island, Bahamas, December 2004, pp. 1914–1919.
20. P. Ogren, E. Fiorelli, and N. Leonard, Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment, *IEEE TAC*, vol. 49, no. 8, pp. 1292–1302, August 2004.

21. J. Cortés, S. Martínez, T. Karatas, and F. Bullo, Coverage control for mobile sensing networks, *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.
22. Z. Tang and U. Ozguner, Sensor fusion for target track maintenance with multiple UAVs based on Bayesian filtering method and hospitability map, in *42nd IEEE CDC*, Maui, HI, December 2003, pp. 19–24.
23. C. Zhang, Q. Sheng, and R. Ordónez, Notes on the convergence and applications of surrogate optimization, in *Fifth International Conference on Dynamical Systems and Differential Equations*, Pomona, CA, June 2004, pp. 1–9.
24. D. A. Castanon and C. G. Cassandras, Cooperative mission control for unmanned air vehicles, in *Proceedings of the AFOSR Workshop on Dynamic Systems and Control*, Pasadena, CA, August 2002, pp. 57–60.
25. L. K. Mestha, Y. R. Wang, A. E. Gil, M. Maltz, and R. Bala, A restricted black GCR–UCR strategy for creating pleasing colors, US Patent Filed No. 20070410, December 2007.
26. P. Zolliker and K. Simon, Continuity of gamut mapping algorithms, *Journal of Electronic Imaging*, vol. 15(1), January–March 2006.
27. J. Morovic, *Color Gamut Mapping*. John Wiley and Sons, Inc., NY, 2007.
28. M. Maltz, S. J. Harrington, and S. A. Bennett, Blended look-up table for printing images with both pictorial and graphical elements, US Patent No. 5734802, March 1997.
29. V. Torczon, On the convergence of the multidirectional search algorithm, *SIAM Journal on Optimization*, vol. 1(1), pp. 123–145, 1991.
30. P. K. Gurram, S. A. Dianat, L. K. Mestha, and R. Bala, Comparison of 1-D, 2-D and 3-D printer calibration algorithms with printer drift, in *IS&T The International Conference on Digital Printing Technologies (NIP21)*, September 18–23 2005, pp. 505–510.
31. L. K. Mestha, P. K. Gurram, A. E. Gil, and P. Ramesh, Algorithms and methods to match color gamuts for multi-machine matching, US Patent No. 20081152, December 2008.

29

The Construction of Portfolios of Financial Assets: An Application of Optimal Stochastic Control

Charles E. Rohrs
Rohrs Consulting

Melanie B. Rudoy
Massachusetts Institute of Technology

29.1 Introduction

The notions of financial engineering have their roots in disciplines that have long been part of the subject matter studied by control theorists. In particular, many of the main results in financial economics are direct applications of the use of random variables, stochastic differential and difference equations and discrete- and continuous-time stochastic control theory. Yet these results have been developed almost completely separately from the developments in the controls field with very few researchers publishing in both areas.

In this chapter, we look at important ideas in the development of portfolio optimization—the problem of using a statistical characterization of asset returns to optimally mix a set of assets so that the whole

performs better than the parts. Consider a portfolio as a weighted linear combination of assets. The mean return of a portfolio is the weighted linear combination of the mean returns of the assets but the variance of the return of the portfolio is less than the weighted combination of the variances of the assets unless the assets are perfectly correlated. The variance of return is a natural measure of risk. The use of correlations between returns to reduce the risk associated with a portfolio of stocks is properly called *hedging*. After models for the evolution of asset returns in time are introduced in Section 29.3, we show in Sections 29.4 and 29.5 that one can perform hedging using the correlations of assets across time as well as across the different assets themselves.

The seminal papers that address these problems are very readable and are accessible electronically.* The one period problem of Section 29.2 was proposed and solved by Markowitz [1]. The expansion of those results to model the returns of individual assets was given by Sharpe [2].† The resulting Capital Asset Pricing Model is developed in Section 29.2.5. Samuelson [3] introduced optimal control techniques to solve the discrete-time portfolio problem where returns are considered after a number of possible chances to adjust the composition of a portfolio. Merton [4] produced the breakthrough reformulation and solution of the optimal portfolio problem in continuous-time. All four of the aforementioned pioneers received the Nobel Prize in Economics for their contributions. After discussing models of assets returns in Section 29.3, we show the discrete-time results in Section 29.4 and the continuous-time results in Section 29.5 on somewhat more complex models than that were used in the original papers.

29.2 Markowitz Mean-Variance Portfolio Theory

The practice of modeling security returns as random variables began in earnest in 1952 with the work of Harry Markowitz [1].‡ Markowitz's key insight was to model the one-period returns of a set of securities as random variables and to use the mean return of a portfolio§ of holdings as a measure of reward and the variance of the portfolio return as a measure of the risk associated with holding that portfolio.

Consider the following simple scenario. An investor chooses between investing an amount of wealth W, all in security A, all in security B or in a portfolio with a 50–50 mix of A and B. The return on A is the random variable r_A, implying that an investment of W in A today will be worth $(1 + r_A)W$ after one period. Likewise, the return on B is the random variable r_B. Assume that r_A and r_B have identical means, μ, and identical variances, σ^2, while the correlation coefficient between r_A and r_B is ρ, so that

$$E\left[(r_{\mathrm{A}} - \mu)(r_{\mathrm{B}} - \mu)\right] = \rho\sigma^2.$$

A simple calculation shows that the return on the 50–50 portfolio is

$$r_{\mathrm{P}} = 0.5r_{\mathrm{A}} + 0.5r_{\mathrm{B}} \text{ with mean } \mu \text{ and variance } 0.5(1 + \rho)\sigma^2.$$

The key observation is that, unless the returns on the two securities are perfectly correlated, the portfolio achieves the same mean return with smaller variance than an investment that is either all in A or all in B.

* For an introduction to the concepts of financial engineering at a somewhat simpler level than pursued here, we recommend [5]. The major results of the field are given in [6]. While [6] is a collection of Merton's previously published papers, it reads more like a text with an excellent development of the underlying mathematics and is the single most important resource in learning the theoretical underpinnings of the field. Two other popular texts are [7,8]. Both of these emphasize an important second approach to asset pricing relying heavily on the use of martingales in their mathematically sophisticated developments.

† Lintner [9] and Mossin [10] are generally credited with independently and essentially concurrently producing similar results to Sharpe [2].

‡ Unfortunately, the technical world failed to follow up a presciently sophisticated early work by Bachelier [11] as it lay dormant until its rediscovery in the 1950s. According to Merton [6], the rediscovery is "generally credited to Samuelson via L.J. Savage."

§ A portfolio of assets is simply a collection of assets. The percentage of wealth invested in each asset is given by a weight; the vector of weights determines the composition of the portfolio.

If N securities whose returns are uncorrelated could be found, the return on an equally weighted portfolio would have mean equal to the averaged mean of the individual securities' returns while the variance would be reduced by a factor of N from the averaged variance of the individual securities' returns. Unfortunately, securities with uncorrelated returns are hard to find. Returns on securities (with positive means) tend to be positively, but not perfectly, correlated so that while diversifying a portfolio is clearly useful, there are diminishing advantages to adding more and more securities.*

We next show Markowitz's formulation and solution to the problem of choosing an optimally weighted portfolio of securities given that the one-period returns of such securities form a random vector with known mean vector and covariance matrix. As in all engineering solutions, it is the modeling process that requires a *leap of faith*. The results of any implementation and the knowledge gained from any modeling process depend on the legitimacy and the accuracy of the model as well as any vagaries inherent in experiencing only one sample function from a statistical characterization.

Let $\boldsymbol{r}$ denote a random vector of returns corresponding to the p securities available to the investor. Suppose $\boldsymbol{r}$ has known mean vector, $\boldsymbol{\mu}$, and covariance matrix, Ψ. The investor chooses a portfolio weight vector, $\boldsymbol{w}$, where w_i represents the fraction of total initial investment put into the ith security. For such an interpretation of $\boldsymbol{w}$, we include a constraint, usually called the budget constraint (BC), that the elements of $\boldsymbol{w}$ sum to one.

Unless explicitly excluded, negative values are allowed in the weight vector as the investor may decide to *short* a security. An investor shorts a security by initially borrowing the security from a firm that owns it and immediately selling it to generate income that can be used to fund investments in other securities. Later, the shorted security must be bought back at the new price and returned to the institution from which it was borrowed. Mechanisms are in place for both institutional and even small individual investors to short stocks and other securities. In general, throughout this work, transaction costs† are ignored as we are interested in the fundamental ways that finance theory uses tools from control theory. Including transaction cost is a very important detail and a subject of much ongoing work.

29.2.1 The Single Period Mean-Variance Problem

The goal of a Markowitz mean-variance investor is to maximize the mean portfolio return, given a limit on the corresponding variance of the portfolio return. The variance constraint can be taken as an equality constraint as greater variance will allow greater mean and *vice versa*. An invertibility condition connected with the covariance matrix will be assumed in the solution and explained thereafter. Note that there is an equivalent dual problem that minimizes the variance of a portfolio's return, given the portfolio's mean return. Formally, the primal problem can be expressed as follows:

$$\left.\begin{array}{c} \boldsymbol{w}^* = \arg \max_{\boldsymbol{w}} \ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu} \\ \text{subject to } \boldsymbol{w}^{\mathrm{T}}\Psi\boldsymbol{w} = \sigma_0^2 \\ \boldsymbol{w}^{\mathrm{T}}\mathbf{1} = 1 \end{array}\right\} \quad P_0,$$

where $\mathbf{1}$ is a vector of all 1's and the variance σ_0^2 defines the allowable *risk budget* set by the investor. The Markowitz mean-variance portfolio optimization framework does not assume that the return distributions are jointly Gaussian or any other distribution; it simply states that the investor makes decisions based only on the first and second moments of the return distribution of the underlying securities.

* We will consider one special asset called a *risk-free* asset with a deterministic return so that its return is uncorrelated with the returns of all other assets. We will see that such an asset imparts a very special structure to the problems considered.

† Transaction costs are the costs involved in trading securities such as brokerage fees, exchange fees, and the effects of the bid-ask spread. At a single point in time, the price at which a security can be bought (the ask price) is usually slightly higher than the price at which a security can be sold (the bid price).

29.2.2 The Solution to the Single Period Mean-Variance Problem

The solution to Problem P_0 is now derived. Instead of utilizing two Lagrange multipliers for the two equality constraints, we first enforce the BC by applying the following affine transformation on $\mathbb{R}^P \rightarrow \mathbb{R}^{P-1}$ to the portfolio weight vector:

$$\boldsymbol{w} = \boldsymbol{c} + D\boldsymbol{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_{p-1} \\ 1 - \sum_{i=1}^{p-1} v_i \end{bmatrix}, \quad \text{where} \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad D = \begin{bmatrix} I_{p-1} \\ -\mathbf{1}^{\mathrm{T}} \end{bmatrix}. \tag{29.1}$$

Problem P_0 can now be rewritten as Problem P_1:

$$\left.\begin{array}{c} \boldsymbol{w}^* = \boldsymbol{c} + D\boldsymbol{v}^* \text{ with } \boldsymbol{v}^* = \arg \max_{\boldsymbol{v}} (\boldsymbol{c} + D\boldsymbol{v})^{\mathrm{T}} \boldsymbol{\mu} \\ \text{subject to } (\boldsymbol{c} + D\boldsymbol{v})^{\mathrm{T}} \Psi (\boldsymbol{c} + D\boldsymbol{v}) = \sigma_0^2 \end{array}\right\} \quad P_1.$$

Now introducing the Lagrange multiplier, λ, we arrive at Problem P_1':

$$\boldsymbol{v}^*, \lambda^* = \arg \max_{\boldsymbol{v}, \lambda} (\boldsymbol{c} + D\boldsymbol{v})^{\mathrm{T}} \boldsymbol{\mu} - \lambda \left((\boldsymbol{c} + D\boldsymbol{v})^{\mathrm{T}} \Psi (\boldsymbol{c} + D\boldsymbol{v}) - \sigma_0^2 \right) \Bigg\} \quad P_1'.$$

Differentiating with respect to $\boldsymbol{v}$ and assuming the needed invertibility yields the solution

$$\boldsymbol{v}^* = \frac{1}{2\lambda^*} (D^{\mathrm{T}} \Psi D)^{-1} D^{\mathrm{T}} (\boldsymbol{\mu} - 2\lambda^* \Psi \boldsymbol{c}). \tag{29.2}$$

Thus the optimal portfolio weight vector is given by

$$\begin{aligned} \boldsymbol{w}^* &= \boldsymbol{c} + D\boldsymbol{v}^* = \boldsymbol{c} + \frac{1}{2\lambda^*} D (D^{\mathrm{T}} \Psi D)^{-1} D^{\mathrm{T}} (\boldsymbol{\mu} - 2\lambda^* \Psi \boldsymbol{c}) \\ &= \frac{1}{2\lambda^*} \boldsymbol{f} + \boldsymbol{g}, \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{f} &= D (D^{\mathrm{T}} \Psi D)^{-1} D^{\mathrm{T}} \boldsymbol{\mu}, \\ \boldsymbol{g} &= \boldsymbol{c} - D (D^{\mathrm{T}} \Psi D)^{-1} D^{\mathrm{T}} \Psi \boldsymbol{c}. \end{aligned}$$

Lastly, the relationship between λ^* and the variance constraint σ_0^2 is given by the following quadratic equation in $(\lambda^*)^{-1}$:

$$\begin{aligned} \sigma_0^2 &= (\boldsymbol{c} + D\boldsymbol{v})^{\mathrm{T}} \Psi (\boldsymbol{c} + D\boldsymbol{v}) \\ &= (\lambda^*)^{-2} 0.25 \boldsymbol{f}^{\mathrm{T}} \Psi \boldsymbol{f} + (\lambda^*)^{-1} \boldsymbol{f}^{\mathrm{T}} \Psi \boldsymbol{g} + \boldsymbol{g}^{\mathrm{T}} \Psi \boldsymbol{g}. \end{aligned}$$

For every value of σ_0^2, there are two values of λ^*, each corresponding to a different value for the mean return. The lower mean return value is discarded.†

The invertibility condition on Ψ assumed above is satisfied if all assets are risky and no asset is redundant, that is, all asset returns have nonzero variance and no asset is a linear combination of the other assets.

† The value of the Lagrange multiplier at the optimum is called the shadow price of risk; it indicates the amount the mean will be increased if the allowable variance is incrementally increased, that is, $\mathrm{d}(w^{*T}\mu) \Big/ \mathrm{d}\left(\sigma_0^2\right) = \lambda^*$.

29.2.3 Including a Risk-Free Asset

In order to satisfy the invertibility condition of Section 29.2.2, all of the assets included in the portfolio are required to be risky (i.e., exhibit nonzero return variance). However, it is often desirable to include a *risk-free* asset (an asset whose return has zero variance) in the portfolio, and therefore additional care must be taken to ensure the resulting portfolio choice problem is well formed. The motivation for including a risk-free asset is twofold. First, the presence of a risk-free asset changes the problem in interesting ways, and second, there are assets such as a U.S. Treasury bonds maturing at the period's end whose nominal returns are well modeled as risk free.

Augmenting the vector of asset returns to include a risk-free security produces the following forms:

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_0 \\ r_{\mathrm{f}} \end{pmatrix} \quad \Psi = \begin{pmatrix} \Psi_{\mathbf{0}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix} \quad \text{so that } D^{\mathrm{T}}\Psi D = \Psi_{\mathbf{0}} \text{ and } \Psi \boldsymbol{c} = 0$$

and

$$\boldsymbol{v}^* = \frac{1}{2\lambda^*}\Psi_{\mathbf{0}}^{-1}(\boldsymbol{\mu}_0 - \mathbf{1}r_{\mathrm{f}}) = \frac{1}{2\lambda^*}\Psi_{\mathbf{0}}^{-1}\boldsymbol{\mu}_{\mathrm{ex}}; \quad \boldsymbol{w}^* = \begin{bmatrix} \boldsymbol{v}^* \\ 1 - \boldsymbol{v}^{*\mathrm{T}}\mathbf{1} \end{bmatrix}.$$

The vector $\boldsymbol{\mu}_{\mathrm{ex}} = (\boldsymbol{\mu}_0 - \mathbf{1}r_{\mathrm{f}})$ is called the *mean excess return.* The corresponding optimal portfolio variance and mean are then found to be

$$\sigma_0^2 = \frac{1}{(2\lambda^*)^2}\boldsymbol{\mu}_{\mathrm{ex}}^{\mathrm{T}}\Psi_{\mathbf{0}}\boldsymbol{\mu}_{\mathrm{ex}}; \quad \mu^* = \frac{1}{2\lambda^*}\boldsymbol{\mu}_{\mathrm{ex}}^{\mathrm{T}}\Psi_{\mathbf{0}}^{-1}\boldsymbol{u}_{\mathbf{ex}} + r_{\mathrm{f}} = \sqrt{\sigma_0^2}\frac{\boldsymbol{\mu}_{\mathrm{ex}}^{\mathrm{T}}\Psi_{\mathbf{0}}^{-1}\boldsymbol{\mu}_{\mathrm{ex}}}{\sqrt{\boldsymbol{\mu}_{\mathrm{ex}}^{\mathrm{T}}\Psi_{\mathbf{0}}\boldsymbol{\mu}_{\mathrm{ex}}}} + r_{\mathrm{f}}.$$

The notation $\sqrt{\sigma_0^2}$ is used to emphasize that, in the presence of a risk-free asset, the optimal portfolio's mean return is an affine function of the allowed *standard deviation* (not the variance) of the portfolio's return. As the standard deviation changes, λ^* changes, thus altering the scaling but not the direction of the vector $\boldsymbol{v}^*$. In other words, as the level of risk varies, the relative proportion of each risky asset in the optimal portfolio remains constant. Thus optimal portfolios across all variance levels consist of various weightings of only two funds or combinations of assets—one degenerate fund composed of only the risk-free asset and the other fund consisting of the other risky assets in the proportion given by a unit vector in the direction of $\boldsymbol{v}^*$. We call this special portfolio the *tangent portfolio*. This idea is known as the *two-fund separation theorem* as described by James Tobin† in [12].

The plot of the means versus the standard deviations of the returns for all optimal portfolios that may contain a risk-free asset results in a straight line called the capital market line (CML). (The CML is illustrated for the following example in Figure 29.1.)

Example: Annual Returns on Stocks and Bonds in the United States

Suppose an investor with a one-year investment horizon must decide how to allocate his wealth between two risky assets, one corresponding to the S&P 500 Index and the other to the 10-Year U.S. Treasury Bond; the return random variables are denoted as r_S and r_B, respectively. Here, the U.S. Treasury bond is treated as a risky asset due to the fact that the face value of the bond fluctuates day by day and when cashed in at the end of one year, it may be worth more or less than the original purchase price plus interest. (In this problem, however, the 1 Year U.S. Treasury Bond can be considered risk-free as its payout in one year is completely known with very small chance of default. Note that this problem uses *nominal* returns although it may be of more interest to work with *real* returns adjusted for inflation.) Historical analysis of past daily returns from 1925 to 2000 obtained

† James Tobin was also a recipient of the Nobel Prize in Economics.

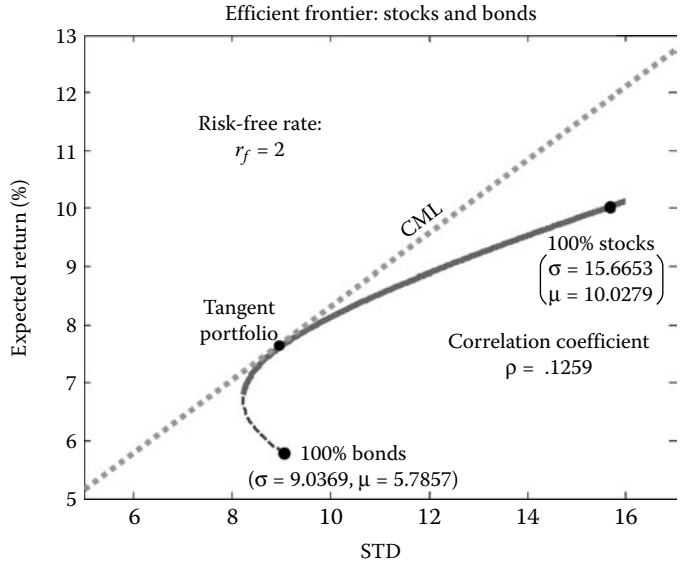

FIGURE 29.1 Mean-variance efficient frontier and CML for a financial universe consisting of stocks (S&P 500 Composite Index), bonds (10-year U.S. Government Treasury Bond) and a risk-free asset with rate $r_f = 2\%$. Returns are given in annual percentages.

from the CRSP database produces the following estimates of the first- and second-order statistics for the annual returns of these two assets.

$$\boldsymbol{\mu} = \begin{bmatrix} E(r_S) \\ E(r_B) \end{bmatrix} = \begin{bmatrix} 10.03 \\ 5.79 \end{bmatrix}; \quad \Psi = \begin{bmatrix} \text{var}(r_S) & \text{cov}(r_S, r_B) \\ \text{cov}(r_S, r_B) & \text{var}(r_B) \end{bmatrix} = \begin{bmatrix} 245.40 & 17.83 \\ 17.83 & 81.67 \end{bmatrix}.$$

As evidenced by the nondiagonal structure of the covariance matrix, Ψ, the two assets are correlated, with a correlation coefficient of $\rho = 0.13$. The set of efficient (optimal) portfolios that results for this example as the allowable risk parameter varies is shown by the solid line in Figure 29.1. This curve is referred to as the *efficient frontier*. The dashed line highlights portfolios that do not maximize return for a given level of risk, but do minimize risk for a given level of return.

When a risk-free asset is also available, the optimal portfolios at various levels of risk (as given by σ_0) fall on the CML as explained in Section 29.2.3. For the example given here, a representative CML for a risk-free rate of 2% is shown as a dotted line in Figure 29.1, and the tangent portfolio is highlighted. If the investor is allowed to be leveraged, he may borrow money from the risk-free asset in order to achieve the set of operating points on the CML to the right of the tangent portfolio. The slope of the CML as given by

$$S = \frac{\mu^* - r_f}{\sqrt{\sigma_0^2}},$$

is referred to as the Sharpe ratio.

The Sharpe ratio can be formed for any portfolio, optimal or not, using its own estimated mean and standard deviation. The Sharpe ratio is used as a benchmark for portfolio performance since, in theory, all optimal portfolios obtain the same maximal Sharpe ratio. Optimal portfolio theory has the (somewhat philosophical) implication that, at least in security returns, the action that is systematically rewarded in the market with increased mean returns is taking on increased risk. The Sharpe ratio is used to adjust for this in comparing portfolios with different levels of (historically estimated) risk.

On a separate note, if the BC is removed, it implies that the investor can borrow for free (i.e , $r_f = 0$), resulting in an efficient frontier that is a straight line through the origin. Without a BC, only the variance constraint determines the *size* (i.e., *scale* or degree of *net leverage*) of the portfolio.

29.2.4 Use of Hedging

Consider an investor who has been satisfied with the variance of returns that results from an all bond strategy of rolling 10-Year U.S. Treasury bonds once a year.* Assuming that the past provides a statistically accurate prediction of the future, this investor can increase his mean return from about 5.8% to about 7.5% while slightly *reducing* the variance of his annual return by optimally splitting his investment between stocks and bonds. The variance is reduced even though he has mixed in some of the higher variance returns of stocks. The risk in the return of the stocks is *hedged* by the return of the 10-year bond. Many words in the world of finance have their original meanings eroded through their misuse in marketing and lay financial reporting. *Hedging* is one of those words that have taken on many connotations, both positive and negative. (The word *arbitrage* has been similarly hijacked.) In this chapter, the word *hedging* is used to connote improving the performance of a portfolio by combining assets that are not perfectly correlated. The above example shows hedging across assets in one time period. After a discussion of how to model the distributions of asset returns across time, we show that an optimal control problem can be formulated and solved with the resulting portfolios displaying hedging across time. That is, the portfolios take advantage of the fact that returns of the assets are not perfectly correlated across time. This allows an investor to hold portfolios in which the sum of the per-period portfolio return variances exceed the total variance constraint, yet the total variance constraint is satisfied due to the existence of negative interstage portfolio return correlations. Thus the per-stage portfolio risk levels are hedged with the knowledge that interperiod portfolio returns will not be perfectly correlated.

29.2.5 The Capital Asset Pricing Model

By assuming equilibrium between the quantity of each asset supplied and the quantity demanded, Lintner [9] argued that the portfolio weights in the tangent portfolio must be equal to the relative value of each risky asset in the market. With this interpretation, the tangent portfolio is interchangeably called the *market portfolio.* In [2], Sharpe showed that, in this special model, if r_{M} is the random variable for the return of the market portfolio with mean μ_{M} and r_i is (as before) the random variable for the return of the ith security with mean μ_i, then, *for each security i*,

$$\mu_i = r_{\mathrm{f}} + \beta_i(\mu_{\mathrm{M}} - r_{\mathrm{f}}) \quad \text{with } \beta_i = \frac{\mathrm{cov}(r_i, r_{\mathrm{M}})}{\mathrm{var}(r_{\mathrm{M}})}. \tag{29.3}$$

This powerful result is usually referred to as the *capital asset pricing model* (CAPM)† or, less often but more specifically, the principle of *linear asset pricing.* The linear relationship between any security's excess mean return and the excess mean return on the market portfolio is called the security market line (SML) for that security (not to be confused with the CML defined above). The beta of a security (or portfolio of securities) measures the *systematic risk* of that security. The variance of the returns of a security measures the *total risk* of that security, both systematic risk and idiosyncratic risk. The CAPM theory says that the market only rewards an investor for taking on systematic risk as idiosyncratic risk can be diversified away with portfolio optimization.

* By rolling the 10-year bond, we mean that the investor buys a new 10-year bond every year and sells the bond bought the previous year.

† There are now other pricing models of capital assets, but CAPM was the first and so it lays claim to the rather generic name and remains the paradigm.

To derive Equation 29.3, one must first use market equilibrium to argue that every risky asset must be present in the tangent (market) portfolio.* One can then postulate a simple portfolio that is a convex combination of security i and the market portfolio, that is, a portfolio with expected return $\mu = \alpha\mu_i + (1-\alpha)\mu_M$ and standard deviation

$$\sigma = \sqrt{\alpha^2\sigma_i^2 + (1-\alpha)^2\sigma_M^2 + 2\alpha(1-\alpha)\text{cov}(r_i, r_M)}. \tag{29.4}$$

$$\text{At} \quad \alpha = 0, \quad \frac{d\sigma}{d\mu} = \frac{(d\sigma/d\alpha)}{(d\mu/d\alpha)} = \frac{\mu_i - \mu_M}{(1/\sigma)\left(-\sigma_M^2 + \text{cov}(r_i, r_M)\right)} = \frac{\mu_i - \mu_M}{\sigma_M(-1+\beta_i)}.$$

The last equality comes from realizing that, at $\alpha = 0$, $\sigma = \sigma_M$, and by using the definition of β_i. Recognizing that the slope at $\alpha = 0$ must equal the slope of the tangent market portfolio, that is, the slope of the CML, $\left((\mu_M - r_f)\right)/\sigma_M$, produces Equation 29.3.

Specifying the market portfolio is a problem when one attempts to test the CAPM model. A broad value-weighted index of stocks, bonds, foreign investments, and proxies for such assets as real estate can be used. When one has such a proxy, one can estimate each stock's β from Equation 29.3 as β takes the form of the regression coefficient in a linear regression of the individual asset's excess return and the excess return of the index. Thus, if CAPM is a valid model, the scatter plot of each asset's excess return versus the market's excess return should cluster about a line through the origin. The line is the SML for that asset.

29.3 Modeling Returns over Time

In the single period portfolio construction framework of Section 29.2, Markowitz was able to obtain powerful results while using only second-order statistics to characterize asset returns. To optimize over multiple time periods with portfolio rebalancing allowed between each period, we must model how returns vary from one time period to the next.

The first step in establishing this model is to select either asset prices or returns as the basic unit of measure in order to describe the value of a tradable security over time. While this distinction may seem trivial due to the simple relationship between the two quantities, the subsequent impact on the choice of statistical models is significant. Whereas returns may be positive or negative, asset prices are constrained to be nonnegative, implying that a two-sided distribution (e.g., Gaussian) should not be used to model them. More importantly, simpler and more accurate models result from using returns rather than prices. It is for these reasons that the basic portfolio choice problem is formulated as a function of the underlying asset returns. Specifically, the log of the asset returns rather than the simple returns are used as shown below.

Let p_k denote the price of a single asset at time t_k, and let R_k denote the corresponding return over the period $(t_k{-}1, t_k]$, so that

$$R_k = \frac{p_k - p_{k-1}}{p_{k-1}} = \frac{p_k}{p_{k-1}} - 1.$$

The return represents the percent change in value of the asset. Here subscript k indicates that the value of the return becomes known at time t_k. This type of return is often referred to as a simple return. A second type of return, the log-return, r_k, is defined as the change in the log of the asset's price over the

* If there were a risky asset that was not a part of the tangent (market) portfolio, there would be no demand for it since every investor, regardless of his risk aversion, want only the tangent portfolio of risky assets. Supply would not meet demand; so no equilibrium would exist until the price of that asset dropped to the point where it was attractive enough to be part of the demanded tangent portfolio. A similar argument shows that the tangent portfolio must be the market portfolio.

length of the investment period as follows:

$$r_k = \log(1 + R_k) = \log(p_k) - \log(p_{k-1}).$$

Log-returns are also known as continuously compounded returns, since the quantity $\log(1 + R_k)$ represents the equivalent continuously compounded rate, r_k^c, corresponding to the simple rate, R_k. When R_k is sufficiently near zero, so that the Taylor series approximation given by

$$r_k = \log(1 + R_k) \approx R_k \tag{29.5}$$

is valid, the log-return is a good proxy for the simple return. Note that the monotonic relationship between the simple and log-returns implies that optimizing for one measure can be equivalent to optimizing for the other.

In a multiperiod setting, the total simple return across a set of N investment periods, denoted by R_T, is computed as the product of the per-stage simple returns.

$$1 + R_T = \prod_{k=1}^{N} (1 + R_k) = \prod_{k=1}^{N} \frac{p_k}{p_{k-1}} = \frac{p_N}{p_0}.$$

One advantage of using log-returns over simple returns is that the multiperiod log-return is equal to the sum of the per-stage returns, rather than their product, and is given by

$$r_T = \log(1 + R_T) = \log\left(\prod_{k=1}^{N} (1 + R_k)\right) = \sum_{k=1}^{N} \big(\log(p_k) - \log(p_{k-1})\big) = \sum_{k=1}^{N} r_k.$$

The additive accumulation of the log-return is beneficial in multiperiod portfolio selection problems, so that efficient computational techniques, such as dynamic programming, can be readily applied.

However, using log-returns do present a problem when dealing with portfolios of assets. While the portfolio simple return is computed as a linear combination of the simple returns of the constituent assets, the portfolio log-return does not have a similarly simple relationship to the underlying assets' log-returns. In order to circumvent this issue, the approximation of Equation 29.5 is used. The string of approximations for the log-return of a portfolio with the vector of portfolio weights at time k given by $\boldsymbol{v}_k$ is as follows:

$$\log\left[\prod_{k=1}^{N} (1 + \boldsymbol{v}_k \boldsymbol{R}_k)\right] = \sum_{k=1}^{N} \log(1 + \boldsymbol{v}_k \boldsymbol{R}_k) \approx \sum_{k=1}^{N} \boldsymbol{v}_k \boldsymbol{R}_k \approx \sum_{k=1}^{N} \boldsymbol{v}_k \boldsymbol{r}_k. \tag{29.6}$$

Thus, the log of the total portfolio return over time can be approximated by the sum over time of the weighted combination of log-returns on the individual assets. If the approximation is not valid over the timescale initially chosen for the problem, one can increase the rate at which the log-price process is sampled until the approximation is acceptable. Thus, in the discrete-time models in the sequel, the per-stage individual asset returns are given by the change in the log-prices, and the per-stage portfolio returns are approximated by a weighted sum of the assets' log-returns. Furthermore, the portfolio returns are assumed to add across stages in accordance with the properties of log-returns.

The fact that it is most natural to use log-prices and log-returns to model the evolution of asset values also creates problems in choosing the optimization criterion in discrete-time problems. One would prefer to perform a mean-variance optimization on the final total return achieved by an investment policy.* However, Equation 29.6 gives us a very usable form as an approximation for the *log* of the total return, not the total return itself. In the interest of creating a solvable optimization problem, we resign ourselves to minimizing the variance of this approximation to the total log-return, given a limit on the mean of

* Given an initial wealth, optimizing the final total return is equivalent to optimizing the final wealth.

this approximation to the total log-return. More intricate approximations to the true return and more sophisticated optimization criteria in discrete-time systems have been made to work by Campbell and Viceira [13,14]. The simple approximations we use in this development show the types of complications involved in solving discrete-time portfolio optimization problems.

When one moves to continuous-time models of assets and trading, many of the issues discussed above disappear and the results become aesthetically pleasing as we show in Section 29.5. Continuous-time models in finance were introduced in 1970 in the PhD thesis and related papers by Robert C. Merton, then a student of Paul Samuelson at MIT. In the Foreword to Merton's 1990 book [6], Samuelson writes, "The pole that propelled Merton to Byronic eminence was the mathematical tool of continuous probability a la Norbert Wiener and Kiyoshi Ito. Suddenly what had been complex approximations became beautiful simple truth."

The question of whether to model returns as evolving in discrete-time or continuous-time marks an important distinction in the finance theory literature. Continuous-time models carry with them the controversial assumption that trading takes place continuously in time. In addition, data on past returns are naturally captured in discrete-time format as are the estimation techniques used to characterize these data. Of course, the continuous trading assumption can be considered an approximation to how fast trades can actually occur; indeed, it is an approximation that is becoming better each day as computer-driven trading technology encourages trading on shorter time scales. In addition, the notion of continuously rebalancing a portfolio is a critical element of the breakthrough Black–Scholes–Merton option pricing theory as seen in [15,16].*

29.3.1 A Discrete-Time Model

Numerous models exist to describe the evolution of log-prices over time. The choice of model depends on the set of properties one is trying to describe, such as cross-asset and temporal return correlations, mean-reverting behavior, or common stochastic and growth trends. A model may describe the behavior of a single asset, or may jointly define the evolution of a set of dependent assets. As a general rule, the properties of a system can be best understood by choosing the simplest model that captures the set of desired behaviors. Simpler models lead to simpler solutions that can help build intuition; more complex models can come closer to mimicking real-life behavior.

One simple model assumes that the per-stage log-returns (per-stage differences in log-prices) are best represented as white noise, that is, they are independent and identically distributed. This widely used model captures the notion that future returns are unpredictable from past returns so that log-prices follow a random walk.† The log-returns may be modeled by a Gaussian distribution or a heavy-tailed distribution, such as a Pareto or Cauchy distribution. However, one must be careful in a multiperiod setting to understand the impact of the single-stage return model on the corresponding multiperiod return. For example, if the single-stage simple returns are assumed to be Gaussian, then the multistage returns are no longer Gaussian. On the other hand, if the single-stage log-returns are assumed to be Gaussian, the equivalent multistage log-returns, now formed as the sum of the per-stage log-returns, are also Gaussian. This, in turn, implies that the single-stage and multistage simple returns follow a shifted log-Gaussian distribution.‡ It is generally accepted that actual log-returns are more heavy tailed than Gaussian but, at this point, little can be done mathematically assuming heavy-tailed distributions.§ The tradeoff is made in modeling financial systems as it is usually made in all engineering systems to work

* Primbs [17] shows an alternate derivation of the Black–Scholes option pricing formula achieved via a quadratic optimal control problem.

† The book [18] introduces the theoretical concepts and has extensive discussions of how well data fit the random walk hypothesis and various other models of asset returns. At the opposite end of the spectrum, a best selling (and my favorite) nonmathematical treatment of investing is memorably entitled *A Random Walk Down Wall Street* [19].

‡ The returns are often called log-normal as $\log(1 + R)$ is Gaussian or normal.

§ The ramifications of heavy tails are the subject of much discussion. See [20].

with a model that may be less precise but that leads to insightful general results. Thus we generally assume log-Gaussian returns.

We consider a system where, starting at an initial time t_0, an investor is allowed to assemble a new portfolio at N points in time, equally spaced by an amount h. The time spacing parameter h is introduced because much of the theory of finance is developed using continuous-time models in the limit as h goes to zero while N grows to keep $Nh = T$, a finite time horizon. Many of the results become more elegant in the continuous-time setting. (E.g., the simple return per sample time goes to zero and the simple return and log-return per sample time converge, avoiding the approximation in portfolio returns that results from using log-returns.)

Let $\boldsymbol{x}[nh]$ be a p-dimensional random vector representing the log-prices of a group of assets that are assumed to evolve according to a first-order vector autoregressive (VAR) process as follows:

$$\boldsymbol{x}[nh + h] = (I - \Pi h)\boldsymbol{x}[nh] + \boldsymbol{\phi}h + \sqrt{h}\Sigma\boldsymbol{z}[nh], \tag{29.7}$$

where I is a $p \times p$ identity matrix, Π is a $p \times p$ matrix with small nonnegative eigenvalues, $\boldsymbol{\phi}$ is a p-dimensional constant vector, $\boldsymbol{z}$ is the p-dimensional zero mean vector, IID across time, with $E[\boldsymbol{z}[n]\boldsymbol{z}^{\mathrm{T}}[n]] = I$, and h is the scalar sampling time. Here we make use of the symmetric square root matrix Σ so that $\Sigma\Sigma = \Psi$, the covariance matrix of the actual noise; hence, we can allow $\boldsymbol{z}[nh]$ to be a *standard* IID noise process. Equation 29.7 describes a model where a deterministic signal plus white, Gaussian noise is passed through a linear system, a model that controls engineers know results in $\boldsymbol{x}[nh]$ being a Gaussian random vector with mean and covariance described using the state transition matrix and convolution sums.

Much of the work in finance theory is performed with $\Pi = 0$. The log-prices then follow a random walk with a drift determined by $\boldsymbol{\phi}$. Many important concepts have been developed using this model even though it assumes the complete unpredictability of future prices given past prices except for the average drift. There is a long history of important papers reporting econometric tests supporting this view [21]. However, most of these tests were made on scalar systems. Since 1990, evidence of predictability across both time and assets has been demonstrated [22]. Such cross-sectional and intertemporal predictability can be captured in the VAR model. This model captures the underlying economic principle that an increase in the valuation of company A at time n may be correlated with movement in the valuation of a company in A's supply chain or a competitor at time $n + 1$. This observation has led to many trading schemes, mostly *ad hoc*, that are referred to either as pairs trading when only two related assets are considered or, more generally, as cointegrated models where Π is not zero [23].

29.3.2 A Continuous-Time Model

A key aspect of Equation 29.7 is that the discrete-time white noise sequence $\boldsymbol{z}$ enters through the scale factor $\sqrt{h}$. This square root behavior is chosen to model two important properties of the log-prices as h goes to zero as we now explain.

Consider the process defined by*

$$\boldsymbol{Z}[nh + h] = \boldsymbol{Z}[nh] + \sqrt{h}\boldsymbol{z}[nh] \text{ with } E\left[\boldsymbol{z}[nh]\boldsymbol{z}[nh]^{\mathrm{T}}\right] = I.$$

As h goes to zero and nh goes to t, $\boldsymbol{Z}(t)$ can be described by a standard continuous-time vector stochastic differential equation. With the $\sqrt{h}$ behavior in the model, the sample paths of the continuous-time $\boldsymbol{Z}(t)$ process are continuous but nowhere differentiable, resulting in a model in which log-prices move smoothly but unpredictably even for very small time increments. With $\boldsymbol{Z}(0) = \boldsymbol{0}$ the process $\boldsymbol{Z}(t)$ is

* The lower-case $\boldsymbol{z}[nh]$ is the standard vector discrete-time white noise (IID) process while the upper-case $\boldsymbol{Z}(t)$ will be used for the standard vector continuous-time Wiener process that results from the limiting process on $\boldsymbol{Z}$ $[nh]$, the discrete-time summed white noise vector.

the standard vector Wiener process whose variance grows linearly with time. The process $\sqrt{h}\boldsymbol{z}[nh]$ has mean zero and covariance hI so that

$$\boldsymbol{Z}[nh] = \boldsymbol{Z}[0] + \sum_{k=0}^{n-1} \sqrt{h}\boldsymbol{z}[kh]; \quad E\left(\boldsymbol{Z}[nh]\right) = \boldsymbol{0}; \quad \text{cov}\left(\boldsymbol{Z}[nh]\right) = nhI.$$

As h goes to a small dt and nh goes to t, the notation $\boldsymbol{dZ}(t)$ is identified with $\boldsymbol{Z}[nh+h] - \boldsymbol{Z}[nh] = \sqrt{h}\boldsymbol{z}[nh]$.

With this understanding of $\boldsymbol{dZ}(t)$ we see that as h goes to a small dt and nh goes to t, Equation 29.7 becomes

$$\boldsymbol{dx}(t) = \Pi \boldsymbol{x}(t)\, dt + \boldsymbol{\phi}\, dt + \Sigma \boldsymbol{dZ}(t) \tag{29.8}$$

with $\Sigma\Sigma = \Psi$ and $\boldsymbol{Z}(t)$ a standard Wiener process.

Equation 29.8 is the continuous-time model of a deterministic constant signal plus white Gaussian noise being passed through a linear system, which results in $\boldsymbol{x}(t)$ being a Gaussian random vector with mean and covariance described using the state transition matrix and convolution integrals.

It is interesting to gain some insight into the quantity $(dZ)^2$. If ζ and z are scalars (for simplicity) with $z[nh]$ being a zero mean, unit variance IID sequence and

$$\zeta[nh+h] = \zeta[nh] + \left(\sqrt{h}z[nh]\right)^2; \quad \zeta[0] = 0,$$

then $\zeta[nh]$ has mean nh and variance $nh^2\left(E\left([z^4[nh]] - 1\right)\right)$. Assuming $E\left[z^4[nh]\right]$ is finite, as h goes to a small dt, $\text{Var}\left(\left(\zeta[nh+h] - \zeta[nh]\right)/h\right)$ goes to 0 and we switch to the continuous-time stochastic differential equation notation and say

$$d\zeta(t) = \left(d\boldsymbol{Z}(t)\right)^2 \text{ with mean } \left(d\zeta(t)\right) = dt \text{ and } \text{var}\left(d\zeta(t)\right) = 0,$$

so $(d\boldsymbol{Z})^2$ goes to the *deterministic* quantity dt.

It is this fact that necessitates the use of Ito calculus [24] when deriving the evolution of a nonlinear function of the output of a stochastic differential equation. Ito's lemma follows from using a Taylor series expansion of a nonlinear function with the above observation of the need to keep squared terms in $d\boldsymbol{Z}$ that behave as deterministic first-order terms in dt.

Ito's Lemma: Let $V(\boldsymbol{x}, t)$ be a twice differentially continuous function and let the p-dimensional vector $\boldsymbol{x}$ evolve according to the stochastic differential equation

$$\boldsymbol{dx} = \boldsymbol{f}(\boldsymbol{x}, t)\, dt + G(\boldsymbol{x}, t)\, \boldsymbol{dZ}(t).$$

Then V evolves according to the stochastic differential equation:

$$\begin{aligned} dV &= \frac{\partial V}{\partial \boldsymbol{x}}\, d\boldsymbol{x} + \frac{\partial V}{\partial t}\, dt + \frac{1}{2}\, d\boldsymbol{x}^{\mathrm{T}} \frac{\partial^2 V}{\partial \boldsymbol{x} \partial \boldsymbol{x}}\, d\boldsymbol{x} \\ &= \frac{\partial V}{\partial \boldsymbol{x}} \left(\boldsymbol{f}(\boldsymbol{x}, t)\, dt + G(\boldsymbol{x}, t)\boldsymbol{dZ}(t)\right) + \frac{\partial V}{\partial t} dt + \frac{1}{2} \boldsymbol{1}^{\mathrm{T}} G^{\mathrm{T}}(\boldsymbol{x}, t) \frac{\partial^2 V}{\partial \boldsymbol{x} \partial \boldsymbol{x}} G(\boldsymbol{x}, t) \boldsymbol{1}\, dt \end{aligned}$$

For example, let $V(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{x}$ and $\boldsymbol{x}$ evolve according to the stochastic differential equation (29.8), then, after applying Ito's lemma and dropping all terms of order $(dt)^y$ with $y > 1$,

$$dV = 2\boldsymbol{x}^{\mathrm{T}} \Pi \boldsymbol{x}\, dt + 2\, \boldsymbol{dx}^{\mathrm{T}} \boldsymbol{\phi}\, dt + 2\boldsymbol{x}^{\mathrm{T}} \Sigma\, \boldsymbol{dZ} + \boldsymbol{1}^{\mathrm{T}} \Psi \boldsymbol{1}\, dt$$

29.4 Discrete-Time Multistage Portfolio Optimization

29.4.1 Problem Formulation

We now consider the problem of constructing a portfolio of risky assets within a multistage setting. The investment time horizon is divided into N intervals, and the portfolio may be rebalanced at the start of each interval.* The log prices of the securities are assumed to be well-modeled by the discrete-time model of Equation 29.7, repeated here

$$\boldsymbol{x}[nh+h] = (I - \Pi h)\boldsymbol{x}[nh] + \boldsymbol{\phi} h + \sqrt{h}\Sigma \boldsymbol{z}[nh].$$

The vector of per-stage log-returns is equal to the change in the asset log-prices, $\boldsymbol{r}[n] = \boldsymbol{x}[n] - \boldsymbol{x}[n-1]$, and the per-stage portfolio log-return is approximated as $r_{\mathrm{p}}[n] \approx \boldsymbol{w}[n-1]^{\mathrm{T}}\boldsymbol{r}[n]$. The cumulative portfolio log-return at the end of N stages is approximated by

$$\sum_{k=0}^{N-1} r_{\mathrm{p}}[k] \approx \sum_{k=0}^{N-1} \boldsymbol{w}[k-1]^{\mathrm{T}}\boldsymbol{r}[k] = \sum_{k=0}^{N-1} \boldsymbol{w}[k-1]^{\mathrm{T}}\big(\boldsymbol{x}[k] - \boldsymbol{x}[k-1]\big).$$

The single-stage Markowitz mean-variance portfolio optimization framework introduced in Section 29.2.1 may be extended to a multistage setting by determining the sequence of portfolio weights, $\{\boldsymbol{w}[0], \ldots, \boldsymbol{w}[N-1]\}$, that solves the dual mean-variance problem. That is, we find the weights that minimize the variance of the cumulative portfolio log-return, subject to a constraint on the corresponding mean of the cumulative portfolio log-return. Formally, we seek the optimal policy, $\boldsymbol{w}^*$, so that

$$\left.\begin{aligned} \sigma^{2*}(\mu_0) = \min_{\boldsymbol{w}[0],\ldots,\boldsymbol{w}[N-1]} & \operatorname{var}\left[\sum_{n=0}^{N-1} \boldsymbol{w}^{\mathrm{T}}[n]\boldsymbol{r}[n+1]\right] \\ \text{such that } & E\left[\sum_{n=0}^{N-1} \boldsymbol{w}^{\mathrm{T}}[n]\boldsymbol{r}[n+1]\right] = \mu_0 \end{aligned}\right\} \quad P_{\mathrm{DT0}}.$$

This formulation does not immediately lend itself to an analytical solution using the standard techniques of stochastic optimal control, such as dynamic programming. The objective function is not to be additive across time due to the square of the final mean in the final covariance. However, it is just good to minimize the mean square of the final log-return since once the mean is set, this differs from the variance by a constant and the same optimal weights result. It is for this reason that we began with the dual of the mean-variance problem, P_{DT0}. Now, the investor can solve the following dynamic portfolio choice problem using a quadratic utility function where we have added an extra positive constant, λ_N, with the usual Lagrange multiplier, γ_N, for notational convenience in displaying the results. We also chose to maximize the negative of the natural utility function. Thus we solve the problem:

$$\max_{\boldsymbol{w}[0],\ldots,\boldsymbol{w}[N-1]} E\left[\gamma_N \sum_{n=0}^{N-1} \boldsymbol{w}^{\mathrm{T}}[n]\boldsymbol{r}[n+1] - \lambda_N \left(\sum_{n=0}^{N-1} \boldsymbol{w}^{\mathrm{T}}[n]\boldsymbol{r}[n+1]\right)^2\right] \quad P_{\mathrm{DT1}}.$$

The scale factors γ_N and λ_N define the shape of the utility function, and are deterministic quantities set by the investor in accordance with his or her risk preferences. The optimal sequence of portfolio weight vectors for Problem P_{DT0} is also optimal for Problem P_{DT1}, for some appropriate choice of γ_N and λ_N.

* The first use of dynamic programming to solve an optimal portfolio problem in discrete-time is generally credited to Paul Samuelson [3].

29.4.2 Optimal Policy Derivation

We begin by determining the solution to Problem P_{DT1} using the dynamic programming algorithm. The value function, that is, the optimal reward-to-go, at the last stage, denoted as $J_N(r_N)$, is given by

$$J_N(r_N) = U(r_N) = \gamma_N r_N - \lambda_N r_N^2.$$

According to the Bellman principle of optimality, at the beginning of the last stage, the investor acts to maximize the sum of his current and the expected reward-to-go. (The current reward is zero since only a function of the final return is rewarded.)

$$w_{N-1}^* = \arg\max_{w_{N-1}} E_{N-1}[0 + J_N(r_N)].$$

The value function at time $N-1$ is given by

$$\begin{aligned} J_{N-1}^* &= \max_{\boldsymbol{w}_{N-1}} E_{N-1}\left[J_N(r_N)\right] = \max_{\boldsymbol{w}_{N-1}} E_{N-1}[\gamma_N r_N - \lambda_N r_N^2] \\ &= \max_{\boldsymbol{w}_{N-1}} E_{N-1}\left[\gamma_N\left(r_{N-1} + \boldsymbol{w}_{N-1}^{\mathrm{T}}(\boldsymbol{x}_N - \boldsymbol{x}_{N-1})\right) - \lambda_N\left(r_{N-1} + \boldsymbol{w}_{N-1}^{\mathrm{T}}(\boldsymbol{x}_N - \boldsymbol{x}_{N-1})\right)^2\right] \\ &= \max_{\boldsymbol{w}_{N-1}} \gamma_N r_{N-1} + \gamma_N \boldsymbol{w}_{N-1}^{\mathrm{T}}\boldsymbol{m}_{N-1} - \lambda_N r_{N-1}^2 - 2\lambda_N r_{N-1}\boldsymbol{w}_{N-1}^{\mathrm{T}}\boldsymbol{m}_{N-1} - \lambda_N \boldsymbol{w}_{N-1}^{\mathrm{T}} S_{N-1}\boldsymbol{w}_{N-1}, \end{aligned}$$

where

$$\boldsymbol{m}_{N-1} = E_{N-1}[\boldsymbol{x}_N - \boldsymbol{x}_{N-1}] \quad S_{N-1} = E_{N-1}\left[(\boldsymbol{x}_N - \boldsymbol{x}_{N-1})(\boldsymbol{x}_N - \boldsymbol{x}_{N-1})^{\mathrm{T}}\right].$$

The optimal portfolio policy for the last stage, $\boldsymbol{w}_{N-1}$, is found to be

$$w_{N-1}^* = \frac{1}{2\lambda_N}(\gamma_N - 2\lambda_N r_{N-1})S_{N-1}^{-1}\boldsymbol{m}_{N-1}.$$

Using this expression for the optimal policy at time $N-1$, the value function can again be expressed as a quadratic function of the cumulative, realized return, as follows:

$$J_{N-1}(r_{N-1}) = \gamma_{N-1} r_{N-1} - \lambda_{N-1} r_{N-1}^2,$$

where

$$\gamma_{N-1} = \gamma_N(1 - \boldsymbol{m}_{N-1}^{\mathrm{T}} S_{N-1}^{-1}\boldsymbol{m}_{N-1}); \quad \lambda_{N-1} = \lambda_N(1 - \boldsymbol{m}_{N-1}^{\mathrm{T}} S_{N-1}^{-1}\boldsymbol{m}_{N-1});$$
$$c_{N-1} = \frac{0.25\gamma_N}{\lambda_N}\boldsymbol{m}_{N-1}^{\mathrm{T}} S_{N-1}^{-1}\boldsymbol{m}_{N-1}.$$

The new coefficients of the substage quadratic objective function, γ_{N-1} and λ_{N-1}, are themselves random variables. The above procedure can be repeated formally for each stage in order to find expressions for the full set of portfolio weight vectors, resulting in the following optimal control policy:

$$\boldsymbol{w}_k^* = \frac{1}{2\lambda_N}(\gamma_N - 2\lambda_N r_k)S_k^{-1}\boldsymbol{m}_k,$$

with

$$\eta_k = \nu_{k+1} - \boldsymbol{m}_{k+1}^{\mathrm{T}} S_{k+1}^{-1}\boldsymbol{m}_{k+1}; \quad \nu_k = E_k[\eta_k]; \quad \boldsymbol{m}_k = E_k[\eta_k \Delta\boldsymbol{x}_{k+1}]; \quad S_k = E_k[\eta_k \Delta\boldsymbol{x}_{k+1}\Delta\boldsymbol{x}_{k+1}^{\mathrm{T}}].$$

The above set of recursive equations for $\{\eta_k, \nu_k, \boldsymbol{m}_k, S_k\}$ does not admit a closed-form solution. The problem is that while the mean and variance of the log-prices $\boldsymbol{x}_k$ can be computed using Equation 29.7

for each time before starting the optimization, the conditional expectations of the more complicated quadratic forms that show up in $\{\eta_k, \nu_k, \boldsymbol{m}_k, S_k\}$ are more difficult. One can perform numerical integrations to compute these forms or one can find them by using forward Monte Carlo techniques. One technique, based on Monte Carlo methods and importance sampling, is given in [23]. This method utilizes a set of sample paths of the log-price process, $\boldsymbol{x}_n$, in order to approximate the required moments in a computationally efficient manner. The resulting solutions trace out the complete set of mean-variance solutions as one of the two parameters (λ_N, γ_N) is held fixed and the other varied. Various suboptimal closed-form approximate solutions to this problem are also given in [23].

29.4.3 Including a Budget Constraint and a Risk-Free Asset

Recall from Section 29.2.2 that a BC of the form $\boldsymbol{w}_k^{\mathrm{T}}\mathbf{1} = 1$ can be enforced by introducing the following linear transformation for the portfolio weight vector at each stage:

$$\boldsymbol{w}_k = \boldsymbol{c} + D\boldsymbol{v}_k = \begin{bmatrix} v_k^1 \\ \vdots \\ v_k^{p-1} \\ 1 - \sum_{i=1}^{p-1} v_k^i \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad D = \begin{bmatrix} I_{p-1} \\ -\mathbf{1}^{\mathrm{T}} \end{bmatrix}.$$

The derivation of the optimal control policy given in Section 29.4.2 can be repeated in order to solve for the new set of controls, $\{\boldsymbol{v}[0] \ldots \boldsymbol{v}[N-1]\}$.

The problem can also be modified to include a risk-free asset in a manner similar to the development in Section 29.2.3.

29.4.4 Formulation as a Linear Quadratic Regulator Problem

The linear quadratic regulator (LQR) problem is a paradigm problem result of optimal control theory. While the original dynamic mean-variance objective given in Problem P_{DT0} cannot be directly mapped into an LQR framework, the quadratic utility version defined by Problem P_{DT1} can. However, the resulting system is a special case of an LQR, in which the state evolution matrices are themselves stochastic. While formulation within the LQR framework provides a convenient and well-known representation for the problem of constructing optimal portfolios, it does not eliminate the accompanying computational difficulties.

In order to map Problem P_{DT1} into the LQR framework, let r_{Ck} be the cumulative portfolio gain at time k, $\boldsymbol{y}_k$ denote the system state and u_k the input.

$$\boldsymbol{y}_k = \begin{pmatrix} r_{Ck} \\ \boldsymbol{w}_k \end{pmatrix}; \quad \boldsymbol{u}_k = \boldsymbol{w}_{k+1} - \boldsymbol{w}_k,$$

The system evolves according to the following linear system:

$$\boldsymbol{y}_{k+1} = A_k\boldsymbol{y}_k + B_k\boldsymbol{u}_k + \boldsymbol{\varepsilon}_k; \quad A_k = \begin{pmatrix} 1 & \Delta\boldsymbol{x}_{k+1}^{\mathrm{T}} \\ \mathbf{0} & \mathrm{I} \end{pmatrix}; \quad \boldsymbol{B}_k = \begin{pmatrix} \Delta\boldsymbol{x}_{k+1}^{\mathrm{T}} \\ \mathrm{I} \end{pmatrix}.$$

The asset log-prices, $\boldsymbol{x}_k$, are not included as part of the state vector, but rather appear within the state transition matrices. It is valid to exclude the prices from the state vector because of the assumption that the control action does not influence the prices (a single trader does not move the market). Thus it is possible to express an otherwise nonlinear system using the LQR framework, at the expense of creating time-dependent, random system matrices.

The optimal set of actions is chosen to maximize the following quadratic cost function:

$$\underset{\varepsilon_0,\ldots,\varepsilon_{N-1}}{E}\left\{\mathbf{y}_N^{\mathrm{T}} Q_N \boldsymbol{y}_n + \sum_{k=0}^{N-1} (\boldsymbol{y}_k^{\mathrm{T}} Q_k \boldsymbol{y}_k + \boldsymbol{u}_k^{\mathrm{T}} R_k \boldsymbol{u}_k)\right\},$$

where the matrices Q_k and R_k are zero for all k except that the terminal cost, Q_N, is given by

$$Q_N = \begin{pmatrix} \lambda_N & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0} & 0 \end{pmatrix}.$$

Initializing the cumulative return to $r_0 = \gamma_N / 2\lambda_N$ produces the desired quadratic objective function as defined in Problem P_{DT1}.

Due to the presence of the log-prices in both A_k and B_k, these matrices are time-varying and stochastic, and therefore the standard LQR solution does not apply. As shown in [25], the optimal control law is still a linear function of the state, of the form

$$\boldsymbol{u}_k = -\left(R_k + E\left[B_k^{\mathrm{T}} K_{k+1} A_k\right]\right)^{-1} E\left[B_k^{\mathrm{T}} K_{k+1} A_k\right] \boldsymbol{y}_k = L_k \boldsymbol{y}_k \tag{29.9}$$

where $K_N = Q_N$ and

$$K_k = Q_k + E[A_k^{\mathrm{T}} K_{k+1} A_k] - E[A_k^{\mathrm{T}} K_{k+1} B_k]\left(R_k + E[B_k^{\mathrm{T}} K_{k+1} B_k]\right)^{-1} E[B_k^{\mathrm{T}} K_{k+1} A_k]. \tag{29.10}$$

The resulting set of moments needed here is identical to the set of moments given in Section 29.4.2 and are subject to the same computational difficulties. While the LQR framework provides a convenient and well-known representation for the problem of constructing dynamic mean-variance optimal (MVO) portfolios, such a formulation does not eliminate the accompanying computational difficulties as the moments needed for Equations 29.9 and 29.10 must still be solved using some numerical integration or Monte Carlo technique.

29.4.5 Example: Intertemporal Hedging

The benefit of the multistage mean-variance approach over the use of single stage mean-variance framework is demonstrated in the following example taken from [23]. In particular, the example demonstrates the benefit of intertemporal hedging that is possible in the dynamic setting. The solution exhibits a higher than expected variance of portfolio returns at each stage that is offset by negative correlations between the per-stage portfolio returns so that the final variance target is met.

Consider a system of two risky assets, in which the log-prices are assumed to evolve according to the process defined in Equation 29.7, with

$$I - \Pi = \begin{pmatrix} 0.7878 & 0.0707 \\ 0.2634 & 0.9122 \end{pmatrix}, \quad \Psi = \Sigma\Sigma = \begin{pmatrix} 0.0400 & 0 \\ 0 & 0.0049 \end{pmatrix},$$

and initial condition $\boldsymbol{x}_0 = (1.75 \quad 4.30)^{\mathrm{T}}$. For simplicity, we restrict our attention here to the case where the investor is given an investment horizon consisting of two stages. The investor must decide how much to allocate to each asset at the beginning of the first stage, and he is allowed to rebalance the portfolio at the beginning of the second stage. In order to determine the optimal portfolio weight vectors, the functions $\{\boldsymbol{m}_0, S_0\}$ and $\{\boldsymbol{m}_1, S_1\}$ are numerically approximated using the method described in [23], over a grid of $M = 5000$ sample paths.

First, suppose the investor sets a risk budget (i.e., standard deviation of the final portfolio return) of 20%, or $\sigma_0^2 = (0.20)^2 = 0.04$. After computing the optimal strategies, Monte Carlo runs were made to statistically characterize each solution. Table 29.1 compares the second-order statistics for the per-stage

TABLE 29.1 Second-Order Statistics Comparing Static versus Dynamic MVO Solutions for Two-Stage Example

	Stage 1, r_1					Stage 2, r_2						Total, r_T	
			Weights					Weights					
Strategy	Mean	Std	w_1	w_2	Net Lev	Mean	Std	w_1	w_2	Net Lev	Corr$[r_1, r_2]$	Mean	Std
Dynamic: no BC	0.31	0.26	−0.87	2.85	1.98	0.14	0.23	−0.15	1.47	1.32	−0.69	0.44	0.20
Dynamic: with BC	0.24	0.24	−0.99	1.99	1.00	0.14	0.23	−0.38	1.62	1.24	−0.60	0.38	0.20
Static: no BC	0.19	0.16	−0.47	1.86	0.39	0.13	0.20	−0.47	1.86	1.39	−0.39	0.32	0.20
Static: with BC	0.19	0.17	−0.64	1.64	1.00	0.13	0.21	−0.64	1.64	1.00	−0.45	0.31	0.20

portfolio returns using both dynamic and static MVO investment strategies, with and without BCs. In the static MVO strategy the investor is not allowed to rebalance his portfolio after the results are in from the first step of the problem. It is solved using the model to compute the statistics of the two-period horizon and then using the original Markowitz result of Section 29.2.2 to find the single optimal portfolio to be held constant over two periods. The quantities in Table 29.1 result from taking sample averages over Monte Carlo runs of each strategy.*

As Table 29.1 reveals, there is a direct relationship between the total expected return and the degree of negative correlation between the interstage portfolio returns. Higher negative correlation between r_1 and r_2 implies that the per-stage portfolio return variances may assume larger magnitudes while the total variance remains constant. The increased amount of per-stage risk is realized through the use of leverage, both at the individual asset and aggregate portfolio levels. This is perhaps best illustrated by comparing the return statistics for the static and dynamic strategies in the case where no BC is used. The case of no BC is the same as the case where there is a BC and there is also a risk-free asset with a risk-free rate of zero.

Whereas the static MVO solution uses a net leverage of 139% at each stage, the dynamic MVO solution first takes on a net leverage of 198%, followed by a second-stage leverage level of 132%. The higher first-stage portfolio return is a direct consequence of the increased leverage, whereas the dynamic and static schemes exhibit comparable returns for the second stage. The increased risk associated with the higher leverage in the first stage of the dynamic case is offset by the more negative correlation coefficient for the portfolio returns between the two stages of −0.69 for the dynamic strategy versus −0.39 in the static case. The total required standard deviation of 20% over the two-stage problem is met by both strategies. Note that nowhere in the problem setup were the mathematics instructed to pick first-stage weights that produce higher first-stage variance and more negative interstage correlations. This is simply an interpretation of what the dynamic programming algorithm does to produce the MVO returns.

Next, by allowing the risk budget to vary between 0% and 40%, we may generate the set of efficient frontiers in mean-standard deviation space as shown in Figure 29.2. The efficient frontiers for the cases without BCs are lines that intercept the y-axis at the origin as expected from the zero risk-free rate interpretation. In the static case, the line associated with a zero rate risk-free asset (i.e., no BC) is tangent to the efficient frontier when there is no risk-free asset as expected. However, in the dynamic case, the same relationship does not exist because, in the dynamic case, the amount of borrowing can change between periods. If we were to fix the amount of borrowing for the two periods, a tangent line would result. Clearly, allowing the amount of borrowing to optimally change can do no worse than fixing it and the line on the dynamic portfolio with no BC outperforms what the tangent line would achieve. Of course the dynamic strategy outperforms the static in both cases; hence the nesting of the efficient frontiers is what it should be.

* One cause of concern for the especially astute observer might be that the dynamic portfolio shows an average leverage of 1.24 for the second stage when a BC is active. This is legitimate since the dynamic portfolio has an average of 24% gain in the first stage and all wealth available at the second stage is invested.

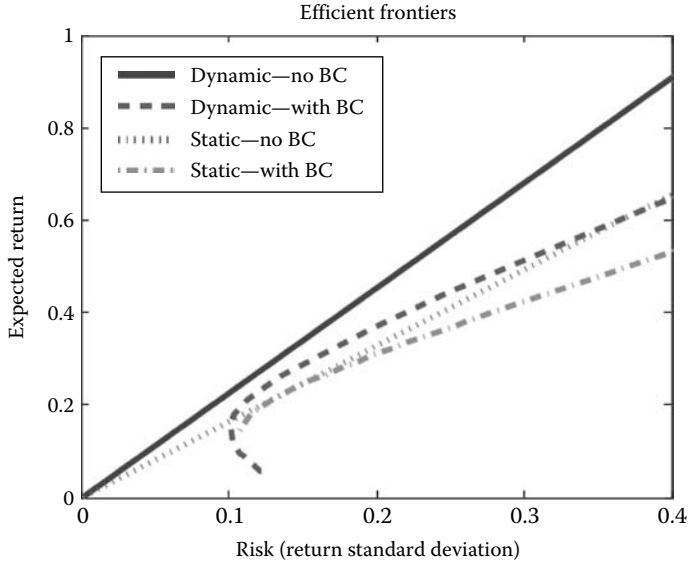

FIGURE 29.2 Efficient frontiers for static and dynamic MVO portfolios, with and without a budget constraint.

29.5 Continuous-Time Portfolio Optimization

The use of stochastic optimal control to solve portfolio problems in continuous-time was pioneered by Merton [4]. We start with the continuous-time model of Equation 29.8,* repeated here:

$$d\boldsymbol{x} = \Pi \boldsymbol{x}\, dt + \boldsymbol{\phi}\, dt + \Sigma\, d\boldsymbol{Z}.$$

Here $\boldsymbol{x}(t)$ is a vector of log-prices, $\boldsymbol{\phi}$ is a constant drift vector, and $\boldsymbol{Z}(t)$ is the standard vector Wiener process. In continuous-time, since all changes are infinitesimally small, we can work directly with the value of the portfolio (called W for Wealth) without the approximations that must be used to make the discrete-time analysis tractable.

Let $\boldsymbol{\nu}(t)$ be the vector of weights so that ν_i is the fraction of the wealth placed in the asset with log-price, x_i. Let there also be an asset called the risk-free bond that has log-price b with

$$db = r_{\mathrm{f}}\, dt,$$

so that the bond's price grows deterministically and exponentially with constant rate r_{f}.

The fraction of the wealth that is invested in the risk-free bond is $(1 - \boldsymbol{\nu}^{\mathrm{T}}\mathbf{1})$ where $\mathbf{1}$ is a vector of one's so that the weights of all investments including the investment in the risk-free bond sum to 1. This enforces the BC ensuring that all the wealth but no more is invested at each time. The ν_i are allowed to be negative to represent short sales of assets. When a bond is bought, money is lent so that if the fraction of wealth invested in the risk-free bond is negative, money is borrowed to finance the purchase of other assets.

Given how the log-prices of assets change, we can consider how the prices themselves change through the functional relationship $s_i = e^{x_i}$, where s_i is the price of security i. Since s_i is a nonlinear function of x_i, and x_i is governed by a stochastic differential equation, we must use Ito's Lemma to find the stochastic differential equation that governs s_i. The notation is a little tricky since the exponentiation takes place on a component-by-component basis.

* In this section, we drop the notational dependence on time when convenient.

Let $\mathbf{\Pi}_{\iota:}$ and $\mathbf{\Sigma}_{\iota:}$ be the ith rows of Π and Σ, respectively, and let Ψ_{ii} be the ith diagonal element of Ψ. Then

$$ds_i = s_i \left(\mathbf{\Pi}_{\iota:}\boldsymbol{x} + \boldsymbol{\phi_i} + \tfrac{1}{2}\Psi_{ii}\right) dt + s_i \mathbf{\Sigma}_{\iota:}\, \boldsymbol{dZ}. \tag{29.11}$$

Note that if $x_i(t+dt) - x_i(t) = \ln\big(s_i(t+dt)\big) - \ln\big(s_i(t)\big)$ had mean $m\,dt$ and variance $\sigma^2\,dt$, then the return $(s_i(t+dt) - s_i(t))/s_i(t)$ would have mean $(m + \frac{1}{2}\sigma^2)\,dt$ and variance $\sigma^2\,dt$. This is the same result that comes from taking the exponential of a Gaussian random variable. Here it comes not from a Gaussian assumption but from the correction explained in Ito's Lemma. The connection comes from the notion that even over very small time frames, $\mathbf{dZ} = \boldsymbol{Z}(t+dt) - \boldsymbol{Z}(t)$ is still the sum of a very large number of independent random variables and thus still acts like a Gaussian distribution.

Assumptions are made on the processes to rule out heavy-tailed distributions. [6, Chapter 3] Of course this is a somewhat problematic part of the financial engineering results, as it is generally agreed that asset returns empirically exhibit heavy tails. Yet, like all modeling efforts, some fidelity to reality is lost to achieve analytical results and their attendant insight and utility.

We can return Equation 29.11 to vector form by defining $\boldsymbol{s}$ as a vector of the s_i, S_D as a diagonal matrix with the s_i along the diagonal and Ψ_D as matching the diagonal elements of Ψ but zero elsewhere. Then

$$d\mathbf{s} = S_{\mathrm{D}} \left(\mathbf{\Pi}\boldsymbol{x} + \boldsymbol{\phi} + \tfrac{1}{2}\Psi_{\mathrm{D}}\mathbf{1}\right) dt + S_{\mathrm{D}}\Sigma\, \boldsymbol{dZ}$$

A vector with the amount of wealth invested in each security is given by $W\boldsymbol{v}$, where W is the amount of the investor's wealth, while $W\boldsymbol{v}^{\mathrm{T}}S_{\mathrm{D}}^{-1}$ is a vector with components equal to the number of shares of each security that is owned. This gives the following equation describing the evolution of wealth:

$$dW = W\left(\boldsymbol{v}^{\mathrm{T}}S_{\mathrm{D}}^{-1}\, d\mathbf{s} + (1 - \boldsymbol{v}^{\mathrm{T}}\mathbf{1})r_{\mathrm{f}}\right) dt$$

Substituting for $d\boldsymbol{s}$

$$\begin{aligned} dW &= W\left(\boldsymbol{v}^{\mathrm{T}}\Pi\boldsymbol{x} + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{\phi} + \tfrac{1}{2}\boldsymbol{v}^{\mathrm{T}}\Psi_{\mathrm{D}}\mathbf{1} + (1 - \boldsymbol{v}^{\mathrm{T}}\mathbf{1})r_{\mathrm{f}}\right) dt + \boldsymbol{v}^{\mathrm{T}}\Sigma\, \boldsymbol{dZ} \\ &= W\left(\boldsymbol{v}^{\mathrm{T}}\Pi\boldsymbol{x} + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{r}_{\mathrm{e}} + \tfrac{1}{2}\boldsymbol{v}^{\mathrm{T}}\Psi_{\mathrm{D}}\mathbf{1} + r_{\mathrm{f}}\right) dt + \boldsymbol{v}^{\mathrm{T}}\Sigma\, \boldsymbol{dZ} \end{aligned} \tag{29.12}$$

where $\boldsymbol{r}_{\mathrm{e}} = \boldsymbol{\phi} - \mathbf{1}r_{\mathrm{f}}$ is the vector of *mean excess log-returns*, that is, returns in excess of the risk-free rate.

Note that, unlike the discrete-time case, the time evolution of the wealth, that is, the value of the portfolio of assets, can be described in a useful way with no approximations. The cost of this simplification is the assumption that the composition of the portfolio is updated continuously.

The optimality criterion that is most commonly used in continuous-time finance problems is the power utility function. Here we apply it only at the final time T.

$$J(T) = E_0\left[\frac{1}{1-\gamma}W(T)^{1-\gamma}\right], \tag{29.13}$$

As long as $\gamma > 0$, this is a standard concave downward utility function that corresponds to a risk averse investor, that is, an investor who is willing to take a smaller average return on an asset provided it has a smaller variance in returns. The properties of variously shaped utility functions are much studied in financial economics [26] and the power utility function is also called the constant relative risk aversion (CRRA) curve. The larger the γ, the more risk averse is the investor. Each of two investors with the same γ is willing to bet the same percentage of her wealth on the same risky asset returns and the percentage of wealth invested in an asset is independent of the overall amount of wealth. This is the CRRA concept.

Solving the portfolio optimization for the power utility for all γ also traces out the optimal mean-variance portfolios. As γ decreases from very large value toward zero, the mean and variance of the optimal final wealth will both increase. As only the first two derivatives of J with respect to W at the optimal point affect the solution, each optimal point is also a solution to a quadratic cost function and

thus the solution to a mean-variance problem. Note that as γ goes through 1, the CRRA criterion becomes expected log-wealth. This is an important special case known as the growth-optimal portfolio. This case has special properties in the most studied case model in which $\Pi = 0$.

Equations 29.8, 29.12, and 29.13 now form a standard stochastic optimal control problem that is solved using the Hamilton–Jacoby–Bellman (HJB) equation.* The HJB equation provides the evolution of the optimal cost-to-go function, $V(t, W, \boldsymbol{x})$, defined as

$$V(t, W, \boldsymbol{x}) = \max_{\boldsymbol{\nu}(s), t \le s \le T} E_t \left[\frac{1}{1-\gamma} W(T)^{1-\gamma} \right].$$

In this case, the V function measures the conditional expected utility at time T given information up to time t and using the optimal weights between times t and T. The HJB equation is given by

$$\begin{aligned} 0 = \max_{\boldsymbol{\nu}} \{ & V_t + V_W W \left(\boldsymbol{\nu}^{\mathrm{T}} \Pi \boldsymbol{x} + \boldsymbol{\nu}^{\mathrm{T}} \boldsymbol{r}_{\mathrm{e}} + \frac{1}{2} \boldsymbol{\nu}^{\mathrm{T}} \Psi_{\mathrm{D}} \mathbf{1} + r_{\mathrm{f}} \right) + V_{\boldsymbol{x}} (\Pi \boldsymbol{x} + \boldsymbol{\phi}) \\ & + \frac{1}{2} V_{WW} W^2 \boldsymbol{\nu}^{\mathrm{T}} \Psi \boldsymbol{\nu} + W V_{W\boldsymbol{x}} \Psi \boldsymbol{\nu} + \frac{1}{2} \mathrm{Tr}(V_{\boldsymbol{xx}} \Psi). \end{aligned} \tag{29.14}$$

Performing the optimization over $\boldsymbol{\nu}$, we obtain an expression for the optimal weights:

$$\boldsymbol{\nu}^* = -(V_{WW} W \Psi)^{-1} \left[V_W \left(\Pi \boldsymbol{x} + \boldsymbol{r}_{\mathrm{e}} + \frac{1}{2} \Psi_{\mathrm{D}} \mathbf{1} \right) + V_{W\boldsymbol{x}} \Psi \right]. \tag{29.15}$$

If we now substitute the optimal weights back into the HJB equation we arrive at a deterministic partial differential equation involving $V(t, W, \boldsymbol{x})$ often called the Bellman equation:

$$\begin{aligned} 0 = & \left[V_t + V_W W r_{\mathrm{f}} + V_{\boldsymbol{x}} (\boldsymbol{\Pi} \boldsymbol{x} + \boldsymbol{\phi}) + \frac{1}{2} \mathrm{Tr}(V_{\boldsymbol{xx}} \Psi) \right] - \frac{1}{2 V_{WW}} \left[V_W \left(\Pi \boldsymbol{x} + \boldsymbol{r}_{\mathrm{e}} + \frac{1}{2} \Psi_{\mathrm{D}} \mathbf{1} \right) + V_{W\boldsymbol{x}} \Psi \right]^{\mathrm{T}} \Psi^{-1} \\ & \times \left[V_W \left(\Pi \boldsymbol{x} + \boldsymbol{r}_{\mathrm{e}} + \frac{1}{2} \Psi_{\mathrm{D}} \mathbf{1} \right) + V_{W\boldsymbol{x}} \Psi \right]. \end{aligned} \tag{29.16}$$

This equation can be approximately solved numerically by creating a multidimensional grid in t, W, and $\boldsymbol{x}$ and using finite difference approximations. More sophisticated techniques are also available.

The usual way of approaching an analytical solution to such an equation is to make an educated guess at the form of $V(t, W, \boldsymbol{x})$ and see where it takes you. A successful guess is called an ansatz. The ansatz for Equation 29.16 in this problem goes back to (at least) [27] and is given by

$$V(t, W, \boldsymbol{x}) = \frac{1}{1-\gamma} W(t)^{1-\gamma} \exp \left(\frac{1}{2} \boldsymbol{x}^{\mathrm{T}} A_0(t) \boldsymbol{x} + \boldsymbol{b}_0^{\mathrm{T}}(t) \boldsymbol{x} + c_0(t) \right).$$

After substituting this ansatz and its derivatives into the Bellman equation, we can manipulate the resulting equation so that we can divide out $V(t, W, \boldsymbol{x})$ and set the coefficient of each of the powers of $\boldsymbol{x}$ equal to 0. The three resulting equations are

$$\begin{aligned} -\dot{A}_0 &= A_0 \Pi + \Pi^{\mathrm{T}} A_0 + A_0 \Psi A_0 + \frac{1-\gamma}{\gamma} (\Psi A_0 + \Pi)^{\mathrm{T}} \Psi^{-1} (\Psi A_0 + \Pi), \\ -\dot{\boldsymbol{b}}_0 &= \Pi \boldsymbol{b}_0 + A_0 \boldsymbol{\phi} + A_0 \Psi \boldsymbol{b}_0 + \frac{1-\gamma}{\gamma} (\Psi A_0 + \Pi)^{\mathrm{T}} \Psi^{-1} \left(\Psi \boldsymbol{b}_0 + r_{\mathrm{e}} + \frac{1}{2} \Psi_{\mathrm{D}} \mathbf{1} \right), \\ \dot{c}_0 &= (1-\gamma) r_{\mathrm{f}} + \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{b}_0 + \frac{1}{2} \mathrm{Tr} \left(\boldsymbol{b}_0 \Psi \boldsymbol{b}_0^{\mathrm{T}} \right) + \frac{1}{2} \mathrm{Tr}(\Psi A_0) \\ &\quad + \frac{1-\gamma}{\gamma} \left(\Psi \boldsymbol{b}_0 + r_e + \frac{1}{2} \Psi_{\mathrm{D}} \mathbf{1} \right)^{\mathrm{T}} \Psi^{-1} \left(\Psi \boldsymbol{b}_0 + r_{\mathrm{e}} + \frac{1}{2} \Psi_{\mathrm{D}} \mathbf{1} \right), \end{aligned}$$

with the boundary conditions $A_0(T) = 0,\ \boldsymbol{b}_0 = \mathbf{0},\ c_0 = 0$.

* A primer in stochastic optimal control and the Hamilton–Bellman–Jacobi equation is contained in another chapter of this handbook. There is also an excellent treatment of these topics in the financial context in [6].

The equation for A_0 is a Riccati equation, with known solution techniques. The other two equations are simply differential equations as each solution is substituted into the next.

The equation for the optimal weights is now given by

$$\boldsymbol{v}^* = \frac{1}{\gamma}\Psi^{-1}\left[(\Psi A_0 + \Pi)\boldsymbol{x} + \Psi \boldsymbol{b}_0 + \boldsymbol{r}_e + \frac{1}{2}\Psi_D \mathbf{1}\right]. \tag{29.17}$$

It is interesting to examine the solution to the problem in the case $\Pi = 0$. This is the case most studied and can be referred to as the *random walk* model for returns. In the random walk model, the solution simplifies greatly. In particular, the equations for the ansatz parameters are solved with $A_0(t) = 0$ and $\boldsymbol{b}_0(t) = 0$ for all t. The optimal weights are then given by $\boldsymbol{v}^* = \frac{1}{\gamma}\Psi^{-1}\left[\boldsymbol{r}_e + \frac{1}{2}\Psi_D \mathbf{1}\right]$. ($c_0(t)$ is not identically zero but does not enter into the portfolio weights.) The optimal portfolio is now independent of $\boldsymbol{x}(t)$; so, despite the fact that the investor can rebalance her portfolio at any time, she keeps the portfolio the same for all time. The portfolio is also independent of the investment horizon T. Such an investment policy is called *myopic*. The investment weights are also those that result from solving a Markowitz mean-variance optimal problem, that is, the weights are proportional to the inverse variance matrix times the mean vector of the returns.

Equation 29.17 generalizes these results in two important ways when $\Pi \neq 0$. First, the optimal portfolio uses the information in the model that provides some predictability of future returns based on the current level of prices as captured in the $\Pi \boldsymbol{x}$ term. Second, how much this predictability is used depends on how close the investor is to the end of his/her investment horizon, T, as captured in the $A_0(t)$ and $\boldsymbol{b}_0(t)$ terms. These terms must converge to their boundary conditions of zero as the end of the investment horizon is approached. Including $\boldsymbol{x}(t)$ in the portfolio decision leads to intertemporal hedging similar to that demonstrated in the discrete-time example of Section 29.4.5. The role of $A_0(t)$ and $\boldsymbol{b}_0(t)$ is to modify the way in which intertemporal hedging is used depending on how close the investor is to his/her day of reckoning at time T.

The ability to solve the portfolio optimization problem with $\Pi \neq 0$ provides some powerful money-making opportunities. These processes are used to model mean-reverting assets and give additional quantitative tools to long-standing investment approaches such as *pair trading*. In pair trading, an investor balances long and short positions (positive and negative weights) in two assets that are affected similarly by the same fundamentals but whose prices move apart and then together due to the idiosyncratic movements of each. The result is movements that are correlated across time—when the prices get separated they are likely to return to their ordinary relationship. Of course, this is true if the future follows a model that is based on the past, a caveat that must be remembered in using any quantitative techniques. A look at pairs trading in the optimal portfolio setting is provided in [17]. With the solution above, one is no longer restricted to just dealing with simple pairs of stocks but can take full advantage of much more intricate multiasset correlations if one can find them.

29.6 Final Remarks

The examples of portfolio optimization shown here are direct applications of stochastic control theory. The problems explained are compelling and the type of modeling and analysis performed is typical of that performed by *quants*, mathematically sophisticated investment managers who use statistical methods to aid in achieving higher returns with lower risk. It is becoming more and more important to better understand the dynamics of financial markets and more engineers and researchers schooled in control theory should be able to aid in this endeavor.

References

1. H. Markowitz. Portfolio selection, *Journal of Finance*, 7(1), 77–91, 1952.
2. W.F. Sharpe, Capital asset prices: A theory of market equilibrium under conditions of risk, *Journal of Finance*, 19, 425–442, 1964.
3. P.A. Samuelson, Lifetime portfolio selection by stochastic dynamic programming, *Review of Economics and Statistics*, 51, 239–246, 1969. http://www.rle.mit.edu/dspg/documents/mbsPhDFinal_Feb09.pdf.
4. R.C. Merton, Lifetime portfolio selection under uncertainty: the continuous-time case, *Review of Economics and Statistics*, 51, 247–57, 1969 (Reprinted in [6]).
5. D.G. Luenberger, *Investment Science*, Oxford University Press, Oxford, 1997.
6. R.C. Merton, *Continuous-Time Finance*, Blackwell Publishing, New York, 1990.
7. J.H. Cochrane, *Asset Pricing*, Princeton University Press, Princeton, NJ, 2005.
8. D. Duffie, *Dynamic Asset Pricing Theory*, Princeton University Press, Princeton, NJ, 2001.
9. J. Lintner, The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets, *Review of Economics and Statistics*, 47(1), 13–37, 1965.
10. J. Mossin, Equilibrium in a capital asset market, *Econometrica*, 35, 368–383, 1966.
11. L. Bachelier, Théorie de la Spéculation, *Annales de l'Ecole Normale Supérieure*, Paris: Bauthier-Villars, 1900.
12. J. Tobin. Liquidity preference as behavior towards risk, *The Review of Economic Studies*, 25, 65–86, 1958.
13. J. Campbell and L. Viceira, Consumption and portfolio decisions when expected returns are time varying, *Quarterly Journal of Economics*, 114, 433–495, 1999.
14. J. Campbell and L. Viceira. *Strategic Asset Allocation: Portfolio Choice for Long-Term Investors*. Oxford University Press, Oxford, 2002.
15. F. Black and M. Scholes, The pricing of options and corporate liabilities. *Journal of Political Economy*, 81 (May–June), 637–54, 1973.
16. R.C. Merton, Theory of rational option pricing, *Bell Journal of Economics and Management Science*, 4 (Spring), 141–83. 1973.
17. J.A. Primbs, A control system based look at financial engineering. 2009. http://www.stanford.edu/~japrimbs/Publications/FT%20draft%201%2020080113.pdf.
18. J.Y. Campbell, A.W. Lo, and A.C. MacKinlay, *The Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ, 1997.
19. B.G. Malkiel, *A Random Walk Down Wall Street*, W.W. Norton & Co., New York, 1973 and 1997.
20. B. Mandelbrot and R.L. Hudson, *The (Mis)Behavior of Markets: A Fractal View of Risk, Ruin, and Reward*, Basic Books, New York, 2004.
21. E. Fama, Efficient capital markets: A review of theory and empirical work, *Journal of Finance*, 25, 383–417, 1970.
22. A.W. Lo and A.C. MacKinlay, When are contrarian profits due to stock market overreaction? *The Review of Financial Studies*, 3(2), 175–205, 1990.
23. M.B. Rudoy, Multistage mean-variance portfolio selection in cointegrated vector autoregressive systems, PhD Dissertation, MIT, 2009.
24. K. Ito, On stochastic differential equations, *Memoirs of the American Mathematical Society*, 4, 1–51, 1951.
25. D. Bertsakis, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2000.
26. C. Huang and R.H. Litzenberger, *Foundations for Financial Economics*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
27. F. Herzog, G. Dondi, H.P. Geering, and L. Schumann, Continuous-time multivariate strategic asset allocation, *Proceedings of the 11th Annual Meeting of the German Finance Association,* Session 2B, pp. 1–34, Tübingen, Germany, October 2004.

30

Earthquake Response Control for Civil Structures

Jeff T. Scruggs
Duke University

Henri P. Gavin
Duke University

30.1 Introduction

In contemporary civil engineering, performance analysis for the dynamic response of structures due to transient environmental loads constitutes a standard part of the design process. Acceptable structural designs must be resilient in the face of earthquake and wind disturbances, as well as blast loads. Considerable uncertainty exists regarding these loads, in terms of their arrival times, as well as their frequency content, intensity, and time duration. Beginning in the 1960s, structural engineers began adopting vibration isolation concepts and technologies, which had first emerged primarily in the aerospace sector, to design supplemental passive mechanical components of structures for the suppression of dynamic response. These included several basic concepts which have since been embraced and implemented in many structures around the world. Most prevalent among these is the installation of highly flexible base isolators in low- and medium-rise buildings. Such systems have been shown to be highly effective at mitigating large accelerations and deformations in the superstructure during earthquakes, and their implementation is becoming standardized in the industry. In high-rise buildings, supplemental tuned mass dampers (usually installed at or near the top floor) have gained acceptance as a technique for reducing structural accelerations due to wind loads. Figure 30.1 illustrates an idealized building structure

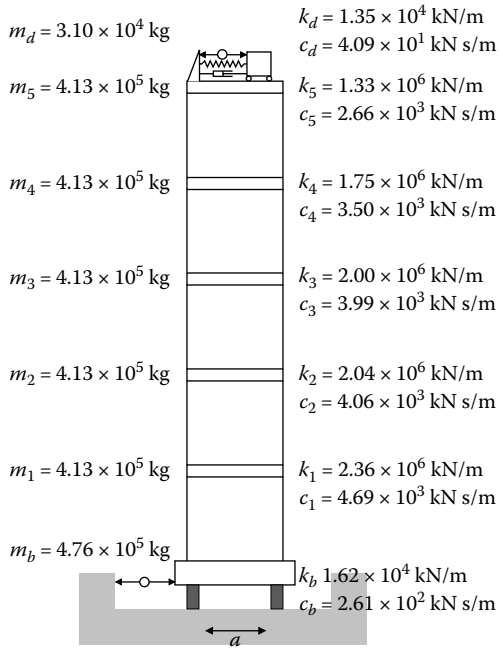

FIGURE 30.1 Illustration of a building with base isolation and a tuned mass damper. The symbols ←○→ indicate energy absorption devices. Numerical values correspond to those used in the example in Section 30.6.

incorporating both a base-isolation system and a tuned mass damper. Additionally, many tall buildings around the world have been fitted with ancillary dissipative damping mechanisms, such as hydraulic or friction dampers installed between stories to dissipate energy.

A fundamental design challenge associated with these technologies concerns the fact that no system can simultaneously make all structural deformations, as well as all absolute accelerations, arbitrarily small. For example, if a structure is designed to be extremely rigid, then dynamic excitation due to an earthquake will transmit extremely high accelerations into the superstructure which can damage building contents and can pose a serious safety risk. Similarly, an isolation system which is extremely effective at shielding a superstructure from an earthquake disturbance may undergo isolator strains high enough to result in damage, or simply to be impractical. It is therefore nontrivial to design structures which strike a good balance between these two extremes.

As early as the mid-1970s, researchers in structural engineering began to consider the idea of using feedback control to obtain better dynamic response tradeoffs than those possible with purely passive systems. During this time, and throughout the 1980s, research in this area focused primarily on the implementation of actively controlled structures, using externally-powered hydraulic or electromechanical actuation, and implementing much of the control theory, which had by that time become standard (such as $\mathcal{H}_2$/Linear Quadratic Gaussian (LQG) techniques) [1]. In 1989 the first actively controlled structure was built in Tokyo by the Kajima Corporation, to suppress wind-induced vibration. However, since that time, only a handful of actively controlled structures have been implemented in practice.

There are a number of reasons for this. Not the least of these is the large investment associated with such systems, and the lingering question as to whether this investment is indeed offset by the reduction in risk. Moreover, the actuators employed by such control systems must have access to significant external power, leading to a precarious interdependency between the reliability of the controlled structure, and that of the surrounding power grid. This makes them decidedly unreliable during earthquakes, and other extreme events likely to cause power outages. The electrical power and energy required to realize the

forces commanded by active control systems can also be quite high (on the order of 0.1 GW), further exacerbating these reliability concerns, and calling into question the practicality of the concept. Beyond these issues, there are also concerns related to stability robustness. Civil structures exhibit significant model uncertainty, and the uncertainty in the disturbance is even greater. As such, there persists a nagging concern that in the presence of all this uncertainty, active control might destabilize an otherwise stable structure.

In the early 1990s, all these issues led to a rally around the use of control devices which are often called "semiactive." Essentially, semiactive devices are controllably dissipative devices, which are incapable of increasing the vibratory energy in a structure. Their controllable passive parameters can, however, be adjusted with high-bandwidth, in real time, in response to feedback via low-power electric control signals. By controlling the dissipative parameters in these devices through feedback, the controlled system can exceed the performance of uncontrolled passive systems and can approach the performance of fully active systems, while being more reliable and requiring much less power to operate.

Semiactive systems have a long history in the vehicle suspensions area for the reduction of cabin acceleration, with investigations beginning in the 1970s [2]. One of the simplest examples of a semiactive device is the variable-orifice damper illustrated in Figure 30.2, for which an adjustable orifice allows the effective viscosity of the device to be adjusted. Other examples include dampers with electrorheological and magneto-rheological (MR) fluids, devices with controllable Coulomb friction, devices with variable stiffness, and electromechanical transducers with controllable electrical shunts. Indeed, the array of novel semiactive devices which have been proposed over the last two decades is vast. It is beyond the scope of this chapter to provide details on all of them, especially as a number of excellent surveys already exist [3–5]. Research has largely focused on device design. Meanwhile, the development of a consistent theory for the control of semiactive systems has developed more gradually.

Our primary focus in this chapter is to provide an overview of the control-theoretic aspects of semiactive systems. To narrow the focus further, we specifically concentrate on the earthquake response suppression problem. It is not the intent of this chapter to provide a survey of the semiactive control literature; rather it is to isolate an interesting class of control problems, motivated by this technology, and to discuss an approach to their solution.

30.1.1 The Idealized Bilinear System Model

Consider an arbitrary vibratory mechanical system, in which n_f semiactive devices are embedded. We presume the structural system to be approximated by the linear state space $x(t) \in \mathbb{R}^{n_x}$ governed by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_f\mathbf{f} + \mathbf{B}_a\mathbf{a} \tag{30.1a}$$

$$\mathbf{v} = \mathbf{B}_f^T\mathbf{x} \tag{30.1b}$$

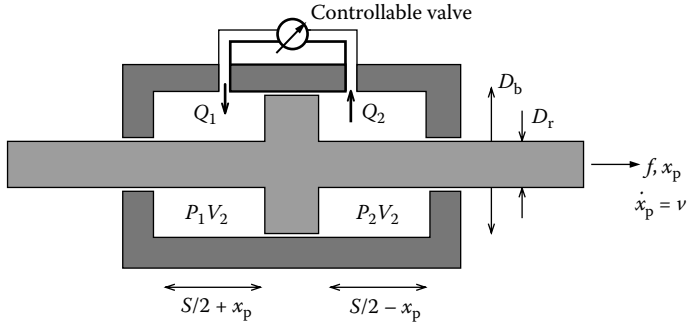

FIGURE 30.2 Schematic representation of a controllable hydraulic damper.

$$\mathbf{y} = \mathbf{C}_y\mathbf{x} + \mathbf{D}_{ya}\mathbf{a} \tag{30.1c}$$

$$\mathbf{z} = \mathbf{C}_z\mathbf{x} + \mathbf{D}_{zf}\mathbf{f} + \mathbf{D}_{za}\mathbf{a} \tag{30.1d}$$

in which $\mathbf{v}(t) \in \mathbb{R}^{n_f}$ is the vector of device velocities, $\mathbf{f}(t) \in \mathbb{R}^{n_f}$ the collocated vector of device forces, $\mathbf{a} \in \mathbb{R}^{n_a}$ the vector of ground accelerations, $\mathbf{y} \in \mathbb{R}^{n_y}$ the feedback measurement vector, and $\mathbf{z} \in \mathbb{R}^{n_z}$ the performance vector. In structural engineering applications, the following model properties can be assumed to hold generally:

1. $\mathbf{A}$ is Hurwitz, because civil engineering structures are always open-loop stable.
2. The transfer function from $\mathbf{f}$ to $\mathbf{v}$ is positive real, and strictly proper. As such, by the Kalman–Yakubovic–Popov lemma, we assume a self-dual, passive state space realization; that is, one in which $\mathbf{A} + \mathbf{A}^T \leqslant 0$, and such that $\mathbf{B}_f$ participates in both Equations 30.1a and 30.1b as shown.
3. We can further assume that $(\mathbf{A}, \mathbf{A} + \mathbf{A}^T)$ is an observable pair, and that consequently, by Lasalle's theorem, $V(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$ is a Lyapunov function for the open-loop system.

Ideal semiactive devices are constrained by the requirement that at every time instant, they must absorb structural energy. The simplest model of a semiactively constrained system is the bilinear, variable-viscosity controller

$$\mathbf{f}(t) = -\mathbf{U}(t)\mathbf{v}(t) \tag{30.2}$$

in which the matrix variable $\mathbf{U}(t)$ can be controlled, subject to algebraic constraints characterized by $\mathbf{U}(t) \in \mathcal{U}, \forall t$. Technically, the only requirement $\mathcal{U}$ must satisfy to ensure instantaneous dissipativity is that

$$\mathbf{v}^T(t)\mathbf{f}(t) = -\mathbf{v}^T(t)\mathbf{U}(t)\mathbf{v}(t) \leqslant 0, \ \forall \mathbf{U}(t) \in \mathcal{U}, \mathbf{v}(t) \in \mathbb{R}^{n_f} \tag{30.3}$$

that is,

$$\mathcal{U} = \left\{ \mathbf{U} \in \mathbb{R}^{n_f \times n_f} : \mathbf{U} + \mathbf{U}^T \geqslant 0 \right\} \tag{30.4}$$

However, the hardware realization of a semiactive system often imposes a much more restrictive domain. For example, it is generally impractical for semiactive fluid dampers to be able to transmit hydraulic power between each other, and this effectively imposes a decentralization constraint on $\mathcal{U}$; that is,

$$\mathcal{U} = \left\{ \text{diag}\{\ldots, u_i, \ldots\} \in \mathbb{R}^{n_f \times n_f} : u_i \geqslant 0 \right\} \tag{30.5}$$

Furthermore, many semiactive systems (such as controllable-orifice dampers or controllable electromechanical shunts) have a maximum viscosity u_e for each device, which restricts each u_i to $[0, u_{ei}]$. (Note that semiactive devices have a minimum viscosity as well, but this can be absorbed into $\mathbf{A}$.) Finally, we note that if we normalize $\mathbf{f} \leftarrow \mathbf{U}_e^{-1/2}\mathbf{f}$ and $\mathbf{v} \leftarrow \mathbf{U}_e^{1/2}\mathbf{v}$, where $\mathbf{U}_e = \text{diag}\{\ldots, u_{ei}, \ldots\}$, then this normalized system model retains the self-duality of state space $\mathbf{x}$, while nondimensionalizing $\mathcal{U}$ as

$$\mathcal{U} = \left\{ \text{diag}\{\ldots, u_i, \ldots\} \in \mathbb{R}^{n_f \times n_f} : u_i \in [0, 1] \right\} \tag{30.6}$$

It is this final characterization of $\mathcal{U}$ which is usually considered in semiactive control theory. However, all controllers discussed in this chapter extend easily to definitions of $\mathcal{U}$ as arbitrary subsets of (30.4), so long as they are convex and bounded.

It will be convenient to also characterize the feasibility region for $\mathbf{f}$, given $\mathbf{v}$, as

$$\mathcal{F}(\mathbf{v}) = \left\{ \mathbf{f} \in \mathbb{R}^{n_f} \ : \ \mathbf{f} = -\mathbf{U}\mathbf{v}, \ \mathbf{U} \in \mathcal{U} \right\} \tag{30.7}$$

For the particular case of decentralized semiactive constraints as in Equation 30.6, this set is characterized by

$$\mathcal{F}(\mathbf{v}) = \left\{ \mathbf{f} \in \mathbb{R}^{n_f} \ : \ f_i^2 + f_i v_i \leqslant 0, \ i \in \{1 \ldots n_f\} \right\} \tag{30.8}$$

We note in general, that if $\mathcal{U}$ is convex, so then is $\mathcal{F}(\mathbf{v})$, for any $\mathbf{v} \in \mathbb{R}^{n_f}$.

In reality, no semiactive device has constraints which are exactly characterized by the bilinear constraints described above. For example, almost all devices have force saturation thresholds. Moreover, many devices exhibit dissipative behavior more reminiscent of controllable Coulomb friction, rather than controllable viscosity. Devices also generally exhibit hysteresis and bandwidth limitations. Later in the chapter, we will discuss some of the ways these issues complicate the theory.

30.1.2 The Semiactive Control Problem

For the idealized system description described above, the objective of semiactive control design is to synthesize a feedback law $\phi : \mathbf{y} \to \mathbf{f} \in \mathcal{F}(\mathbf{v})$, which either minimizes or guarantees a bound on some measure of $\mathbf{z}$ for the closed loop. We note that, embedded in this problem statement, is the implication that $\mathbf{v}$ is a subspace of $\mathbf{y}$, because precise knowledge of $\mathbf{v}$ is necessary to impose $\mathbf{f} \in \mathcal{F}(\mathbf{v})$. Without getting into the specific performance measures that might be appropriate for these applications (this will be done in Section 30.3), we state a few performance-independent observations regarding this problem, which make it interesting from a control-theoretic point of view.

1. All feasible controllers are asymptotically stable. This follows from a simple Lyapunov argument. We know that $\mathbf{A} + \mathbf{A}^T \leqslant 0$, and thus $\mathbf{x}^T\mathbf{x}$ is a Lyapunov function for the open-loop system. In closed-loop, we have for the free response,

$$\frac{d}{dt}\mathbf{x}^T(t)\mathbf{x}(t) = \mathbf{x}^T(t)\left[\mathbf{A} + \mathbf{A}^T - \mathbf{B}_f\left(\mathbf{U}(t) + \mathbf{U}^T(t)\right)\mathbf{B}_f^T\right]\mathbf{x}(t) \tag{30.9}$$

$$\leqslant \mathbf{x}^T(t)\left[\mathbf{A} + \mathbf{A}^T\right]\mathbf{x}(t),\ \forall \mathbf{U}(t) \in \mathcal{U} \tag{30.10}$$

   Note that this is still true even if the structural model is uncertain, so long as the actual structural system is positive real (which civil structures always are). As such, stability robustness is ensured by the physics of the problem, and is thus irrelevant as a design issue. This is, in fact, one of the primary selling points of semiactive systems.
2. The only feasible, autonomous control law resulting in a linear closed-loop system is static velocity feedback; that is, $\mathbf{U}(t) = \mathbf{U}_0, \forall t$. However, for $\mathcal{U}$ as defined by Equation 30.6, this feedback law can be realized with linear, time-invariant viscous dampers. Therefore, it may be said in general that in order for semiactive devices to be worth implementing, the feedback controller *must be nonlinear*.
3. In the absence of stability robustness issues, the only differentiating factor to discriminate between controllers is their performance. However, in light of note 2 above, this performance must be assessed in the context of nonlinear feedback. Because the nonlinearity is bilinear in $\mathbf{U}$ and $\mathbf{x}$, it does not lend itself to linearization approximations, and also cannot be absorbed into an uncertainty, as is often done in robust control. As such, it is nontrivial to design semiactive controllers which adhere to analytically computable performance measures, even if the system model is known precisely. Performance-bounded control for uncertain semiactive systems, as well as for systems in which the state space (Equation 30.1a) is nonlinear, are even more challenging but are more representative of actual structural systems. Almost all problems in these areas remain open.

30.1.3 Nomenclature

Before moving on, we establish some notational conventions. $\|\mathbf{q}\|_2$ and $\|\mathbf{q}\|_\infty$ are the Euclidean and infinity norms for a vector $\mathbf{q} \in \mathbb{R}^{n_q}$. Unless otherwise stated, all time-valued vector functions $\mathbf{q}(t)$ are presumed to have support on $t \in [0, \infty)$. As such, $\|\mathbf{q}\|_{\mathcal{L}_2}$ and $\|\mathbf{q}\|_{\mathcal{L}_\infty}$ are the Lebesgue norms for $\mathbf{q}(t)$ on $\mathcal{L}_2$ and $\mathcal{L}_\infty$, respectively; that is $\|\mathbf{q}\|_{\mathcal{L}_2}^2 = \int_0^\infty \mathbf{q}^T(t)\mathbf{q}(t)dt$ and $\|\mathbf{q}\|_{\mathcal{L}_\infty}^2 = \sup_{t\in[0,\infty)} \mathbf{q}^T(t)\mathbf{q}(t)$. The weighted Euclidean norm is $\|\mathbf{q}\|_{\mathbf{R}}^2 = \mathbf{q}^T\mathbf{R}\mathbf{q}$. For a matrix $\mathbf{R}$, $\lambda_{max}\{\mathbf{R}\}$ and $\lambda_{min}\{\mathbf{R}\}$ are the maximum and minimum eigenvalues. Similarly, $\sigma_{max}\{\mathbf{R}\}$ and $\sigma_{min}\{\mathbf{R}\}$ are the maximum and minimum singular values.

For a time-valued vector $\mathbf{q}(t)$, $\hat{\mathbf{q}}(s)$ is the Laplace transform. For a stationary stochastic process $\mathbf{q}(t)$, $\mathcal{E}\,\mathbf{q}$ is the expectation. The functions sat(·), sgn(·), and hvs(·) are the saturation, sign, and Heaviside step functions, respectively.

30.2 Earthquake Disturbance Models

In this section, we provide some background on the customary way in which earthquakes are modeled and simulated in structural engineering, and connect these practices with a stochastic disturbance characterization amenable to the design and evaluation of controllers.

30.2.1 Response Spectra in Structural Design

For purposes of structural design, earthquake ground motions are characterized by their effects on structures. Ground motion effects are quantified by spectra of the peak responses they induce for a set of simple linear oscillators of differing natural frequencies and a specified damping level. The displacement response $r(t)$ of a simple oscillator with natural period $T_\mathrm{n} = 2\pi/\omega_\mathrm{n}$ and damping ratio ζ, to an earthquake ground acceleration $a(t)$, is modeled by

$$\ddot{r}(t) + 2\zeta\omega_\mathrm{n}\dot{r}(t) + \omega_\mathrm{n}^2 r(t) = -a(t)\,. \tag{30.11}$$

The acceleration response spectrum, S_a, is a plot of the peak displacement response as a function of T_n for a constant ζ (e.g., 0.05), and is expressed in units of acceleration by multiplying by ω_n^2; that is,

$$S_a(T_\mathrm{n}, \zeta) = \max_t |r(t; T_\mathrm{n}, \zeta)| \frac{4\pi^2}{T_\mathrm{n}^2}. \tag{30.12}$$

Such spectra are central to earthquake-resistant structural design procedures; equivalent static design forces are conveniently obtained by multiplying the the mass of a structure by the value of S_a corresponding to the structure's fundamental natural period and damping, and for a design-level ground shaking. The particular acceleration record(s) used for design purposes depends on many factors such as the proximity of the building site to faults, the characteristic magnitude of earthquakes at those faults, and the characteristics of the soil at the building site.

Ground motions resulting in large spectral accelerations occur with less frequency than those resulting in smaller responses, and earthquake response spectra are categorized by the frequency with which spectral accelerations may be exceeded. Return periods (the inverse of the exceedance frequency) used in structural design are typically on the order of 500 years to 2500 years and depend, in part, upon the intended use of the structure.

The frequency content of earthquake ground motions is predominantly within a frequency band of 1–5 Hz. The mechanical impedance of structures may be detuned from this frequency range through the use of compliant elements in the structural foundation. Such base isolation systems comprise components that are flexible in the horizontal direction, with displacement capacities on the order of 20–50 cm, and components that add damping to the foundation. The fundamental resonant frequency of an isolated structure is typically designed to be in the range of 0.3–0.6 Hz. Base isolation systems have been shown to be effective in protecting structures from low to moderate levels of earthquake ground motions. However, ground motions within a 10 km range of an earthquake fault can sometimes exhibit a pulse-like characteristic with significant coherence and with frequency content predominantly in the range of the fundamental natural frequency of the base-isolated structure. This type of ground motion can place particularly high demands on base-isolated structures, especially with regard to the displacement capacity of the isolators. Near-fault ground motions with pulse-like characteristics therefore merit special consideration in the design and analysis of base-isolated structures.

30.2.2 Earthquake Models for Control Design and Evaluation

For purposes of control system design, earthquake ground accelerations in each direction k can be modeled as an independent, enveloped, filtered white-noise process; that is, $a_k(t) = e(t)p_k(t)$, where $e(t)$ is the envelope (usually the same for all directions), and $p_k(t)$ is a stationary stochastic process with spectral density $\Phi_{pk}(\omega) = |G_w(j\omega)|^2$, for some minimum-phase filter G_w. The envelope of the ground acceleration record [6] may be expressed by

$$e(t) = (\alpha\beta)^{-\alpha}\, t^{\alpha}\, e^{(\alpha - t/\beta)} \tag{30.13}$$

in which the parameter β describes the decay time constant and the product $\alpha\beta$ describes the rise time of the envelope. The filter G_w may be modeled as the second-order system

$$G_w \sim \left[\begin{array}{cc|c} 0 & 1 & 0 \\ -(2\pi f_g)^2 & -4\pi\zeta_g f_g & \bar{a} \\ \hline 0 & 4\pi\zeta_g f_g & 0 \end{array}\right], \tag{30.14}$$

where f_g is a ground motion frequency parameter, ζ_g is a ground motion damping parameter, and $\bar{a}$ is a scaling parameter. A record is synthesized in four steps by: (1) filtering a sequence of uncorrelated Gaussian samples through the system of Equation 30.14; (2) multiplying the filtered Gaussian sample by the envelope of Equation 30.13; (3) detrending the resulting filtered white noise sample such that $a(0) = 0$ and $\sum a(t_i) = 0$; and (4) scaling the record such that the peak ground velocity (PGV),

$$\text{PGV} = \max_t |v_g(t)| = \max_j \left| \sum_{i=1}^{j} a(t_i)\, \Delta t \right|, \tag{30.15}$$

equals a specified value. The specified PGV value determines the intensity of the ground motion; Because each record is scaled to the same PGV value, the amplitude constant $\bar{a}$ in the filter in Equation 30.14 is arbitrary.

To connect the disturbance models used in the areas of control and structural design, the properties of $e(t)$ and $G_w(j\omega)$ should result in ground motions having response spectra $S_a(T_n, \zeta)$ representative of the seismic hazard at the proposed building site. To this end, sets of prerecorded earthquake ground motions, representative of design-level ground shaking for a particular region, may be used as a basis for identifying envelope functions and filter models. The set of prerecorded records are first normalized so that the PGV of each record in the set matches the median PGV for the set. The model fitting process is then carried out in two steps. In the first step, envelope parameters (α, β) for the model are identified as the mean of the envelope parameters identified from each record in the data set. In the second step, filter parameters $(f_g, \zeta_g, \bar{a})$ are adjusted so that the mean spectral accelerations from the enveloped filtered noise matches the mean spectral accelerations of the set of prerecorded earthquake records.

Two sets of historical earthquake records representative of magnitude 7 earthquakes in California (measured at sites approximately 4 and 15 km, respectively, from an epicenter) have recently been assembled for a FEMA study on the response of buildings to strong earthquake ground motions [7]. The set with more distant epicenters, the so-called "far-field" set, consists of 44 records from earthquakes with magnitudes from 6.5 to 7.6 measured at sites located 11–21 km from the epicenter and with relatively firm local soil conditions. The set with the close epicenters, the so-called near-fault set, consists of 56 records from earthquakes with magnitudes from 6.5 to 7.6 measured at sites from 1.7–8.8 km from the epicenter and with relatively firm local soil conditions. Twenty-eight of the 56 "near-fault" records exhibit a distinct pulse. The far-field ground motion data set forms the basis for the ground motion model in the example in Section 30.6.

Fitting the two envelope parameters to the sets of ground acceleration records, and fitting the two the filter parameters to the spectral acceleration of the recorded ground motion while scaling each record to match the specified PGV value, results in the parameter values shown in Table 30.1. The peak ground

TABLE 30.1 Parameter Values for Earthquake Ground Motions

Set	$\bar{a}$	PGA	PGV	α	β	f_g	ζ_g
FF	63.6	361	33	4	2	1.5	0.9
NFNP	87.7	537	52	3	2	1.3	1.1
NFP	124	525	80	1	2	0.5	1.8
	cm/s^2	cm/s^2	cm/s	-	s	Hz	–

Abbrevations: FF = far-field; NFNP = near-fault without pulse; and NFP = near-fault with pulse.

acceleration (PGA) values reported in Table 30.1 correspond to the average PGA values for a set of simulated records, each having the specified PGV. The filter input parameter $\bar{a}$ values in Table 30.1 would result in average PGV values matching the specified PGV values, on average.

Examples of synthetic ground motion records for far-field and near-fault cases are shown in Figure 30.3. Note that the characteristic time for the velocity pulse in the near-fault example is commensurate with the time scale of the envelope function. This type of earthquake ground motion is much less stationary in nature than the "far-field" record, in which the ground motion record contains many cycles of motion within the duration of the record. Figure 30.4 illustrates the mean and mean-plus-standard deviation spectral accelerations of the original ground motion records as well as for the stochastic ground motion model. Note here that the model agrees with the original data, at least to a rough approximation, and that the coefficient of variation associated with the model ground motions is no greater than that of the original data set. The spectral accelerations of the near-fault model are about twice those of the far-field model at natural periods of 2–3 s.

More detailed descriptions of stochastic ground motion [6] are linked explicitly to the characteristics of the fault rupture, the geophysical environment of the earthquake, and the local site conditions. For the purposes of control design and analysis, the simplified models presented here are representative of two important classes of ground motion disturbances and are calibrated with an adequate sample of historical records.

30.3 Performance Measures for Control Design

In earthquake engineering, one of the difficulties associated with control problems is that the closed-loop performance measures which best motivate the application of sophisticated control technology are not

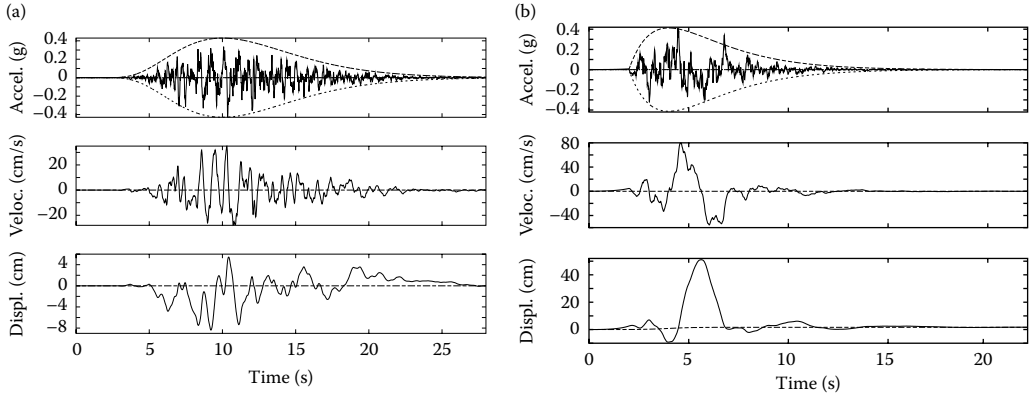

FIGURE 30.3 Representative records from far-field and impulsive near-fault records. The envelope function is applied to the acceleration record. (a) Far-field data set and (b) near-fault with pulse data set.

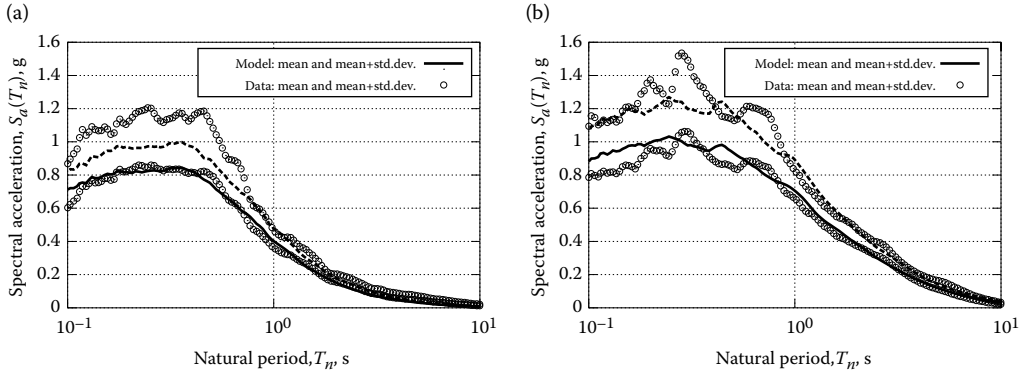

FIGURE 30.4 Mean and mean-plus-standard deviation spectral accelerations for two earthquake data sets and the corresponding spectral accelerations from models. (a) Far-field. (b) near-fault with pulse.

the most tractable from a synthesis point of view. Ultimately, the implementation of control in buildings and bridges is motivated solely by the idea that through control, these structures will be made more reliable under seismic excitation. In structural engineering, a persuasive interpretation of "reliability" is the probability that $\mathbf{z}$ will remain inside a safe region $\mathcal{D}_z$ for the entirety of an uncertain disturbance. Oftentimes, $\mathcal{D}_z$ is taken as a hypercube in $\mathbb{R}^{n_z}$, and components of $\mathbf{z}$ are referred to as a "failure modes." The set of failure modes typically includes survivability-related quantities (such as inter-story drifts and structural stresses), serviceability-related quantities (such as the absolute accelerations of sensitive locations of the structure), and possibly quantities related to limitations of the control hardware (such as force levels), each normalized by thresholds of comparable severity. Assessment of the probability of first passage, that is,

$$P_{\mathcal{D}_z} = P\left[\mathbf{z}(t) \notin \mathcal{D}_z \ \text{ for some } t \in [0, \infty)\right] \tag{30.16}$$

is then predicated on a probabilistic model of the disturbance $\mathbf{a}(t)$, such as that described in Section 30.2.

Note that the use of Equation 30.16, as an objective function in optimal design, does not require (Equation 30.1a) to be linear. In modern structural reliability analysis for passive structures, it never is. Among other reasons, this is because the passive structural dissipation mechanisms in steel and concrete structures are inherently hysteretic. However, for structures equipped with control systems, for which it is assumed that structural response is regulated primarily by control devices and not by passive energy dissipation in the main structure, the linearity assumption for Equation 30.1a is often assumed in order to make the control design problem tractable. However, even when linearity is assumed, the explicit optimization of the probability of first-passage through the boundary of $\mathcal{D}_z$, over a probabilistically parametrized domain for $\mathbf{a}$, is not analytically tractable except under very particular circumstances, all of which seem to require unconstrained (i.e., active) control devices. More generally, such optimal control problems can only be approached via large-dimensional simulation-based optimization techniques. Such approaches can be useful, but have the disadvantage of only providing a case-specific control design, without giving much in the way of broader insight into the nature of semiactive control problem.

In this chapter, we report on semiactive control design techniques which leverage existing results from optimal control theory, to arrive at easily computable feedback laws. Even though the control laws we will discuss are suboptimal, they do have analytically provable and easily computable upper bounds on performance. The performance measures chosen for optimization do not exactly align with the reliability-based objectives which would be ideal. We begin by adjusting our definition of $\mathcal{D}_z$, redefining it as the neighborhood $\{\mathbf{z} \in \mathcal{L}_\infty \ : \|\mathbf{z}\|_{\mathcal{L}_\infty} \leqslant 1\}$. As such, we have substituted an included spherical domain for

the hypercube, which would ideally be used to characterize $\mathcal{D}_z$. We next consider two standard control-theoretic performance measures, which under different circumstances, serve as reasonable surrogate performance measures for control design.

30.3.1 Near-Fault Design

First, we consider the the worst-case peak gain, as

$$J_{peak} = \sup_{\|\mathbf{a}\|_{\mathcal{L}_\infty} \leqslant 1} \|\mathbf{z}\|_{\mathcal{L}_\infty} \tag{30.17}$$

Such a measure is reasonable if the primary objective of the controller is to protect against near-fault phenomena. Because the semiactive constraint is homogeneous, the optimal feedback law will be the same for any upper bound $\|\mathbf{a}\|_{\mathcal{L}_\infty}$.

The probabilistic interpretation is that if $\bar{a} = \|\mathbf{a}\|_{\mathcal{L}_\infty}$ is uncertain, with a probability density $\rho(\bar{a})$, then it follows that the probability $P_{\mathcal{D}_z}$ is bounded by

$$P_{\mathcal{D}_z} \leqslant \int_0^\infty hvs\left(J_{peak}\bar{a} - 1\right) \rho(\bar{a})\, d\bar{a} \tag{30.18}$$

The above bound is usually quite conservative, in the sense that the $\mathbf{a}(t)$ yielding the worst-case $\|\mathbf{z}\|_{\mathcal{L}_\infty}$, conditioned on $\|\mathbf{a}\|_{\mathcal{L}_\infty} = \bar{a}$, may be highly unlikely. However, it does provide a useful upper bound on the probability $P_{\mathcal{D}_z}$ associated with an earthquake with a probabilistic characterization of peak intensity. Moreover, independently of $\rho(\bar{a})$, the minimization of this bound is equivalent to the minimization of J_{peak}, about which a considerable body of knowledge exists.

30.3.2 Far-Field Design

For geographical sites which are not too close to an epicenter, the envelope function $e(t)$ changes more slowly, and there is also more significant high-frequency content in $\mathbf{a}(t)$ which leads to rather high values for $\|\mathbf{a}\|_{\mathcal{L}_\infty}$, even though much of this high-frequency content is filtered by the structure. As such, the use of J_{peak} as the optimization objective can lead to extremely conservative design.

In this case one can assume $e(t)$ has time constants which are slower than those of the structure, and that the optimal controller for constant $e(t) = e_0$ will be similar to that associated with slowly varying $e(t)$. Because the semiactive constraint is homogeneous, the optimal controller for any magnitude of e_0 will be the same, and we may thus take it to be unity, without loss of generality. Furthermore, it is commonly assumed that, for control design purposes, the response of $\mathbf{z}(t)$ with $e(t) = e_0$ has statistics which are approximately stationary.

With these assumptions, we can reëxpress Equation 30.1 in terms of a white-noise input $\mathbf{w}(t) \in \mathbb{R}^{n_w}$ with spectral intensity equal to $\mathbf{I}$, as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_f\mathbf{f} + \mathbf{B}_w\mathbf{w} \tag{30.19a}$$

$$\mathbf{v} = \mathbf{B}_f^T\mathbf{x} \tag{30.19b}$$

$$\mathbf{y} = \mathbf{C}_y\mathbf{x} + \mathbf{D}_{yw}\mathbf{w} \tag{30.19c}$$

$$\mathbf{z} = \mathbf{C}_z\mathbf{x} + \mathbf{D}_{zf}\mathbf{f} \tag{30.19d}$$

where $\mathbf{x}$ has been augmented to include the dynamic states of the earthquake disturbance filter $G_{wk}(j\omega)$, for each acceleration direction k. As in standard linear optimal stochastic control design, $\mathbf{w}$ may be defined to include measurement noise, in addition to the exogenous noise injection to the disturbance filter.

Because our motivation is to protect against first-passage events, we are primarily interested in suppressing the tails of the response distribution for $\mathbf{z}(t)$. Consequently, controllers which tend to perform well for far-field earthquakes are those which suppress higher order moments in the distribution for $\mathbf{z}(t)$.

The stationarity assumption allows us to presume this objective as being the same for all t. If it is assumed that this distribution is approximately Gaussian in closed-loop (which is not strictly true, due to the nonlinearity of the semiactive control law, but is often a reasonable approximation), then this objective is equivalent to suppressing second-order moments. This is the rationale behind the prevalent use of standard Quadratic-Gaussian measure, that is,

$$J_{quad} = \mathcal{E}\,\mathbf{z}^T\mathbf{z} \tag{30.20}$$

as a design objective in earthquake response control, with the stochastic dynamics of $\mathbf{a}(t)$ modeled as described above. This optimization objective has been shown by many researchers to be quite effective for design of earthquake response controllers for the far-field case. However, it is important to recognize the large number of assumptions and approximations that are necessary to connect it with the original, reliability-based motivation.

30.4 Control Design

Almost all the successful techniques for semiactive control synthesis are developed first in a state-feedback context, and then extended to output feedback through the imposition of certainty-equivalence on a state estimate obtained with a sufficiently-high-bandwidth Luenberger observer or a Kalman–Bucy filter. This practice has some subtle theoretical problems associated with it, which will be discussed in Subsection 30.4.4. For now, we simply state that the difficulties for semiactive output feedback stem from the elusiveness of a separation principle analogous to that associated with, for example, linear $\mathcal{H}_2$ and $\mathcal{H}_\infty$ problems.

As such, for the time being we have that $\mathbf{y} = \mathbf{x}$. In Subsection 30.4.1 through 30.4.3 below, we have attempted to distill many of the more common state feedback techniques in this area, into a more unified framework. Although many are not usually presented this way, most full-state feedback algorithms can be framed as a generalized saturation, that is,

$$\mathbf{f}(t) = \phi(\mathbf{x}(t)) = \underset{\mathbf{f}\in\mathcal{S}(\mathbf{x}(t))}{\operatorname{argmin}} \|\mathbf{f} - \mathbf{K}\mathbf{x}(t)\|_{\mathbf{R}}^2 \tag{30.21}$$

where $\mathbf{R} > 0$ and $\mathcal{S}(\mathbf{x}(t)) \subseteq \mathcal{F}(\mathbf{v}(t))$ is convex and nonempty for all $\mathbf{x}(t)$. With these assumptions, the minimization above has a unique extremum for all $\mathbf{x}(t)$.

As such, the above controller attempts to match control force $\mathbf{f}(t)$ to a time-invariant linear full-state feedback law $\mathbf{K}\mathbf{x}(t)$, subject to constraints. When such matching is impossible for $\mathbf{f} \in \mathcal{S}(\mathbf{x}(t))$, weighting matrix $\mathbf{R}$ determines the measure of closeness of $\mathbf{f}(t)$ to $\mathbf{K}\mathbf{x}(t)$. Methods differ merely by the way they synthesize $\mathbf{K}$, $\mathbf{R}$, and $\mathcal{S}$.

30.4.1 Lyapunov-Bounded Design

The simplest control synthesis tools for semiactive systems are in the context of Lyapunov theory, and are oriented around the concept of quadratic stability. There are many control design approaches in the literature which may be interpreted as special cases of this framework, including most methods based on physical energy absorption heuristics. It is not the purpose of this chapter to provide a complete survey of all the techniques which fall in this category. However, we do single out the work of Leitmann and Reithmeier [8,9], whose research in this area appears to generalize that of many other related techniques.

For the positive-semidefinite quadratic form $V(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x}$, its derivative is

$$\dot{V} = \mathbf{x}^T\left[\mathbf{P}\left(\mathbf{A} - \mathbf{B}_f\mathbf{U}\mathbf{B}_f^T\right) + \left(\mathbf{A} - \mathbf{B}_f\mathbf{U}\mathbf{B}_f^T\right)^T\mathbf{P}\right]\mathbf{x} + 2\mathbf{x}^T\mathbf{P}\mathbf{B}_a\mathbf{a} \tag{30.22}$$

which, for some $\mathbf{U}_0 \in \mathcal{U}$, is equivalent to

$$\dot{V} = \mathbf{x}^T \left[\mathbf{P}\left(\mathbf{A} - \mathbf{B}_f \mathbf{U}_0 \mathbf{B}_f^T\right) + \left(\mathbf{A} - \mathbf{B}_f \mathbf{U}_0 \mathbf{B}_f^T\right)^T \mathbf{P} \right] \mathbf{x} + 2\mathbf{x}^T \mathbf{P}\left(\mathbf{B}_a \mathbf{a} - \mathbf{B}_f(\mathbf{U} - \mathbf{U}_0)\mathbf{v}\right) \tag{30.23}$$

We know that for any $\mathbf{U}_0 \in \mathcal{U}$, $\mathbf{A} - \mathbf{B}_f \mathbf{U}_0 \mathbf{B}_f^T$ is Hurwitz. As such, we can choose $\mathbf{P}$ as the solution to the Lyapunov equation

$$\mathbf{P}\left(\mathbf{A} - \mathbf{B}_f \mathbf{U}_0 \mathbf{B}_f^T\right) + \left(\mathbf{A} - \mathbf{B}_f \mathbf{U}_0 \mathbf{B}_f^T\right)^T \mathbf{P} + \mathbf{Q} = \mathbf{0} \tag{30.24}$$

for some $\mathbf{Q} \geqslant 0$. By Lasalle's theorem, observability of $(\mathbf{A} - \mathbf{B}_f \mathbf{U}_0 \mathbf{B}_f^T, \mathbf{Q})$ implies $\mathbf{P} > 0$. Equation 30.23 reduces to

$$\dot{V} = -\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{x}^T \mathbf{P}\left(\mathbf{B}_a \mathbf{a} - \mathbf{B}_f(\mathbf{U} - \mathbf{U}_0)\mathbf{v}\right) \tag{30.25}$$

$$= -\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{x}^T \mathbf{P}\left(\mathbf{B}_a \mathbf{a} + \mathbf{B}_f \mathbf{U}_0 \mathbf{v}\right) + 2\mathbf{x}^T \mathbf{P} \mathbf{B}_f \mathbf{f} \tag{30.26}$$

The most basic interpretation of Lyapunov-based control is to design the full-state feedback law $\phi : \mathbf{x}(t) \to \mathbf{f}(t)$ so as to minimize this derivative at every time; that is,

$$\mathbf{f}(t) = \underset{\mathbf{f} \in \mathcal{F}(\mathbf{v}(t))}{\operatorname{argmin}} \mathbf{x}^T(t) \mathbf{P} \mathbf{B}_f \mathbf{f} \tag{30.27}$$

For the usual case of $\mathcal{F}$ diagonally constrained as in Equation 30.8, this results in the simple element-by-element "bang-bang" control law for the diagonal components, as

$$f_i(t) = -v_i \operatorname{hvs}\left\{ v_i \mathbf{B}_{f_i}^T \mathbf{P} \mathbf{x} \right\} \tag{30.28}$$

where $\mathbf{B}_{f_i}$ is the ith column of $\mathbf{B}_f$. This leads to the closed-loop derivative

$$\dot{V} = -\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{x}^T \mathbf{P} \mathbf{B}_a \mathbf{a} + \operatorname{tr}\left\{ \mathbf{B}_f^T \mathbf{x} \mathbf{x}^T \mathbf{P} \mathbf{B}_f \left(2\mathbf{U}_0 - \mathbf{I}\right) - \left| \mathbf{B}_f^T \mathbf{x} \mathbf{x}^T \mathbf{P} \mathbf{B}_f \right| \right\} \tag{30.29}$$

where $|\cdot|$ denotes element-by-element absolute value. The trace above is always negative, and thus we have that

$$\dot{V} \leqslant -\mathbf{x}^T \mathbf{Q} \mathbf{x} + 2\mathbf{x}^T \mathbf{P} \mathbf{B}_a \mathbf{a} \tag{30.30}$$

The above approach can result in sliding mode behavior for the closed-loop system, and this is one of its major drawbacks. Indeed, control designs based on the above idea often lead to very high accelerations in the structure, due to the instantaneous switching operations of the semiactive devices. The switching surfaces for the controller are the subspaces characterized by $\mathbf{B}_{f_i}^T \mathbf{P} \mathbf{x} = 0$ and $\mathbf{B}_{f_i}^T \mathbf{x} = 0$. The latter of these cannot be a sliding surface because the control force $f_i = -u_i(t) \mathbf{B}_{f_i}^T \mathbf{x}$ is zero on either side of the switching surface. However, the former can indeed produce sliding modes, and it is important to ensure that $\dot{V} < 0$ on the surface, where $\mathbf{U}(t)$ is undefined. This can be assured by requiring that $\mathbf{x}^T \mathbf{Q} \mathbf{x} < 0$ for x in $\mathcal{N}\{\mathbf{B}_f^T \mathbf{P}\}$.

Alternatively, sliding mode behavior can be avoided (and accelerations reduced) by augmenting the right-hand side of Equation 30.23 with $\mathbf{v}^T(\mathbf{U} - \mathbf{U}_0)^T \mathbf{R}(\mathbf{U} - \mathbf{U}_0)\mathbf{v}$, for some $\mathbf{R} > 0$. Then, the semiactive controller ϕ is the minimizer of this augmented expression, which is equivalent to

$$\mathbf{f}(t) = \underset{\mathcal{F}(\mathbf{v}(t))}{\operatorname{argmin}} \left\| \mathbf{f} + \left(\mathbf{U}_0 \mathbf{B}_f^T + \mathbf{R}^{-1} \mathbf{B}_f^T \mathbf{P}\right) \mathbf{x}(t) \right\|_{\mathbf{R}}^2 \tag{30.31}$$

Note that this has the form of Equation 30.21, with $\mathcal{S}(\mathbf{x}(t)) = \mathcal{F}(\mathbf{v}(t))$, $\mathbf{K} = -\mathbf{U}_0 \mathbf{B}_f^T - \mathbf{R}^{-1} \mathbf{B}_f^T \mathbf{P}$, and $\mathbf{R}$ any positive-definite matrix. For $\mathcal{U}$ diagonal and $\mathcal{F}$ constrained by Equation 30.8, choosing $\mathbf{R} = \operatorname{diag}\{...r_i...\}$ results in simplification of the above controller to element-by-element saturation, that is,

$$f_i(t) = -v_i(t) \underset{[0,1]}{\operatorname{sat}} \left\{ u_{0i} + \frac{1}{r_i v_i(t)} \mathbf{B}_{f_i}^T \mathbf{P} \mathbf{x}(t) \right\} \tag{30.32}$$

The resultant closed-loop system still adheres to Equation 30.30, but the degree to which $\dot{V}$ is reduced beyond this bound will decrease as $\mathbf{R}$ is increased.

We now know how to synthesize $\mathbf{K}$, given $\mathbf{Q}$ and $\mathbf{U}_0$, but it remains to be shown how these terms are chosen. One of the more popular approaches for this is the technique proposed by Leitmann in [8]. By the Cauchy–Schwartz inequality, Equation 30.30 implies that

$$\dot{V} \leqslant -\mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\left(\mathbf{x}^T\mathbf{P}\mathbf{x}\right)^{1/2}\left(\mathbf{a}^T\mathbf{B}_a^T\mathbf{P}\mathbf{B}_a\mathbf{a}\right)^{1/2} \tag{30.33}$$

Recognizing that $\mathbf{a}^T\mathbf{B}_a^T\mathbf{P}\mathbf{B}_a\mathbf{a} \leqslant \lambda_{max}\{\mathbf{B}_a^T\mathbf{P}\mathbf{B}_a\}\mathbf{a}^T\mathbf{a}$ and $\mathbf{x}^T\mathbf{Q}\mathbf{x} \geqslant \lambda_{min}\{\mathbf{Q}\mathbf{P}^{-1}\}\mathbf{x}^T\mathbf{P}\mathbf{x}$, the above implies the bound

$$\frac{\|\sqrt{\mathbf{P}}\mathbf{x}\|_{\mathcal{L}_\infty}}{\|\mathbf{a}\|_{\mathcal{L}_\infty}} \leqslant \frac{2\sqrt{\lambda_{max}\left\{\mathbf{B}_a^T\mathbf{P}\mathbf{B}_a\right\}}}{\lambda_{min}\left\{\mathbf{Q}\mathbf{P}^{-1}\right\}} \tag{30.34}$$

Recognizing that $\|\sqrt{\mathbf{P}}\mathbf{x}\|_{\mathcal{L}_\infty} \geqslant \sqrt{\lambda_{min}\{\mathbf{P}\}}\|\mathbf{x}\|_{\mathcal{L}_\infty}$, $\lambda_{max}\{\mathbf{B}_a^T\mathbf{P}\mathbf{B}_a\} \leqslant \sigma_{max}^2\{\mathbf{B}_a\}\lambda_{max}\{\mathbf{P}\}$, and $\lambda_{min}\{\mathbf{Q}\mathbf{P}^{-1}\} \geqslant \lambda_{min}\{\mathbf{Q}\}/\lambda_{max}\{\mathbf{P}\}$, we obtain the "Leitmann bound", as

$$\frac{\|\mathbf{x}\|_{\mathcal{L}_\infty}}{\|\mathbf{a}\|_{\mathcal{L}_\infty}} \leqslant \frac{2\sigma_{max}\{\mathbf{B}_a\}\lambda_{max}\{\mathbf{P}\}^{3/2}}{\lambda_{min}\{\mathbf{Q}\}\lambda_{min}\{\mathbf{P}\}^{1/2}} \tag{30.35}$$

Generally, designs based on Equation 30.35 involve two steps. The first of these involves finding the $\mathbf{U}_0^* \in \mathcal{U}$ which minimizes the bound above, and the corresponding $\mathbf{P}^*$. The approach can be made more tractable by assuming $\mathbf{Q}$ to be some (not necessarily optimal) value, such as $\mathbf{Q} = \mathbf{I}$. Then, the first step is accomplished by optimizing $\mathbf{U}_0$ to minimize $\lambda_{max}\{\mathbf{P}\}^3/\lambda_{min}\{\mathbf{P}\}$. Although generally nonconvex, this optimization is tractable in the context of static optimal feedback design. The second step involves imposing the semiactive control law as in Equation 30.32, using $\mathbf{P} = \mathbf{P}^*$, and a value of $\mathbf{R}$, which provides a sufficient degree of smoothness near the switching surfaces. This controller then guarantees to improve upon the optimal bound attained with $\mathbf{U}(t) = \mathbf{U}_0^*,\ \forall t$.

It should be noted that the Leitmann bound is rather conservative, and that less-conservative bounds may be used in its place, with little modification to the theory. One such bound is

$$\frac{\|\mathbf{x}\|_{\mathcal{L}_\infty}}{\|\mathbf{a}\|_{\mathcal{L}_\infty}} \leqslant \frac{2\sqrt{\lambda_{max}\{\mathbf{B}_a^T\mathbf{P}\mathbf{B}_a\}/\lambda_{min}\{\mathbf{P}\}}}{\lambda_{min}\{\mathbf{Q}\mathbf{P}^{-1}\}} \tag{30.36}$$

In this case, even if $\mathbf{Q}$ is fixed at an assumed value, the optimization above over $\mathbf{U}_0 \in \mathcal{U}$ is more complicated, algebraically.

A related family of Lyapunov-based approaches specify $\mathbf{P}$ directly, constrained to the requirement that $(\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T)^T\mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T) \leq 0$. For example, if we choose $\mathbf{P} = \mathbf{I}$, this results in $V(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$, which (due to our assumption of a self-dual realization) is a Lyapunov function for the structure irrespective of how $\mathbf{U}(t)$ is controlled. When combined with the further convention that $\mathbf{x}^T\mathbf{x}$ is equal to the total structural energy in free response, this is a common approach to semiactive control design. Such controllers maximize the instantaneous mechanical power absorption. However, in this case $\mathbf{Q}$ is only positive semidefinite, and the Leitmann bound is equal to infinity. Thus, as reported in [8], energy-based semiactive control methods, although popular, are generally not the most useful methods for ensuring bounds on closed-loop performance.

30.4.2 J_{peak}-Bounded Design

The Lyapunov-based designs discussed above have the disadvantage of being realization-dependent. Their use in design thus implicitly requires that the state space be scaled such that all points on the boundary of the neighborhood $\|\mathbf{x}\| = 1$ represent responses of comparable severity. On its own, this is not problematic. However, the eigenvalues $\lambda_{max}\{\mathbf{P}\}$ and $\lambda_{min}\{\mathbf{Q}\}$ may correspond to principal axes in the state space which are far from the subspaces of importance, thus making the bounds rather conservative. Additionally, these methods guarantee bounds on time-invariant quadratic forms involving only the system state. They

cannot be used to guarantee bounds on functions involving the control force. In application to civil engineering structures this is a significant drawback, because the fundamental tradeoff in control design is often between structural deformation and absolute acceleration, the latter of which can involve control forces explicitly.

However, similar synthesis methods exist which are based only on input–output behavior, and are thus realization-independent. Indeed, Lyapunov-based methods are actually special cases of semiactive controllers which ensure a bound on J_{peak}. One such approach is discussed below. This technique also admits (but does not require) the inclusion of control force terms in J_{peak}.

Consider the case in which the performance vector is defined as $\mathbf{z} = \mathbf{C}_z\mathbf{x} + \mathbf{D}_{zf}\mathbf{f}$ (i.e., $\mathbf{D}_{za} = \mathbf{0}$.). For time-invariant $\mathbf{U}(t) = \mathbf{U}_0 \in \mathcal{U}$, consider that if $\mathbf{P} > 0$ satisfies

$$\dot{V} = \tfrac{d}{dt}\mathbf{x}^T\mathbf{P}\mathbf{x} = \begin{bmatrix}\mathbf{x}\\ \mathbf{a}\end{bmatrix}^T \begin{bmatrix}-\mathbf{Q}(\mathbf{P},\mathbf{U}_0) & \mathbf{P}\mathbf{B}_a\\ \mathbf{B}_a^T\mathbf{P} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{a}\end{bmatrix} \leqslant 0 \tag{30.37}$$

over the region $\mathbf{a}^T\mathbf{a} \leqslant \mathbf{x}^T\mathbf{P}\mathbf{x}$, where $\mathbf{Q}(\mathbf{P},\mathbf{U}_0) = -(\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T)^T\mathbf{P} - \mathbf{P}(\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T)$, then $\|\sqrt{\mathbf{P}}\mathbf{x}\|_{\mathcal{L}_\infty}/\|\mathbf{a}\|_{\mathcal{L}_\infty} \leqslant 1$. Furthermore enforcing the additional inequality

$$\begin{bmatrix}\mathbf{P} & \mathbf{C}_z^T - \mathbf{B}_f\mathbf{U}_0^T\mathbf{D}_{zf}\\ \mathbf{C}_z - \mathbf{D}_{zf}\mathbf{U}_0\mathbf{B}_f^T & \mathbf{I}\gamma^2\end{bmatrix} > 0 \tag{30.38}$$

ensures that $J_{peak} < \gamma$. Given $\mathbf{U}_0$, $\{\mathbf{P},\gamma\}$ can be optimized via semidefinite programming. This is an elementary application of the $\mathcal{S}$-procedure [10], which states that Equation 30.37 holds whenever $\mathbf{a}^T\mathbf{a} \leqslant \mathbf{x}^T\mathbf{P}\mathbf{x}$, if there exists $\tau > 0$ such that

$$\begin{bmatrix}-\mathbf{Q}(\mathbf{P},\mathbf{U}_0) & \mathbf{P}\mathbf{B}_a\\ \mathbf{B}_a^T\mathbf{P} & \mathbf{0}\end{bmatrix} + \tau\begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & -\mathbf{I}\end{bmatrix} \leqslant 0 \tag{30.39}$$

Any combination of $\{\mathbf{P},\mathbf{U}_0,\tau,\gamma\}$ resulting in feasibility of Equations 30.38 and 30.39 is thus guaranteed to adhere to the bounds $\|\sqrt{\mathbf{P}}\mathbf{x}\|_{\mathcal{L}_\infty}/\|\mathbf{a}\|_{\mathcal{L}_\infty} \leqslant 1$ and $J_{peak} < \gamma$.

The minimization of γ, subject to Equations 30.39 and 30.38, is nonconvex over $\{\mathbf{P} > 0, \mathbf{U}_0 \in \mathcal{U}, \gamma > 0, \tau > 0\}$. However, for fixed $\{\mathbf{U}_0,\tau\}$, the matrix inequalities become linear, and may be solved uniquely for optimal $\{\mathbf{P},\gamma\}$ using any standard LMI-based optimization technique, including interior-point or primal-dual methods. By extension, this optimization may be nested in a one-dimensional search over τ, for the minimal bound γ associated with a fixed $\mathbf{U}_0$. Furthermore, it is straight-forward to show that the feasibility domain for τ is compact, and bounded by $[0, -\max_k \frac{1}{2}\,\mathrm{Re}\,\lambda_k\{\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T\}]$.

Although the optimization of γ over $\mathbf{U}_0 \in \mathcal{U}$ is nonconvex, it is tractable for low n_f through any of a variety of LMI-based static feedback optimization techniques. The most straight-forward of these involves iteratively re-solving for $\{\mathbf{P},\mathbf{U}_0,\tau,\gamma\}$ through completion of the square. Starting from some fixed and feasible $\{\mathbf{U}_0^k,\tau^k\}$, we find $\{\mathbf{P}^k,\gamma^k\}$ for minimal γ^k via a convex optimization, and then note that for all feasible $\{\mathbf{U}_0,\mathbf{P},\tau,\gamma\}$,

$$\begin{aligned}-\mathbf{Q}(\mathbf{P},\mathbf{U}_0) + \tau\mathbf{P} \leqslant\ & \tau^k\mathbf{P} + \tau\mathbf{P}^k - \tau^k\mathbf{P}^k - \mathbf{Q}(\mathbf{P},\mathbf{U}_0^k) - \mathbf{Q}(\mathbf{P}^k,\mathbf{U}_0) + \mathbf{Q}(\mathbf{P}^k,\mathbf{U}_0^k)\\ & + \mathbf{B}_f\left(\mathbf{U}_0 - \mathbf{U}_0^k\right)^T\mathbf{W}_1^{-1}\left(\mathbf{U}_0 - \mathbf{U}_0^k\right)\mathbf{B}_f^T + \mathbf{W}_2^{-1}\left(\tau - \tau^k\right)^2\\ & + \left(\mathbf{P} - \mathbf{P}^k\right)\left(\mathbf{B}_f\mathbf{W}_1\mathbf{B}_f^T + \mathbf{W}_2\right)\left(\mathbf{P} - \mathbf{P}^k\right)\end{aligned} \tag{30.40}$$

for arbitrary $\mathbf{W}_1, \mathbf{W}_2 > 0$, with the equality holding at $\{\mathbf{U}_0^k,\mathbf{P}^k,\tau^k,\gamma^k\}$. Thus, we substitute the above into term (1, 1) of Equation 30.39, and use Schur transformations to get a more conservative, but linear, matrix inequality. Subject to this conservative LMI, together with Equation 30.38, we then find a new solution $\{\mathbf{U}_0^{k+1},\tau^{k+1},\mathbf{P}^{k+1},\gamma^{k+1}\}$ for minimal γ^{k+1}. Because the previous solution $\{\mathbf{U}_0^k,\mathbf{P}^k,\tau^k,\gamma^k\}$ is on

the boundary of the feasibility domain, it follows that $\gamma^{k+1} \leqslant \gamma^k$. Repetition of this procedure will converge to a locally optimal solution for $\mathbf{U}_0$.

Denote γ^* as the optimal γ obtainable via the above optimization, and take $\{\mathbf{U}_0^*, \mathbf{P}_0^* \tau_0^*\}$ to be other variables at this optimum. As with the Leitmann bound, the value of γ^* can be a rather conservative estimate of J_{peak}.

With $\mathbf{U}_0^*$ found, we now wish to find a nonlinear state feedback controller $\phi : \mathbf{x}(t) \to \mathbf{f}(t)$ which guarantees to improve on this bound, that is, for which it can be guaranteed that for the closed-loop system, there exists a $\gamma < \gamma^*$ such that $J_{peak} < \gamma$. Such a controller can be found by observing that for any $\mathbf{R} > 0$, any $\mathbf{f}(t)$ satisfying

$$\|\mathbf{f}(t) + \left(\mathbf{U}_0^* \mathbf{B}_f^T + \mathbf{R}^{-1}\mathbf{B}_f^T \mathbf{P}^*\right)\mathbf{x}(t)\|_{\mathbf{R}}^2 \leqslant \|\mathbf{R}^{-1}\mathbf{B}_f^T \mathbf{P}^* \mathbf{x}(t)\|_{\mathbf{R}}^2 \tag{30.41}$$

$$\|\mathbf{C}_z \mathbf{x}(t) + \mathbf{D}_{zf}\mathbf{f}(t)\|_2^2 \leqslant \mathbf{x}^T(t)\mathbf{P}^*\mathbf{x}(t)\gamma^{*2} \tag{30.42}$$

for all t guarantees to further reduce $\dot{V}(t)$ for every $\mathbf{x}(t)$, while still satisfying $\mathbf{z}^T(t)\mathbf{z}(t) \leqslant V(t)$. An effective strategy is to minimize the left-hand side of Equation 30.41 subject to the constraint that $\mathbf{f}(t)$ satisfy Equation 30.42. With this approach, we again arrive at a controller of the general form in Equation 30.21, with $\mathbf{R} > 0$ arbitrary, $\mathbf{K} = -\mathbf{U}_0^* \mathbf{B}_f^T - \mathbf{R}^{-1}\mathbf{B}_f^T \mathbf{P}^*$, and

$$\mathcal{S}(\mathbf{x}(t)) = \mathcal{F}(\mathbf{v}(t)) \cap \left\{\mathbf{f} \ : \ \|\mathbf{C}_z \mathbf{x}(t) + \mathbf{D}_{zf}\mathbf{f}\|_2^2 \leqslant \mathbf{x}^T(t)\mathbf{P}^*\mathbf{x}(t)\gamma^{*2}\right\} \tag{30.43}$$

This optimization is convex for all $\mathbf{x}(t)$, the solution is continuous in $\mathbf{x}$ for any $\mathbf{R} > 0$, and it can be found in a finite number of computational steps as a routine application of Lagrange multipliers.

Assigning $\mathbf{K}$, $\mathbf{R}$, and $\mathcal{S}$ as above, controller (Equation 30.21) is thus guaranteed to improve γ beyond γ^*, resulting in an improvement over the time-invariant optimal $\mathbf{U}_0^* \in \mathcal{U}$. However, the actual *margin* of this improvement is generally not computable. Furthermore, because γ^* is merely an upper bound for J_{peak} with $\mathbf{U}(t) = \mathbf{U}_0^*$, but not necessarily equal to it, it is not guaranteed that the true value of J_{peak} will be reduced through this controller. As such, the degree to which this design approach is useful will vary with the problem data.

At present, a semiactive control theory which guarantees to improve the *true* value of J_{peak} remains an open problem. It is clear that J_{peak} can be explicitly optimized over all $\mathbf{U}_0 \in \mathcal{U}$ as a constrained, static $\mathcal{L}_1$ optimal feedback problem, about which a considerable body of knowledge exists. The challenge lies in the design of the nonlinear full-state controller for $\mathbf{f}(t)$, such that an improvement in J_{peak} (relative to the case with $\mathbf{U}(t) = \mathbf{U}_0$) can be assured.

30.4.3 J_{quad}-Bounded Design

The design of controllers for bilinear semiactive systems, toward the minimization of J_{quad} as defined in (30.20), is the oldest form of the semiactive control problem in the literature. It dates back to the seminal 1974 paper by Karnopp et al., which introduces the concept of a variable-orifice semiactive damper, for use in automotive suspensions [11]. Subsequent analyses in the early 1980s, by both Karnopp [12] and Margolis [13], were among the first to investigate a technique, which has subsequently found wide use in earthquake response control applications, and which is usually called "clipped optimal" semiactive control. The idea is simply to design a linear LQG controller to minimize J_{quad} (presuming, for the output-feedback case, a certain level of measurement noise to design the associated Kalman–Bucy filter). Then, the constraint $\mathbf{f} \in \mathcal{F}(\mathbf{v})$ is imposed as an element-by-element saturation on the output of the LQG feedback law. For the state-feedback case, this gives a controller of the form in Equation 30.21, with $\mathcal{S}(\mathbf{x}) = \mathcal{F}(\mathbf{v})$, $\mathbf{R} = \mathbf{I}$, and $\mathbf{K}$ determined as the optimal gain associated with the LQG problem with performance J_{quad}.

In application to civil engineering structures, this technique can be quite effective [14], especially for problems in which there is only one control device, and only one dominant vibratory mode. However,

it does have some disadvantages. Generally, control design must be done iteratively, with performance evaluated through time-domain simulation, because no analytical expressions exist for the degree of depreciation from the optimal performance achieved by the presaturated LQG controller. Indeed, it is possible to define reasonable performance measures for which this depreciation is so large that the closed-loop system produces a performance J_{quad}, which is actually worse than that of an optimized set of linear viscous dampers [15]. Speaking broadly, clipped-optimal controllers can sometimes behave rather poorly when multiple structural modes are significant, or when structural acceleration suppression is especially important.

On the other hand, it is actually possible to solve the optimal control problem exactly, for the full-state feedback law $\phi : \mathbf{x} \rightarrow \mathbf{f}$ which minimizes J_{quad}. Some of the more general techniques for this are detailed by Ying et al., in [16]. The problem essentially distills to a stationary stochastic Bellman problem, with its solution involving the solution to an associated partial differential equation on $\mathbb{R}^{n_x}$, for the Bellman function $V(\mathbf{x})$. With $V(\mathbf{x})$ found, determination of the optimal control ϕ then just amounts to evaluation of the gradient of $V(\mathbf{x})$, and some routine algebra. Although the homogeneity of the bilinear semiactive control problem can be exploited to enhance the computational burden associated with the solution for $V(\mathbf{x})$, the trouble with this approach is the same curse of dimensionality that hampers many Bellman-type problems. Applications of this approach have mostly been applied to very simple structural systems, with only a few degrees of freedom.

In this chapter, we discuss an approach, which is a compromise between these two extremes. To the authors' knowledge, the technique we discuss originated in the automotive suspensions area with the work of Tseng and Hedrick [17], who called it "steepest gradient" control. Scruggs et al. investigated an analogous technique for civil applications, which is generalized to multidevice systems [15]. On the one hand, the resultant feedback laws are mathematically very simple. Indeed, ϕ has the form of Equation 30.21. Furthermore, the synthesis of ϕ scales well, computationally, to systems with higher dimensionality. On the other hand, the technique is suboptimal. However, it does have a very important property, in that it is guaranteed to yield performance J_{quad} which is better than the best performance achievable with time-invariant damping; that is, $\mathbf{U}(t) = \mathbf{U}_0, \forall t$. As such, it adheres to a bound which is meaningful from a technological point of view.

As with the case of Lyapunov-bounded and J_{peak}-bounded designs, this control design approach consists of two steps. The first consists of the optimization of fixed $\mathbf{U}_0 \in \mathcal{U}$, for minimal J_{quad}. This is an optimal static output-feedback LQG problem, with the added condition that the output feedback law (i.e., $\mathbf{U}_0$) is constrained to $\mathcal{U}$. As is well known, static output-feedback LQG problems do not in general have closed-form solutions, and can only be solved iteratively. Customarily, the primary challenge associated with these problems is the identification of a stabilizing initial guess for the convergence algorithm. However, for the class of problems we consider, this is a trivial, as asymptotic stability is known to hold for all $\mathbf{U}_0 \in \mathcal{U}$. Were $\mathbf{U}_0$ not constrained to $\mathcal{U}$, various contractive solution algorithms (such as the Levine–Athans iteration) could be used to arrive at the optimum. However, the existence of structural constraints on $\mathbf{U}_0$ motivates a gradient-based approach. The matrix gradient of J_{quad}, with respect to $\mathbf{U}_0$, is

$$\frac{\partial J_{quad}}{\partial \mathbf{U}_0} = -2\left(\mathbf{B}_f^T\mathbf{P}\mathbf{S}\mathbf{B}_f + \mathbf{D}_{zf}^T\left[\mathbf{C}_z - \mathbf{D}_{zf}\mathbf{U}_0\mathbf{B}_f^T\right]\mathbf{S}\mathbf{B}_f\right) \tag{30.44}$$

where $\mathbf{P}$ and $\mathbf{S}$ obey

$$\mathbf{0} = \left[\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T\right]^T\mathbf{P} + \mathbf{P}\left[\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T\right] + \left[\mathbf{C}_z - \mathbf{D}_{zf}\mathbf{U}_0\mathbf{B}_f^T\right]^T\left[\mathbf{C}_z - \mathbf{D}_{zf}\mathbf{U}_0\mathbf{B}_f^T\right] \tag{30.45}$$

$$\mathbf{0} = \left[\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T\right]\mathbf{S} + \mathbf{S}\left[\mathbf{A} - \mathbf{B}_f\mathbf{U}_0\mathbf{B}_f^T\right]^T + \mathbf{B}_w\mathbf{B}_w^T \tag{30.46}$$

The value of J_{quad}, as a function of $\mathbf{U}_0$ is

$$J_{quad} = \mathrm{tr}\left\{\mathbf{B}_w^T\mathbf{P}\mathbf{B}_w\right\} \tag{30.47}$$

These results, which are now standard in the control literature, can readily be used to obtain local optima in $\mathcal{U}$ via gradient-based convergence algorithms, and using Lagrange multipliers as necessary to enforce $\mathbf{U}_0 \in \mathcal{U}$ along the optimization path.

Let the optimal performance obtained via the above approach be J^*_{quad}, and let the optimal $\mathbf{U}_0$ be $\mathbf{U}^*_0$. Furthermore, let $\mathbf{P}^*$ and $\mathbf{S}^*$ be the corresponding solutions to the Lyapunov equations above with $\mathbf{U}_0 = \mathbf{U}^*_0$. Then the second step for the semiactive control design is to find a nonlinear controller $\phi : \mathbf{x} \to \mathbf{f}$ which guarantees to produce $J_{quad} < J^*_{quad}$. This can be done by recognizing an important property of stochastic control with quadratic performance. In general, it can be said that for any stabilizing feedback law ϕ (linear or nonlinear), the following equality holds:

$$J_{quad} = J^*_{quad} + \mathcal{E}\left\{\left\|\mathbf{D}_{zf}\left(\mathbf{f} - \mathbf{K}\mathbf{x}\right)\right\|_2^2 - \left\|\mathbf{D}_{zf}\left(-\mathbf{U}_0\mathbf{v} - \mathbf{K}\mathbf{x}\right)\right\|_2^2\right\} \tag{30.48}$$

where

$$\mathbf{K} = -\left[\mathbf{D}_{zf}^T\mathbf{D}_{zf}\right]^{-1}\left[\mathbf{B}_f^T\mathbf{P}^* + \mathbf{D}_{zf}^T\mathbf{C}_z\right] \tag{30.49}$$

This suggests a feedback controller of the form (Equation 30.21), in which $\mathcal{S}(\mathbf{x}) = \mathcal{F}(\mathbf{v})$, $\mathbf{R} = \mathbf{D}_{zf}^T\mathbf{D}_{zf}$, and $\mathbf{K}$ is synthesized as in Equation 30.49; that is, the controller chooses $\mathbf{f}(t)$ at each time to minimize the first norm in the expectation in Equation 30.48. Because $-\mathbf{U}_0\mathbf{v}(t) \in \mathcal{F}(\mathbf{v}(t))$, it follows that this results in a nonpositive difference between the two norms in Equation 30.48 for all t, and therefore results in a nonpositive expectation. The controller thus guarantees $J_{quad} \leqslant J^*_{quad}$. The inequality holds strictly, in all but very special cases.

We thus arrive at a synthesis technique like the one for J_{peak}-bounded control. In both cases, the resultant nonlinear controller attempts to make a quantity minimal at every time, and in doing so, outperforms the optimal feasible static controller. Also in both cases, the margin of this improvement cannot be solved generally in closed form.

30.4.4 Output Feedback

Consider, now, the design of dynamic feedback controllers $\phi : \mathbf{y} \to \mathbf{f} \in \mathcal{F}(\mathbf{v})$. We presume that $\mathbf{y}$ is measurable in the absence of noise. Recall that in order for the problem assumptions to be consistent, this assumption must be true at least for the subspace of $\mathbf{y}$ which contains $\mathbf{v}$, because $\mathbf{v}(t)$ must be known precisely at each t in order to impose $\mathbf{f}(t) \in \mathcal{F}(\mathbf{v}(t))$. For this problem, a reasonable strategy is to construct a casual state estimator which maps $\{\mathbf{f}, \mathbf{y}\} \to \boldsymbol{\xi}$, and then impose certainty equivalence on the state estimate $\boldsymbol{\xi}(t)$, resulting in the generalized semiactive output-feedback controller

$$\mathbf{f}(t) = \underset{\mathbf{f} \in \mathcal{S}(\mathbf{x}(t))}{\operatorname{argmin}} \|\mathbf{f} - \mathbf{K}\boldsymbol{\xi}(t)\|_{\mathbf{R}}^2 \tag{30.50}$$

with $\mathbf{K}$, $\mathbf{R}$, and $\mathcal{S}(\mathbf{x}(t))$ synthesized the same way they would be for full-state feedback, as described in the previous subsections.

Assuming the earthquake to be modeled as filtered white noise as in Equation 30.19 (as is done for J_{quad}-bounded design), our assumption of zero-noise feedback is equivalent to the assumption that $\mathbf{D}_{yw} = \mathbf{0}$. As such, the optimal estimator for the J_{quad}-bounded semiactive control problem, as it has been posed here, generally results in a singular optimal filtering problem; that is, one in which the optimal estimator dynamics are improper. (The root cause of this is the necessary assumption of precise knowledge for $\mathbf{v}$.) We can, however, obtain a suboptimal estimate $\boldsymbol{\xi}(t)$ by the design of a Kalman–Bucy filter with a low-intensity, fictitious measurement noise, resulting in a Kalman gain $\mathbf{L}$. So computed, the observer

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} + \mathbf{B}_f\mathbf{f} + \mathbf{L}\left(\mathbf{y} - \mathbf{C}_y\boldsymbol{\xi}\right) \tag{30.51}$$

will produce $\boldsymbol{\xi}$ which retains some residual bias (i.e., $\mathcal{E}\,\boldsymbol{\xi}(\mathbf{x} - \boldsymbol{\xi})^T \neq \mathbf{0}$), due to the presence of the fictitious measurement noise in the computation of $\mathbf{L}$. Because of the unavoidable nonlinearity of the closed-loop system, exact removal of this mean-square bias is challenging, and remains an unresolved issue in

the literature on semiactive control. However, qualitatively speaking, as one makes the bandwidth of the observer higher, the variance $\mathcal{E}(\mathbf{x}-\boldsymbol{\xi})(\mathbf{x}-\boldsymbol{\xi})^T$ typically becomes quite small. (However, it does not asymptotically approach zero, unless the particular problem exhibits minimum-phase dynamics between **a** and **y**. Details on this asymptotic case is covered elsewhere in this handbook.) As such, the issue of obtaining a true certainty-equivalence principle for J_{quad}-bounded semiactive control may, in many applications, be only of marginal benefit.

For the case of Lyapunov- or J_{peak}-bounded design, the system model is Equation 30.1, for which it is not necessary to model the dynamics of **a**. As such, if it is also assumed that $\mathbf{a}(t)$ can be measured precisely, and used as feedback along with **y**, then a true asymptotic observer of the form

$$\dot{\boldsymbol{\xi}} = \mathbf{A}\boldsymbol{\xi} + \mathbf{B}_f\mathbf{f} + \mathbf{B}_a\mathbf{a} + \mathbf{L}\left(\mathbf{y} - \mathbf{C}_y\boldsymbol{\xi} - \mathbf{D}_{ya}\mathbf{a}\right) \tag{30.52}$$

can be constructed, for which $\boldsymbol{\xi}(t) \to \mathbf{x}(t)$ with zero bias. The error dynamics are dictated by **L**, which can be designed by any of a variety of standard asymptotic observer techniques. However, in many cases, real-time feedback for **a** is not available, or contains significant noise. In this case asymptotic estimation of the entire system state is not generally possible, and observer design is often approached in the same way as it is for J_{quad}-bounded control.

Speaking more broadly, observers for semiactive systems exhibit some subtle challenges in the modeling of measurement uncertainty. In the above discussion, we have presumed **v** and **f** to be known precisely. Because the control design must presume knowledge of the semiactive constraint domain $\mathcal{F}(\mathbf{v})$, knowledge of one necessarily implies knowledge of the other. However, another way of approaching the semiactive output feedback problem is to presume $\mathbf{U}(t)$ to be known precisely, while both **v** and **f** are presumed uncertain. It is less common for the device characterizations for semiactive systems to adhere to this uncertainty model. Nonetheless, for J_{quad}-bounded control this has been examined by Scruggs et al. in [15], where it is shown that this problem results in a feedback law which still guarantees the same bound on J_{quad} as with state feedback. However, in this case the form of the controller is more complicated than that of Equation 30.50, and does not exhibit certainty-equivalence. Rather, the controller for $\mathbf{U}(t)$ must keep track of the time-varying covariance matrix for the state estimate, and must explicitly balance the dual tasks of "good estimation" and "good control."

30.5 Nonideal Device Models

The semiactive control theory discussed in Section 30.4 is idealized, in two distinct senses. First, it presumes that the constraint $\mathbf{f} \in \mathcal{F}(\mathbf{v})$ is homogeneous in **v**; or equivalently, that the $\mathcal{F}(\mathbf{v})$ constitutes a double-infinite cone in $\{\mathbf{f}, \mathbf{v}\}$ space. In actuality, all devices violate this assumption to one degree or another. Second, it presumes that **U** can be instantaneously transitioned from one value in $\mathcal{U}$ to another. In actuality, semiactive devices exhibit dynamics which limit their dynamic capabilities. In this section, we talk briefly about these issues, and how the control theory in Section 30.4 can be adjusted to accommodate more realistic device models.

30.5.1 Inhomogeneity of $\mathcal{F}(\mathbf{v})$

Most semiactive devices have a maximum force, beyond which saturation occurs. This is the case, for example, in most hydraulic dampers. Moreover, many controllable dampers have a force capability which resembles controllable friction; that is,

$$f_k(t) = -u_k(t)\,\mathrm{sgn}(v_k(t)) \tag{30.53}$$

in which $u_k(t) \in [0, 1]$ is the independent control variable. Many semiactive devices have $\mathcal{F}(\mathbf{v})$ regions which exhibit more complicated shapes.

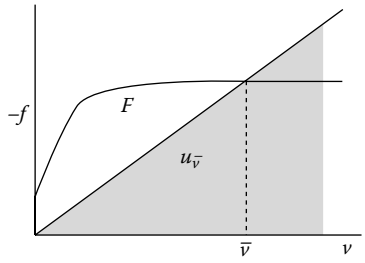

FIGURE 30.5 Example of conservative approximation of an inhomogeneous semiactive device by a homogeneous one, for $v \leqslant \bar{v}$.

The simplest way to accommodate these issues in the context of the theory in Section 30.4 is to first characterize a velocity $\bar{v}_k$ for each device $k \in \{1 \ldots m\}$, which represents the maximum velocity which can be reasonably anticipated to occur during the dynamic response. Then, for many device characterizations there exists a set $\mathcal{U}_{\bar{v}}$ such that

$$-\mathbf{U}\mathbf{v} \in \mathcal{F}(\mathbf{v}),\ \forall \mathbf{v}, \mathbf{U}\ :\ |v_k| \leqslant \bar{v}_k,\ \mathbf{U} \in \mathcal{U}_{\bar{v}} \tag{30.54}$$

This is illustrated in Figure 30.5. Choosing the largest possible $\mathcal{U}_{\bar{v}}$, control design proceeds as described in Section 30.4, by first optimizing $\mathbf{U}_0 \in \mathcal{U}_{\bar{v}}$, synthesizing $\mathbf{R}$ and $\mathbf{K}$ the same way, and defining $\mathcal{S}(\mathbf{x})$ using the actual (inhomogeneous) force feasibility domain $\mathcal{F}(\mathbf{v})$.

30.5.2 Dynamic Limitations

To illustrate a typical manner in which device dynamics manifest themselves, we return to the example of the variable-orifice damper illustrated in Figure 30.2.

Controllable damping devices suitable for regulating 1 MN of force at velocities of 1 m/s and pressures of 20 MPa at flow rates of 40 L/s can be made with proportional control valves having response times of 100 ms from commercial off-the-shelf components. A proportional- or servo-valve controls the flow of hydraulic fluid between chambers in the device, as shown in Figure 30.2. The piston has an area A_p, the pressure differential across the two chambers is $p_2 - p_1$, the device contains a volume of fluid V_T where $V_T = V_1 + V_2$. The diameter of the bore is D_b and the rod diameter is D_r.

Neglecting seal friction, the force, f, in the piston rod is $A_\mathrm{p}(p_2 - p_1)$ where A_p is the piston area. Assuming incompressible flow in the valve, $Q_1 = Q_2 = Q$, and approximating a linear pressure–flow relationship for the controllable valve, $(p_2 - p_1) = c(u)Q$, where $c(u)$ is the controllable pressure–flow coefficient and $u \in [0, 1]$ is the valve position. Considering fluid compressibility within the chambers 1 and 2, $\dot{p}_1 V_1 = -\beta \dot{V}_1$ and $\dot{p}_2 V_2 = -\beta \dot{V}_2$, where β is the bulk modulus of the hydraulic fluid which can range from 80 kN/cm^2 to 200 kN/cm^2. By conventions $\dot{V} > 0$ means volumetric expansion and $\dot{p} > 0$ means increasing hydrostatic compression. In chamber 1, $\dot{V}_1 = -Q_1 + A_\mathrm{p} v$ and in chamber 2, $\dot{V}_2 = Q_2 - A_\mathrm{p} v$. Substituting the valve equation, the equilibrium equation and the compressibility equations, Patten [18] obtained

$$\dot{p}_2 = -\frac{\beta}{V_2(x_\mathrm{p})} \frac{f}{A_\mathrm{p} c(u)} + \frac{\beta}{V_2(x_\mathrm{p})} A_\mathrm{p} v, \tag{30.55}$$

and

$$-\dot{p}_1 = -\frac{\beta}{V_1(x_\mathrm{p})} \frac{f}{A_\mathrm{p} c(u)} + \frac{\beta}{V_1(x_\mathrm{p})} A_\mathrm{p} v. \tag{30.56}$$

Adding and multiplying by A_p results in a model for the nonlinear damper dynamics

$$\dot{f} = A_p(\dot{p}_2 - \dot{p}_1) = -\frac{\beta}{c(u)}\left(\frac{1}{V_1(x_p)} + \frac{1}{V_2(x_p)}\right) f + \beta A_p^2 \left(\frac{1}{V_1(x_p)} + \frac{1}{V_2(x_p)}\right) v\,. \tag{30.57}$$

The valve opening variable, u, has a first order lag modeled by

$$\dot{u} = \frac{1}{T_u}(u^* - u) \tag{30.58}$$

where T_u is the valve time constant and u^* is the valve control input. Response times in many electromechanical systems may be shortened, within limits, by over-driving the valve. In such cases, the valve dynamics can be modeled by

$$\dot{u} = \frac{1}{T_u} \underset{[-1,1]}{\text{sat}} \left\{K_u(u^* - u)\right\} \tag{30.59}$$

where K_u is the valve gain. The valve coefficient $c(u)$ may be assumed to vary linearly with the valve variable u,

$$c(u) = (1 - u)c_{\min} + uc_{\max}. \tag{30.60}$$

The constants $c_{\min}$ and $c_{\max}$ and the constraints $0 \le u \le 1$ provide sector bounds on the damping force. The valve time lag overrides all other dynamics present in the nonlinear model and, as may be seen in Figure 30.6, the device behavior may be linearized to

$$f(v, u) = c(u)A_p^2 v. \tag{30.61}$$

Figure 30.6 is generated using the parameters values shown in the Table 30.2, a prescribed displacement of $x_p = t\sin(\pi t)$ cm, for $0 < t < 10$ s, and an *ad hoc* time-varying valve control input,

$$u^* = 50|x_p|^{3/2} \cdot \text{hvs}[-f \cdot x_p] \tag{30.62}$$

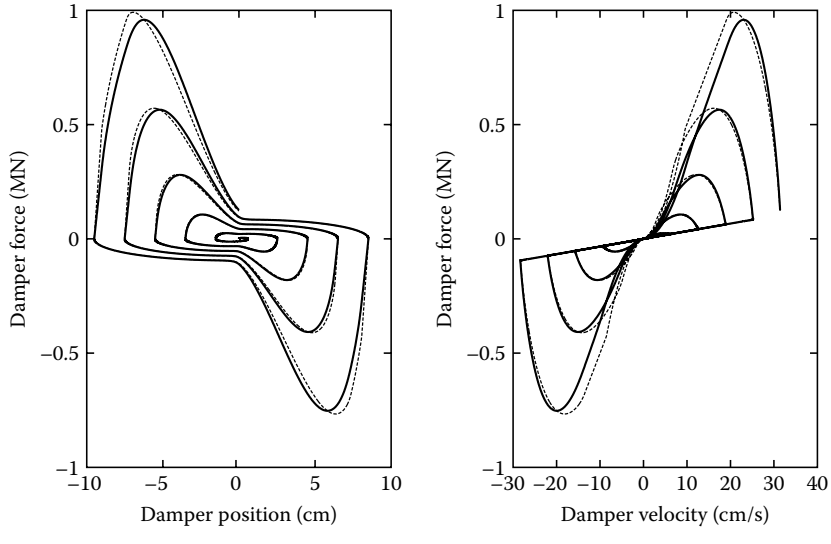

FIGURE 30.6 Comparison of the response of the nonlinear damper model (Equation 30.57, solid) with a linearized damper model (Equation 30.61, dashed) controlled according to Equations 30.59, 30.60, and 30.62 with parameters of Table 30.2.

TABLE 30.2 Controllable Damper Parameter Values

Hydraulic bulk modulus	β	100	kN/cm^2
Piston area	A_p	550	cm^2
Stroke	S	130	cm
Min. valve coefficient	c_{min}	110	Pa s/cm^3
Max. valve coefficient	c_{max}	1654	Pa s/cm^3
Valve time constant	T_u	0.1	s
Valve gain	K_u	10	—

where $\dot{x}_p = v$ in (m/s). The value of u^* is constrained to upper and lower bounds, thus resulting in a limitation to how fast u can be changed.

The presence of time lag T_u is difficult to incorporate precisely into the ideal bilinear semiactive control theory discussed in Section 30.4. However, these effects can be penalized in the formulation of control performance, by augmenting $\mathbf{z}$ to include $\mathbf{f}_h$, where $\hat{\mathbf{f}}_h(s) = W(s)\hat{\mathbf{f}}(s)$, and where $W(s)$ is a high-pass filter with corner frequency $1/T_u$.

Fluid pressures in this example damper are within the normal working range for commercial hydraulics (10–20 MPa). MR fluids may also prove to be promising for large-scale structural control applications, as demonstrated by the commercialization of a MR damper with a 6 MPa pressure rating and a 200 kN force capacity [19].

30.6 Example

As an example of the closed-loop performance of the semiactive control methods described in this chapter, the J_{quad}-bounded controller of Section 30.4.3 is applied to the structural model described in Figure 30.1. This structural model is based upon a large, laboratory-scale model that was developed for experimentation on passive base isolation systems. A mathematical model of this lab-scale structure has been used to simulate the behavior of semiactive damping in seismic isolation systems [20]. For this study, the mass, damping, and stiffness parameters of the laboratory-scale structure were scaled up by a factor of 70 in an attempt to represent a full-scale structural frame. The parameter values given in Figure 30.1 are the scaled-up values. The total mass of the scaled-up structure, W/g, is 2450 tons. The passive isolation force f_b is modeled by a parallel combination of elastic stiffness k_b, viscous damping c_b (given in Figure 30.1), and a hysteretic force,

$$f_b = k_b x_b + c_b \dot{x}_b + F_y(1-\kappa)z \tag{30.63}$$

where x_b is the deformation of the base isolation system and z is an evolutionary hysteretic variable

$$\dot{z} = \left[1 - |z|\left(0.5\,\mathrm{sgn}(z\dot{x}_b) + 0.5\right)\right]\dot{x}_b/D_y. \tag{30.64}$$

The parameters in this hysteresis model are the yield force F_y, the yield displacement D_y, and the strain hardening stiffness ratio, κ. The yield force is typically specified as a fraction of the structural weight, $F_y = F_{yr}W$. In this study the post-yield stiffness, k_b is fixed at 162.4 kN/cm, which results in a first-mode natural period of 2.5 s as $F_y \to 0$. The post-yield elastic stiffness is related to the pre-yield stiffness, F_y/D_y, by the stiffness ratio parameter, κ, as shown in Figure 30.7. Thus, $D_y = F_{yr}W\kappa/k_b$.

Controllable dampers modeled by Equations 30.59 and 30.61 are incorporated into the isolation system at the base and into the mass damper system on the roof. For the controllable damper in the base isolation system, $c_{min}A_p^2 = 333$ kN/cm/s, $c_{max}A_p^2 = 5000$ kN/cm/s, and $T_u = 0.1$ s as shown in Table 30.2

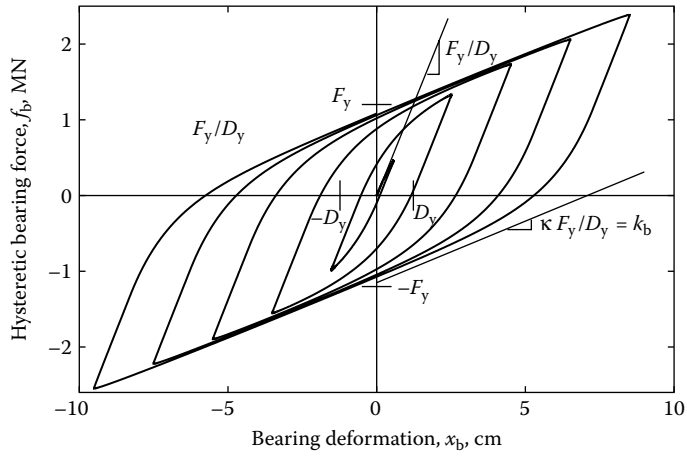

FIGURE 30.7 Hysteretic bearing force for $x_b = t\sin(\pi t)$ cm/s, $F_y = 0.05W$, $\kappa = 1/6$, $k_b = 162.4$ kN/cm, and $c_b = 2.618$ kN/cm/s.

and Figure 30.6. For the controllable damper in the mass damper system, $c_{\min}A_p^2 = 0.50$ kN/cm/s, $c_{\max}A_p^2 = 8.0$ kN/cm/s, and $T_u = 0.01$ s. For both dampers the dimensionless valve gain, K_u, is set to 10.

The earthquake disturbance is modeled with Equations 30.13 and 30.14 with the detrending method described in Section 30.2.2. The disturbance model parameters used in this example correspond to "far-field" earthquake sources. In both the passive and semiactive cases, each earthquake record was scaled to a PGV value of 0.33 m/s (the coefficient of variation of PGV was zero) and corresponds in a PGA of 0.4 g, with a coefficient of variation about 16%.

Performance is quantified in terms of four response metrics, the peak total roof acceleration,

$$\text{PRA} = \max_t |a(t) + \ddot{r}_5(t)|, \tag{30.65}$$

the peak total first floor acceleration,

$$\text{PFFA} = \max_t |a(t) + \ddot{r}_1(t)|, \tag{30.66}$$

the peak first story displacement,

$$\text{PFFD} = \max_t |r_2(t) - r_1(t)|, \tag{30.67}$$

and the peak isolation system displacement,

$$\text{PBD} = \max_t |r_1(t)|, \tag{30.68}$$

where $r_i(t)$ is the displacement of floor level i relative to the ground and $a(t)$ is the ground acceleration. The acceleration performance metrics indicate the potential for damage to building contents, the first story displacement indicates potential for structural damage, and the peak isolation system displacement indicates the potential for exceeding design limitations on the isolation bearings.

In order to rigorously assess the benefits of semiactive damping, it is critically important to first optimize a passive damping system; comparisons between semiactively controlled systems and lightly damped, uncontrolled systems are nugatory. Optimized passive damping systems can comprise linear viscosity, nonlinear viscosity, and hysteresis. For this study two passive damping optimizations are carried out. In the first optimization viscous damping is placed in parallel with a hysteretic isolation

system with $F_y = 0.05W$ and $\kappa = 1/6$. In the second optimization viscous damping is placed in parallel with a hysteretic isolation system with $F_y = 0.005W$ and $\kappa = 1/4$. Both optimizations are carried out with the far-field (FF) disturbance model. The passive damping systems were optimized to minimize the floor and roof responses without consideration on the peak base displacement, as the peak base displacement decreases monotonically with damping in the isolation system. It is typically found that in the vicinity of optimal damping, performance metrics are relatively insensitive to variations in the viscous damping rates. Such is the case in this study and the optimized values are therefore reported approximately in round numbers. For an isolation system with $F_y = 0.05W$ and $\kappa = 1/6$, $c_{b,opt} \approx 800$ kN/m/s and for an isolation system with $F_y = 0.005W$ and $\kappa = 1/4$, $c_{b,opt} \approx 1500$ kN/m/s. Performance metrics with optimized damping are lower for $F_y = 0.005W$ than for $F_y = 0.05W$ and this case is taken as the optimized passive damping system. The mass damper on the roof is tuned to the second mode of vibration. A damping value c_a of 2.8 kN/cm/s maximally suppresses the second mode.

Three cases are compared in this example. The baseline (BL) case is representative of the current state of the practice for seismic base isolation. The base isolation system has a yield force, F_y of $0.05W$ and a stiffness ratio κ of $1/6$. The viscous damping in this isolation system is 2.618 kN/cm/s and the tuned mass damper parameters are set to their optimized values. The optimal passive (OP) damping case represents performance limits that can be achieved via passive linear viscous damping with a small amount of hysteretic damping, $F_y = 0.005W$. The viscous damping in the isolation system and in the tuned mass damper system are set to their optimized values. For the semiactive damping case, the controller (Equation 30.21) with $\mathbf{K}$, $\mathbf{R}$, and $\mathcal{S}$ assigned according to the J_{quad}-bounded design equations is implemented using the following weightings: $z_1 = 10$ cm, for base displacement; $z_2 = 2$ mm, for first story drift; $z_3 = 0.1$ g, for first floor acceleration; and $z_4 = 0.1$ g, for roof acceleration.

In order to capture the effects of random variability in the earthquake disturbance, 250 transient response simulations are carried out using the FF ground motion parameters for each of the three cases. Performance metrics are presented in Table 30.3 in terms of their average values and their coefficients of variation. Figures 30.8 through 30.10 illustrate histograms of the performance metrics from the 250 simulations. Optimized passive damping reduces peak floor and roof responses, as intended, but these improvements come at the cost of increased base displacements. Semiactive damping provides reductions in peak responses for all performance metrics as compared to both the BL system and the optimized passive system. Responses in the semiactive system are reduced by 13% (at the roof) to 30% (in the isolation system) as compared to the optimized passive system. Of equal importance are the reductions in response variability. Responses in semiactively damped systems are lower and have less variability than optimally damped passive damping systems.

TABLE 30.3 Comparison of Baseline (BL), Optimized Passive (OP), and Semiactive (SA) Damping Performance

Metric		BL	OP	SA	SA-OP
avg PRA	g	0.126	0.098	0.085	−13%
cov PRA		0.136	0.240	0.158	−34%
avg PFFA	g	0.117	0.091	0.074	−19%
cov PFFA		0.139	0.251	0.164	−35%
avg PFFD	mm	0.910	0.773	0.631	−18%
cov PFFD		0.157	0.259	0.159	−39%
avg PBD	cm	9.454	12.088	8.433	−30%
cov PBD		0.269	0.285	0.200	−30%

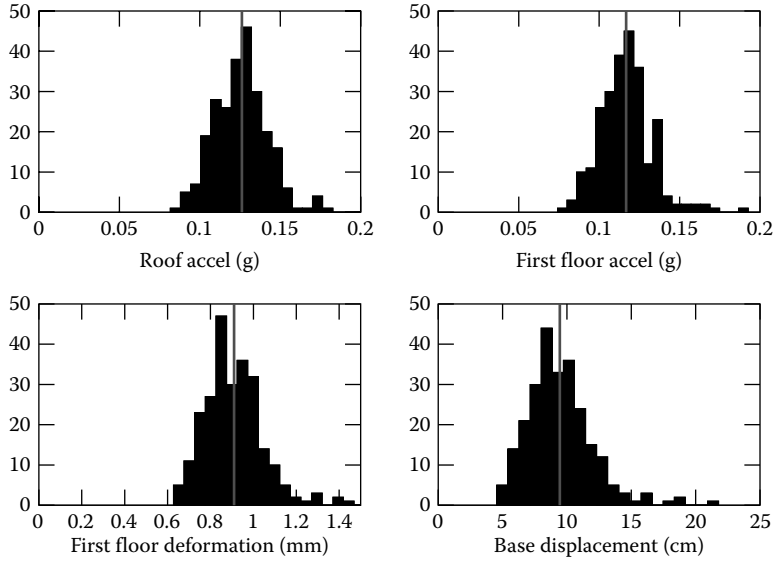

FIGURE 30.8 Response histograms for the baseline (BL) system: passive hysteretic damping.

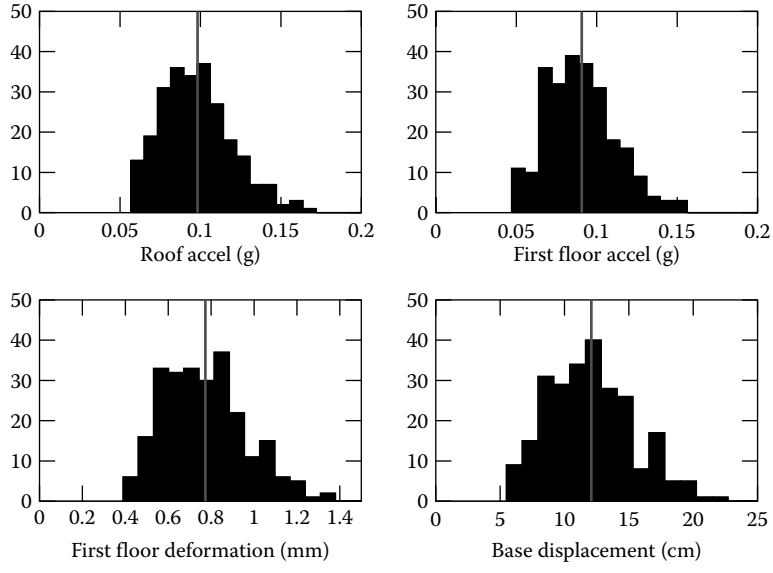

FIGURE 30.9 Response histograms for optimized passive (OP) viscous damping in the isolation system and the tuned mass damper.

30.7 Summary

Vibration control in civil engineering structures presents a number of challenges related to controller synthesis, actuator development and characterization, disturbance modeling and performance assessment. Issues related to power consumption in actuators and the need for robust stability in the presence of potentially significant modeling errors motivates the development of semiactive damping approaches in

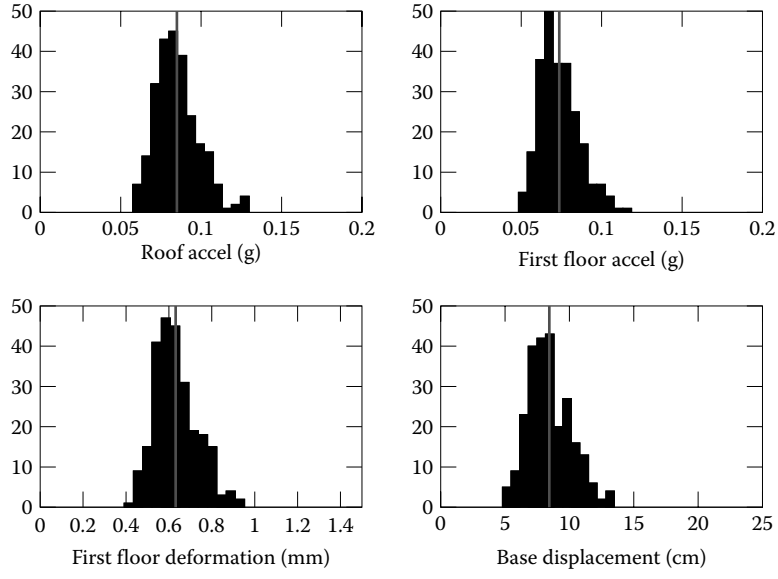

FIGURE 30.10 Response histograms for semiactive (SA) damping in the seismic isolation and the tuned mass damper.

which control forces are regulated by adjusting viscous damping coefficients. Semiactive devices capable of generating forces large enough for civil engineering applications can be assembled from conventional hydraulics. The forces produced by these devices are regulated through the adjustment of a valve. In so doing, Watts of electrical power can regulate Megawatts of mechanical power and the physical dissipative nature of the control devices guarantees closed-loop stability. These characteristics are naturally appealing but present significant challenges in the synthesis of control algorithms. Further, in order to fully describe the advantages of damping that is modulated via feedback it is critically important to assess the closed-loop performance in comparison to a system that has optimized levels of constant damping.

In controlling earthquake disturbances that are not strong enough to severely damage a structure it may be argued that a quadratic performance measure is appropriate. For potentially damaging earthquakes, however, minimization of peak responses are clearly of great interest. This chapter outlines iterative methods for optimizing feedback laws based on Lyapunov stability, on minimizing a quadratic objective, and on minimizing a peak response objective. These methodologies implicitly satisfy the instantaneous dissipativity constraint on the control hardware, and guarantee that the performance of the semiactive system outperforms the performance of the optimized static damping system. Analytical expressions for margin of performance improvement achieved with these controllers is still an open problem, as is a sound theory for the design of unbiased observers for semiactively-controlled systems. Until such observers are developed, the separation principle does not strictly hold, although it is often a reasonable approximation to assume it does.

Closed-loop semiactive systems are always nonlinear, and performance is assessed through transient response simulations. This chapter provides three earthquake disturbance models in terms of enveloped and filtered Gaussian white-noise processes. These disturbance models are helpful not only in the development of augmented dynamics matrices for controller synthesis but also in the assessment of the probability distributions of performance metrics.

We have illustrated many of these concepts for an earthquake-excited, base-isolated structure with a tuned-mass damper. Semiactive devices are located within the isolation system at the base and in the tuned mass damper on the roof. The earthquake disturbance model was presumed to possess far-field

characteristics, and a quadratic performance measure was used as an optimization objective. Response metrics for the semiactively-controlled system are lower and have less variability than those for the optimal constant damping systems. This result is characteristic of systems in which the disturbance is sufficiently persistent. Further, this quadratic controller is found to be more effective in suppressing *peak* responses than the optimized passive damping system. For near-fault earthquake models, which can exhibit pulse phenomena and which are highly nonstationary, control synthesis techniques which explicitly guarantee bounds on peak responses, such as those in Sections 30.4.1 and 30.4.2, may provide better closed-loop performance.

References

1. T.T. Soong, *Active Structural Control: Theory and Practice*, Addison-Wesley, Reading, MA, 1990.
2. D. Hrovat, Survey of advanced suspension developments and related optimal control applications, *Automatica*, vol. 33(10), pp. 1781–1817, 1997.
3. G.W. Housner, L.A. Bergman, T.K. Caughey, A.G. Cassiakos, R.O. Claus, S.F. Masri, R.E. Skelton, T.T. Soong, B.F. Spencer Jr., and J.T.P. Yao, Structural control: past, present and future, *Journal of Engineering Mechanics*, vol. 123(9), pp. 897–971, 1997.
4. B.F. Spencer Jr. and S. Nagarajaiah, State of the art in structural control, *ASCE Journal of Structural Engineering*, vol. 129(7), pp. 845–856, 2003.
5. M.D. Symans and M.C. Constantinou, Semiactive control systems for seismic protection of structures: A state-of-the-art review, *Engineering Structures*, vol. 21(6), pp. 469–487, 1999.
6. D.M. Boore, Simulation of ground motion using the stochastic method, *Pure and Applied Geophysics*, vol. 160, pp. 635–676, 2003.
7. FEMA, Quantification of building seismic performance factors ATC-63 Project Report—90% Draft, FEMA report P695, April 2008.
8. G. Leitmann, Semiactive control for vibration attenuation, *Journal of Intelligent Material Systems and Structures*, vol. 5, pp. 841–846, 1994.
9. G. Leitmann and E. Reithmeier, A control scheme based on ER-materials for vibration attenuation of dynamical systems, *Applied Mathematics and Computation*, vol. 70, pp. 247–259, 1995.
10. S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.
11. D. Karnopp, M.M. Crosby, and R.A. Harwood, Vibration control using semi-active force generators, *ASME Journal of Engineering for Industry*, vol. 96, pp. 619–626, 1974.
12. D.C. Karnopp, Active damping in road vehicle suspension system, *Vehicle System Dynamics*, vol. 12, pp. 291–316, 1983.
13. D.L. Margolis, The response of active and semi-active suspensions to realistic feedback signals, *Vehicle System Dynamics*, vol. 12, pp. 317–330, 1983.
14. S.J. Dyke, B.F. Spencer Jr., M.K. Sain, and J.D. Carlson, Modeling and control of magnetorheological dampers for seismic response reduction, *Smart Materials and Structures*, vol. 5(5), pp. 565–575, 1996.
15. J.T. Scruggs, A.A. Taflanidis, and W.D. Iwan, Nonlinear stochastic controllers for semiactive and regenerative systems yielding guaranteed quadratic performance bounds. Part 1: State feedback control, *Journal of Structural Control and Health Monitoring*, vol. 14, pp. 1101–1120, 2007.
16. Z.G. Ying, W.Q. Zhu, and T.T. Soong, A stochastic optimal semi-active control strategy for ER/MR dampers, *Journal of Sound and Vibration*, vol. 259, pp. 45–62, 2003.
17. H.E. Tseng and J.K. Hedrick, Semi-active control laws—optimal and sub-optimal, *Vehicle System Dynamics*, vol. 23, pp. 545–569, 1994.
18. W.N. Patten, C. Mo, J. Kuehn, and J. Lee, A primer on design of semiactive vibration absorbers (SAVA), *Journal of Engineering Mechanics*, vol. 124, no. 1, 61–68, 1998.
19. G. Yang, B.F. Spencer Jr., J.D. Carlson and M.K. Sain, Large-scale MR fluid dampers: modeling and dynamic performance considerations, *Engineering Structures*, vol. 24, no. 3, 309–323, 2002.
20. J.C. Ramallo, E.A. Johnson, and B.F. Spencer, Smart base isolation systems, *Journal of Engineering Mechanics*, vol. 128, pp. 1088–1099, 2002.

31

Quantum Estimation and Control

Matthew R. James
Australian National University

Robert L. Kosut
SC Solutions

31.1 Introduction

"I would like to describe a field, in which little has been done, but in which an enormous amount can be done in principle. This field is not quite the same as the others in that it will not tell us much of fundamental physics (in the sense of, "What are the strange particles?") but it is more like solid-state physics in the sense that it might tell us much of great interest about the strange phenomena that occur in complex situations. Furthermore, a point that is most important is that it would have an enormous number of technical applications.

What I want to talk about is the problem of manipulating and controlling things on a small scale.

As soon as I mention this, people tell me about miniaturization, and how far it has progressed today. They tell me about electric motors that are the size of the nail on your small finger. There is a device on the market, they tell me, with which you can write the Lord's Prayer on the head of a pin. But that is nothing; that is the most primitive halting step in the direction I intend to discuss. It is a staggeringly small world that is presented and described below. In the year 2000,

> when they look back at this age, they will wonder why it was not until the year 1960 that anybody began to move seriously in this direction.
>
> *Why cannot we write the entire 24 volumes of the Encyclopedia Brittanica on the head of a pin?*"
>
> **Richard P. Feynman**
> *There's Plenty of Room at the Bottom, American Physical Society Caltech,*
> Dec. 29, 1959

There is equal "wonder" looking back from this end of the time tunnel that Feynman was able to herald the promise of quantum technology so many years ago. In our quest for a deeper understanding of physical and biological phenomena, only now, perhaps, we move seriously into the control of the "small-scale" world of quantum mechanics. The rules of this world herald new types of materials and devices [1–3]. Quantum information systems and instruments of measurement promise an exponential improvement in speed and/or resolution compared to their classical counterparts. Many of these systems inherently rely on estimation and control for their normal operation, for example, atomic clocks, measuring electrical, thermal, and photonic characteristics, biometrics, magnetometry, gravimetry, and the many proposed implementations of a quantum computer. Instrumentation noise, decoherence, and modeling errors are all sources of uncertainty which either separately or in combination hinder the ability of the material or device to meet performance demands. Some of these systems require estimation to determine if the system is meeting performance demands and then apply a control adapted to the specific estimated error, for example [4–8].

Our aim here is to show how estimation and control can be applied to quantum systems. Furthermore, we believe that these tools from control engineering, as appropriately modified and or developed, are the means by which many hopes and dreams envisioned for a quantum technology can come about.

31.1.1 Quantum Estimation and Control

"To observe is to disturb" goes the dictum. So, how can one estimate *anything* in quantum mechanics? The answer is that although the system behaves according to rules of probability, these rules are, well, rules! So the probability of outcomes are the laws. Hence, statistical approaches to *estimation*, such as maximum likelihood (ML), least-squares, filtering, and so on, do apply. Furthermore, the fundamental statistical behavior of quantum systems can be *controlled* using *feedback*.

In *The Human Use of Human Beings: Cybernetics and Society* (1950), Norbert Wiener introduces feedback control in this way:

> "This control of a machine on the basis of its actual performance, rather than its expected performance is known as *feedback* . . . It is the function of control . . . to produce a temporary and local reversal of the normal direction of entropy."

The classic classroom example of feedback control is the mechanical governor used by James Watt in the eighteenth century to regulate the speed of his steam engines (Figure 31.1).

The actual engine speed raises the balls by centrifugal force. As these rise, the linkages are arranged to close down the intake valve. The speed is maintained in the neighborhood of an equilibrium. Feedback control was essential for the stable operation of steam engines, and was thus a critical enabling technology for these machines which powered the industrial revolution. A precise analysis was not made until the mid-1800s when Clerk Maxwell put his mind to it.

What is it that is so compelling about this apparatus? Firstly, it is easy to understand how it regulates the speed of a rotating steam engine. Secondly, and perhaps more importantly, *it is a part of the device itself*. A naive observer would not distinguish this mechanical piece from the rest. And this device, if we think of it having a thought, almost needs no thoughts at all! It need not know any real detail about the object it controls; no knowledge of steam, pressure, flow, friction, metal fatigue, anchor bolt placement, and so on. Almost nothing. Yet, it is the fundamental piece without which the steam engine might explode. Due

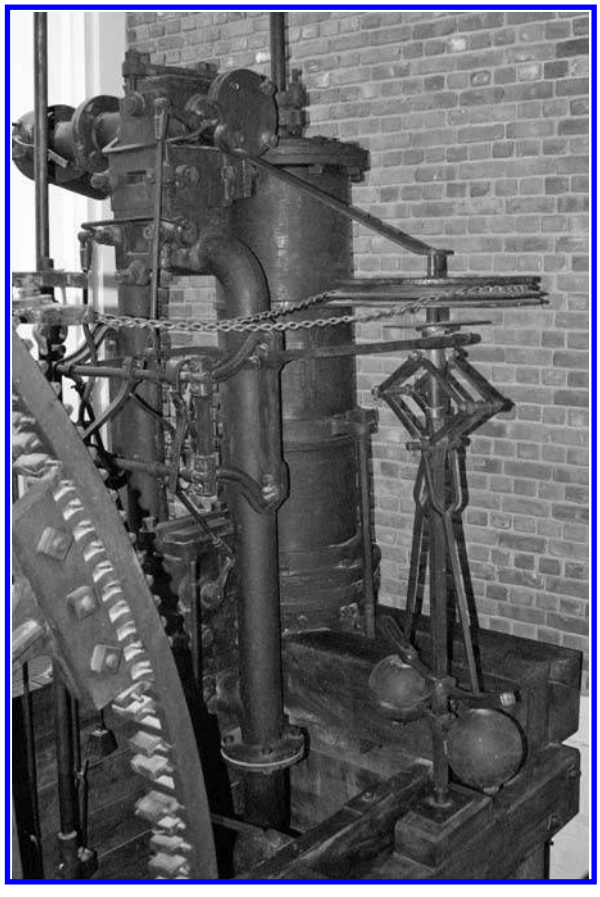

FIGURE 31.1 Boulton and Watt steam engine, 1788, showing the mechanical governor (metal ball mechanism) [located at London Science Museum].

to its seeming simplicity, the notion of feedback takes on a mysterious property. It is both intangible, and yet, fundamental to the stability of the device, because it responds to the effect of the actual rotational speed. Without this simultaneously intangible *and real* feedback, the device would not exist!

Steam engines, are of course macroscopic systems described by classical physics, and control engineering has been founded on classical models. At this point in time, it is beginning to be possible to monitor and manipulate objects at the nanoscale. One can realistically contemplate controlling single atoms (Figure 31.2).

At the atomic scale, the laws of quantum physics are needed, and in fact provide a significant new resource for technological exploitation, as can be seen in recent advances in quantum information and computing, precise metrology, atom lasers, quantum electromechanical systems, and quantum chemistry. *Quantum control* refers to the control of physical systems whose behavior is dominated by the laws of quantum physics, and control theory is being developed that takes into account quantum physics (e.g., [9–41]).

What types of quantum control can be envisaged? As for classical (i.e., nonquantum) systems, we distinguish between open- and closed-loop control. Open-loop control has its usual meaning—a predetermined classical control signal is applied to the plant, in this case a quantum system, and no feedback is involved (Figure 31.3).

Closed-loop or feedback control also has its usual meaning—control actions depend on information gained while the plant is operating—however, care must be taken as to the nature of the controller

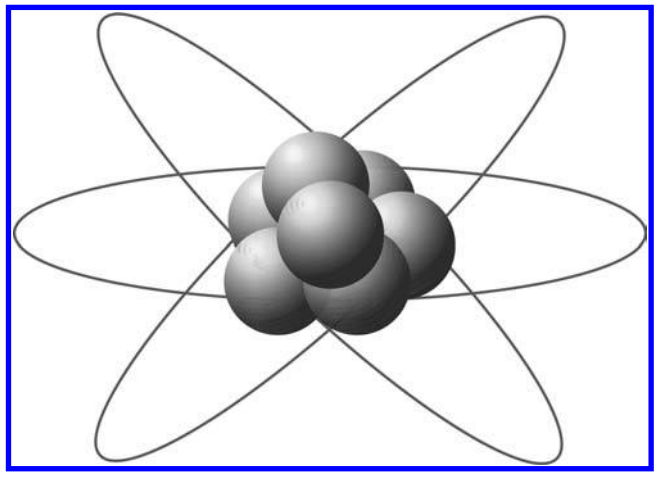

FIGURE 31.2 Model of an atom.

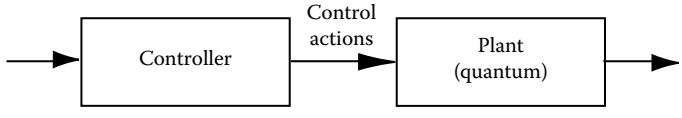

FIGURE 31.3 Open-loop control.

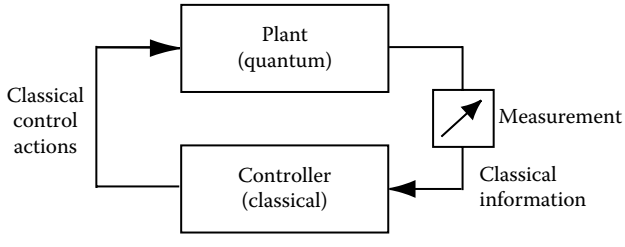

FIGURE 31.4 Closed-loop measurement feedback control.

and what is meant by "information." When the controller is a classical system, which can only process classical information, some form of measurement of the quantum plant is needed, see Figure 31.4. This is called *measurement feedback* quantum control. The theory and applications that have been developed for measurement feedback depend on quantum filtering theory [42,43], as we explain in Sections 31.5.2 through 31.5.4. Measurement feedback quantum control can be effective for a wide range of applications, and has the benefit that the control algorithms can be implemented in conventional classical hardware (provided it is fast enough).

It is also possible to use another quantum system as the controller, see Figure 31.5. This type of feedback *does not use measurement*, and the information flowing in the loop is fully quantum. This exchange of quantum information may be directional, via a *quantum signal* (such as a beam of light), or bidirectional, via a direct physical coupling. This is called *coherent* or *quantum feedback* quantum control. While quantum feedback is conceptually simple, at present little is known about how to systematically *design* fully (CF) coherent feedback loops. In Section 31.5.5, we describe one recent example of coherent feedback design. The benefits of coherent feedback include the preservation of quantum information, and that the timescales of the controller can be better matched to the quantum plant (which could have very fast dynamics).

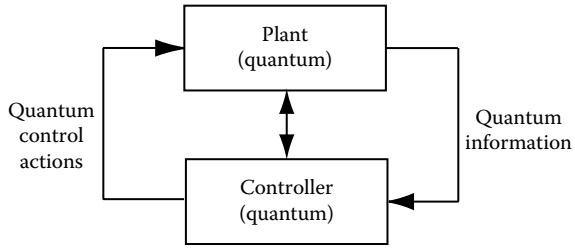

FIGURE 31.5 Closed-loop feedback control with no measurement.

31.2 Some Quantum Mechanics

31.2.1 Preliminaries

Let $\mathcal{H}$ be a separable Hilbert space with inner product $\langle \cdot, \cdot \rangle$ (taken to be linear in the second argument and conjugate linear in the first) and norm $\|\psi\| = \sqrt{\langle \psi, \psi \rangle}$. Basic examples are (i) $\mathcal{H} = \mathbb{C}^n$, the space of n-dimensional complex vectors, where $\langle \psi, \phi \rangle = \sum_{k=1}^{n} \psi_k^\dagger \phi_k$, where $\psi_k^\dagger$ denotes the adjoint (complex conjugate) of ψ_k, and (ii) $\mathcal{H} = L^2(\mathbb{R})$, the space of complex-valued functions on $\mathbb{R}$ that have square integrable components, with inner product $\langle \psi, \phi \rangle = \int \psi^\dagger(x)\phi(x)\, dx$.

Let $\mathscr{B}(\mathcal{H})$ be the Banach space of *bounded operators* $A : \mathcal{H} \to \mathcal{H}$. The commutator of two operators is defined by $[A, B] = AB - BA$. For any $A \in \mathscr{B}(\mathcal{H})$, its *adjoint* $A^\dagger \in \mathscr{B}(\mathcal{H})$ is an operator defined by $\langle A^\dagger \psi, \phi \rangle = \langle \psi, A\phi \rangle$ for all $\psi, \phi \in \mathcal{H}$. An operator $A \in \mathscr{B}(\mathcal{H})$ is called *normal* if $AA^\dagger = A^\dagger A$. Two important types of normal operators are *self-adjoint* ($A = A^\dagger$) and *unitary* ($A^\dagger = A^{-1}$). The *spectral theorem* for a normal operator A says that (discrete case) there exists a complete set of orthonormal eigenvectors (such a set forms a basis for $\mathcal{H}$), and A can be written as $A = \sum_n a_n P_n$, where P_n is the projection onto the nth eigenspace (diagonal representation) with associated eigenvalue a_n. In Dirac's bra-ket notation, the eigenvectors are written $|n\rangle$ and the projections $P_n = |n\rangle\langle n|$. The projections resolve the identity (orthogonally): $\sum_n P_n = I$. If A is self-adjoint, the eigenvalues a_n are all real. In this notation, most often written by physicists, the "ket" $|\psi\rangle$ is always a unit vector with adjoint $\langle\psi|$. The norm of $|\psi\rangle$ is thus $\|\psi\| = \sqrt{\langle\psi|\psi\rangle} = 1$. (We will not always adhere to the ket-notation, and so sometimes write ψ and implicitly assume that it is a unit vector, i.e., $\psi^\dagger\psi = 1$.)

Tensor products are used to describe *composite systems*. If $\mathcal{H}_1$ and $\mathcal{H}_2$ are Hilbert spaces, the *tensor product* $\mathcal{H}_1 \otimes \mathcal{H}_2$ is the Hilbert space consisting of linear combinations of the form $\psi_1 \otimes \psi_2$, and inner product $\langle \psi_1 \otimes \psi_2, \phi_1 \otimes \phi_2 \rangle = \langle \psi_1, \phi_1 \rangle \langle \psi_2, \phi_2 \rangle$. Here, $\psi_1, \phi_1 \in \mathcal{H}_1$ and $\psi_2, \phi_2 \in \mathcal{H}_2$. If A_1 and A_2 are operators on $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, then $A_1 \otimes A_2$ is an operator on $\mathcal{H}_1 \otimes \mathcal{H}_2$ and is defined by $(A_1 \otimes A_2)(\psi_1 \otimes \psi_2) = A_1\psi_1 \otimes A_2\psi_2$. Often, $A_1 \otimes A_2$ is written A_1A_2.

31.2.2 The Postulates of Quantum Mechanics

In quantum mechanics [44] physical quantities like energy, spin, position, and so on, are expressed as *observables*; these are represented as self-adjoint operators ($A = A^\dagger$) acting on a Hilbert space $\mathcal{H}$.

The *state* of a quantum system is a unit vector $\psi \in \mathcal{H}$ or $|\psi\rangle \in \mathcal{H}$. In the discrete case every element ψ_k is a possible state of the system with a probability of occurence $|\psi_k|^2$. Hence, $\|\psi\| = 1$ means that all outcomes can occur. (The same applies in the continuous case, for example, at a spatial point $r = (x, y, z)$, $\|\psi(r)\|^2 dxdydz$ is the probability that a particle would be found in the differential volume.) The state $|\psi\rangle$ is called a *pure state*. Pure states are special cases of a more general notion of state referred to as a *density operator* or *density matrix*. A density operator ρ is a positive self-adjoint operator on $\mathcal{H}$ with trace one. Pure states are of the form $\rho = |\psi\rangle\langle\psi|$. More generally, states that are convex combinations of pure states are called *mixed states*: $\rho = \sum_n \lambda_n |\psi_n\rangle\langle\psi_n|$.

The postulates of quantum mechanics state that for a *closed* system the evolution of states obeys the Schrödinger equation

$$i\frac{d}{dt}|\psi\rangle = H|\psi\rangle. \tag{31.1}$$

Here, H is an observable called the *Hamiltonian*, and represents the energy of the system, and $i = \sqrt{-1}$. Since $|\psi(t)\rangle$ has unit norm, the evolution from one time to another must be unitary, that is, $\psi(t) = U(t)\psi(0)$ where the unitary transition matrix, referred to as the *propagator*, obeys a matrix version of the Schrödinger equation, $i\dot{U} = HU,\ U(0) = I$ (the identity). Density operators also evolve unitarily, $\rho(t) = U(t)\rho U^{\dagger}(t)$, so, by the Schrödinger equation (31.1) we have $i\dot{\rho} = [H, \rho] = H\rho - \rho H$. We may view state vectors as fixed in time, while observables are taken to evolve according to $A(t) = U^{\dagger}(t)AU(t)$: this is the *Heisenberg picture*.

The numerical value of a measurement of A is an eigenvalue of A. If the system is in state ρ at the time of the measurement, and A has the spectral decomposition $A = \sum_n a_n P_n$, the value a_n occurs with probability

$$\text{Prob}(a_n) = \text{Tr}[\rho P_n]. \tag{31.2}$$

After the measurement, if the value a_n is recorded, the state "collapses" to

$$\rho' = \frac{P_n \rho P_n}{\text{Prob}(a_n)}. \tag{31.3}$$

This is Von Neumann's state reduction. When a quantum system is in a pure state $|\psi\rangle$, the expected value of an observable A is defined in terms of the Hilbert space inner product: $\langle A\rangle = \langle\psi, A\psi\rangle$. Using the spectral decomposition of A gives $\langle A\rangle = \sum_n \lambda_n \langle\psi, P_n\psi\rangle$. If the system is in the mixed state ρ, then $\langle A\rangle = \text{Tr}(A\rho) = \sum_n \lambda_n \text{Tr}(P_n\rho)$. (We will use the notations $\langle A\rangle$ or $\mathbb{P}[A]$ for the expected value of an observable A.)

In a more abstract mathematical sense, if $\mathscr{C}$ is a commutative collection of operators (a commutative $*$-algebra), then by the *spectral theorem* [13, Theorem 2.4] a density operator ρ determines a classical probability distribution $\mathbf{P}$ and for all $C \in \mathscr{C}$ a classical random variable $\iota(C)$ on a classical probability space constructed from the spectrum of $\mathscr{C}$ such that $\mathbb{P}[C] = \mathbf{P}[\iota(C)]$. When the context is clear we may abuse the notation and simply write C for both the observable $C \in \mathscr{C}$ or the corresponding classical random variable $\iota(C)$. In case of the postulate stated above for an observable A, the projections P_n generate such a commutative collection $\mathscr{C}$.

31.2.3 Open Quantum Systems

Open quantum systems are quantum systems that form part of a larger closed system. Figure 31.6 illustrates a representation of a system S which is "open" to the environment E. The environment consists

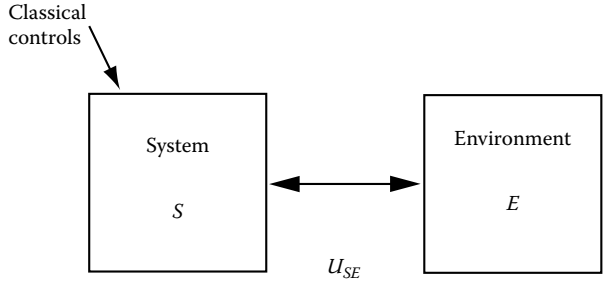

FIGURE 31.6 Representation of an open quantum system.

of an inaccessible part of the whole system, for example, a heat bath, nuclear spins, phonons, and so on. The complete *SE* system (a composite system) obeys the normal evolutionary dynamics of quantum mechanics as given by a unitary U_{SE}, which may depend on externally applied classical controls.

Here the state of the *S*-system is accessible, while the state of the *E*-system is not accessible. We will refer to the *S*-system as the *system* and the the *E*-system as the *environment* or *bath*. Since not every state of the universe is accessible, it is of basic interest to describe the potentially nonunitary transformation of the system state from one time to another.

In general, the state-to-state dynamics of any *open* quantum system can be described in a canonical form known as the *Kraus operator sum representation* (OSR) [45]. This formulation can account for many forms of error sources as well as decoherence. Let ρ_{in}^S denote the system state at some initial time and ρ_{out}^S at a later time. If the input states $\rho_{\text{in}}^S \in \mathbf{C}^{n_S \times n_S}$ and $\rho^E \in \mathbf{C}^{n_E \times n_E}$ are uncorrelated, that is, they form a tensor product input to U_{SE}, then the state at the *S*-system output is given by the Kraus OSR,

$$\rho_{\text{out}}^S = \sum_{\mu=1}^{\kappa} K_\mu \rho_{\text{in}}^S K_\mu^\dagger, \quad \sum_{\mu=1}^{\kappa} K_\mu^\dagger K_\mu = I_S. \tag{31.4}$$

The $K_\mu \in \mathbf{C}^{n_S \times n_S}$, called OSR elements, as constrained above, ensure that the quantum system is trace-preserving, that is, $\text{Tr}\rho_{\text{in}}^S = 1$ implies $\text{Tr}\rho_{\text{out}}^S = 1$. Additionally, all the quantum statistics produced by a measurement on the *S*-system are captured by the OSR. Specifically, for any observable $A = \sum_n a_n P_n$ on the *S*-system,

$$\text{Prob}(a_n) = \text{Tr}(P_n \rho_{\text{out}}^S) = \sum_\mu \text{Tr}(P_n K_\mu \rho_{\text{in}}^S K_\mu^\dagger). \tag{31.5}$$

The output state is obtained by "tracing out" the environmental states, referred to as the *partial trace operation*,* denoted by

$$\rho_{\text{out}}^S = \text{Tr}_E[U_{SE}(\rho_{\text{in}}^S \otimes \rho^E)U_{SE}^\dagger]. \tag{31.6}$$

The output state of the open system is thus a combination of the unitary dynamics U_{SE} and the average influence of the *E*-system on the *S*-system [45].

However, in other situations, the state of the *S*-system is not accessible, and the *E*-system, or parts of the *E*-system, are accessible, and available for feedback control. For instance, in Figure 31.7 the *S*-system is an atom, while the *E*-system is the external freely traveling field. The atom cannot be directly measured. Instead, an observable $y_0(t)$ of the field is measured, and this information can be processed classically and used in measurement feedback control, Section 31.5.3. Alternatively, the field need not be measured and instead may be processed coherently by another quantum system, as in coherent feedback control, Section 31.5.5.

31.2.4 Convexity and Quantum Mechanics

Convexity arises very naturally in quantum mechanics and plays an important role particularly in quantum estimation where a number of problems can be formulated as a convex optimization. Consider, for example, the following convex sets which arise from some of the basic aspects of quantum mechanics in a Hilbert space of dimension *n*:

* If ρ is a state on the composite system $S \otimes E$, then $\rho_1 = \text{Tr}_E[\rho]$ is a state on *S* such that for all *S*-system observables *X*, $\text{Tr}[\rho_1 X] = \text{Tr}[\rho(X \otimes I)]$.

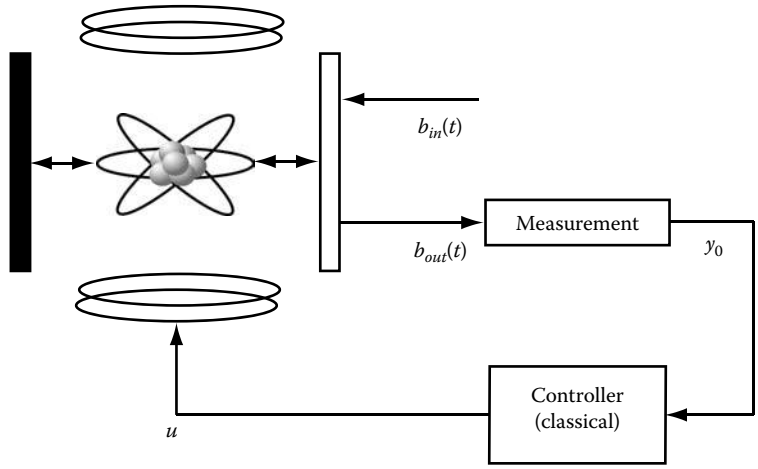

FIGURE 31.7 Atom in cavity feedback control. In the notation of Section 31.2.3, the atom is the *S*-system, while the fields $b_{in}(t)$ and $b_{out}(t)$ constitute the *E*-system. Here, an observable $y_0(t)$ of the *E*-system is measured and used in the feedback loop.

Probability outcomes	$\{p_\alpha \in \mathbf{R}\}$	$\sum_\alpha p_\alpha = 1, \quad p_\alpha \geq 0$
Density matrix	$\{\rho \in \mathbf{C}^{n\times n}\}$	$\mathrm{Tr}\ \rho = 1, \quad \rho \geq 0$
Positive operator valued measure (POVM)	$\{O_\alpha \in \mathbf{C}^{n\times n}\}$	$\sum_\alpha O_\alpha = I_n, \quad O_\alpha \geq 0$
OSR in a fixed basis set $\{ B_i \in \mathbf{C}^{n\times n} \mid i = 1, \ldots, n^2 \}$	$\{X \in \mathbf{C}^{n^2\times n^2}\}$	$\sum_{ij} X_{ij} B_i^\dagger B_j = I_n, \quad X \geq 0$

31.2.5 The Harmonic Oscillator

The quantum harmonic oscillator is one of the most important examples because of its tractability and application in modeling [46, Box 7.2], [44, Sec. 10.6], [47, Sec. 4.1]. Models for the optical cavity and boson fields are based on the quantum harmonic oscillator. The Hilbert space for the quantum harmonic oscillator is $\mathcal{H} = L^2(\mathbf{R}, \mathbf{C})$, the vector space of square integrable functions defined on the real line. Operators for this system may be expressed in terms of the annihilation operator a, with $a^\dagger$ the adjoint of a, and the *canonical commutation relations* $[a, a^\dagger] = 1$. The action of the annihilation operator may be expressed as

$$(a\psi)(x) = x\psi(x) - i\frac{d\psi}{dx}(x)$$

on a domain of functions (vectors) ψ in $\mathcal{H}$. The eigenvalues of $a^\dagger a$ are the numbers $0, 1, 2, \ldots$ (number of quanta), with corresponding eigenvectors denoted ψ_n ($n = 0, 1, 2, \ldots$) called *number states*. We have $a\psi_n = \sqrt{n}\,\psi_{n-1}$ and $a^\dagger\psi_n = \sqrt{n+1}\,\psi_{n+1}$.

If the harmonic oscillator has Hamiltonian $H = \omega a^\dagger a$, the evolution of the annihilation operators is defined by $a(t) = U^\dagger(t)aU(t)$, where $U(t)$ is the unitary operator (or matrix, depending on the context) solving Schrodinger's Equation 31.1, that is, $\dot{a}(t) = -i[a(t), H] = -i\omega a(t)$, with initial condition $a(0) = a$. Thus, $a(t) = e^{-i\omega t}a$, and from this we see explicitly that the commutation relations are preserved: $[a(t), a^\dagger(t)] = [a, a^\dagger] = 1$ for all t.

The real and imaginary *quadratures* are the self-adjoint operators $q = \frac{1}{\sqrt{2}}(a + a^\dagger)$ and $p = -\frac{i}{\sqrt{2}}(a - a^\dagger)$, respectively. We may form the vector $x = (q, p)^T$, so that in quadrature form the oscillator dynamics are given by $\dot{x}(t) = Ax(t)$, explicitly

$$\begin{bmatrix} \dot{q}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \begin{bmatrix} q(t) \\ p(t) \end{bmatrix}. \tag{31.7}$$

Notice that the matrix A appearing in Equation 31.7 has a special form. The commutation relations are $[q, p] = i$, which may be expressed in terms of the vector x as follows: $[x_j, x_k] = iJ_{jk}$, where

$$J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

The matrix A satisfies $AJ + JA^T = 0$, and this relation is enough to determine the Hamiltonian H: we have $H = \frac{1}{2}x^T Rx$, where $R = \frac{1}{2}(-JA + A^T J)$. As we shall see below, algebraic relations play a fundamental role in characterizing quantum systems and may be exploited for physical realizations [25,31].

The harmonic oscillator forms the building blocks for the quantum linear systems discussed in this chapter (Section 31.5.1). We will always consider Gaussian states, which means that the probability distributions of all quadratures are classical Gaussian measures.

31.2.6 Boson Fields

In quantum mechanics, an electromagnetic field, such as a beam of light, is described as a boson field. In the systems to be discussed in Sections 31.2.7 and 31.5, we use quantum stochastic models which arise in rotating wave and Markovian approximations of more basic models. This idealization affords considerable transparency and tractability, and provides accurate descriptions of a wide range of situations in quantum optics.

A boson field channel is represented by an infinite collection of oscillators with singular commutation relations. In the time domain, we have annihilation operators $b(t)$, which satisfy $[b(t), b^\dagger(s)] = \delta(t - s)$. When the field is in the vacuum state, the covariance is $\langle b(t)b^\dagger(t')\rangle = \delta(t - t')$. Real and imaginary field quadratures are defined by $b_r(t) = b(t) + b^\dagger(t)$ and $b_i(t) = -i(b(t) - b^\dagger(t))$, respectively (analogous to phasor representations in AC circuit analysis). When the field is in the vacuum state, the two quadratures are each equivalent to classical Wiener processes, but they do not commute. In quadrature form, the covariances are captured in the |non-negative| Hermitian matrix F defined by

$$F = \langle \begin{bmatrix} b_r(t) \\ b_i(t) \end{bmatrix} \begin{bmatrix} b_r(s) & b_i(s) \end{bmatrix} \rangle = (I + iJ)\delta(t - s). \tag{31.8}$$

31.2.7 Optical Cavity

The cavity is a basic element in quantum optical systems (Figure 31.8a). In Figure 31.8b, a schematic representation of a cavity is shown consisting of a pair of mirrors between which a trapped electromagnetic (optical) mode is set up, whose frequency depends on the separation between the mirrors. This mode is described by a harmonic oscillator with annihilation operator a (as in Section 31.2.5). The partially transmitting mirror affords the opportunity for this mode to interact with an external free field B. When the external field is in the vacuum state, energy initially inside the cavity mode may leak out, in which case the cavity system is a damped harmonic oscillator [47].

The cavity is an example of an open quantum system, Section 31.2.3, where the S-system is the internal cavity mode, and the E-system is the external freely travelling field. The Schrodinger equation for the

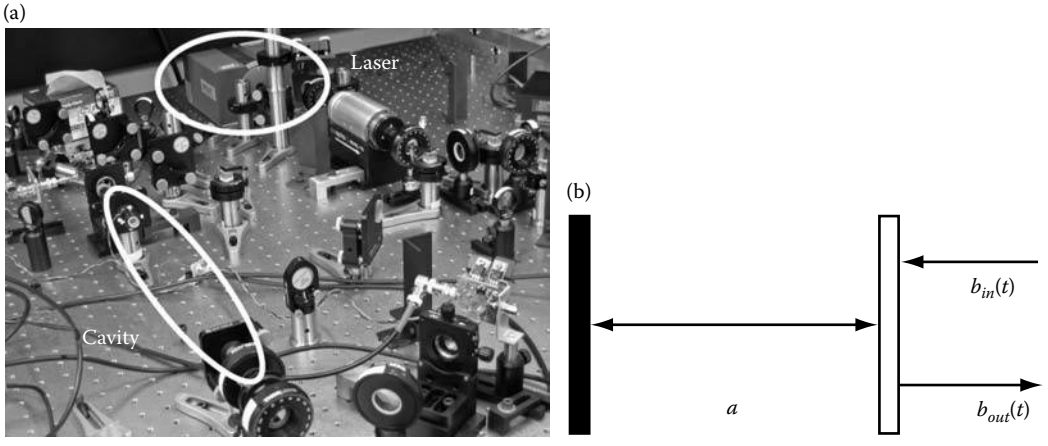

FIGURE 31.8 (a) A cavity constructed from a pair of mirrors in a quantum optical system (courtesy by E. Huntington). (b) A simplified representation of a cavity consisting of two mirrors, one of which is perfectly reflecting while the other is partially transmitting (shown unfilled). The partially transmitting mirror enables the light mode a inside the cavity to interact with an external light field, such as a laser beam. The external field is separated into input $b_{in}(t)$ and output $b_{out}(t)$ components by a Faraday isolator (not shown).

cavity is, in Stratonovich form,

$$\dot{U}(t) = \{\sqrt{\gamma}\, a b_{in}^{\dagger}(t) - \sqrt{\gamma}\, a^{\dagger} b_{in}(t) - i\omega a^{\dagger} a\} U(t), \quad U(0) = I. \tag{31.9}$$

Here, γ is a measure of the strength of the coupling of the cavity mode to the external field, and ω is a frequency parameter corresponding to the level of detuning between the cavity and the external field. The operator $L = \sqrt{\gamma}\, a$ is called the coupling operator. The cavity mode evolves according to $a(t) = U^{\dagger}(t) a U(t)$, so that

$$\dot{a}(t) = -(\frac{\gamma}{2} + i\omega) a(t) - \sqrt{\gamma}\, b_{in}(t), \quad a(0) = a. \tag{31.10}$$

In quadrature form,

$$\begin{bmatrix} \dot{q}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} -\frac{\gamma}{2} & \omega \\ -\omega & -\frac{\gamma}{2} \end{bmatrix} \begin{bmatrix} q(t) \\ p(t) \end{bmatrix} + \begin{bmatrix} -\sqrt{\gamma} & 0 \\ 0 & -\sqrt{\gamma} \end{bmatrix} \begin{bmatrix} b_{in,r}(t) \\ b_{in,i}(t) \end{bmatrix}. \tag{31.11}$$

Commutation relations are preserved. Equation 31.11 is a linear system of the form $\dot{x}(t) = Ax(t) + Bw(t)$, where A and B are the real matrices in Equation 31.11 and $w(t) = (b_{in,r}(t), b_{in,i}(t))^T$. The matrices A and B satisfy $iAJ + iJA^T + BTB^T = 0$, where $T = \frac{1}{2}(F - F^T)$. If this relation is satisfied, the Hamiltonian $H = \frac{1}{2} x^T R x$ and coupling operator $L = Mx$ are determined by $R = \frac{1}{4}(-JA + A^T J)$ and $M = \frac{\sqrt{\gamma}}{2}(1, i)$.

The output field $y(t) = (b_{out,r}(t), y_{out,i}(t))^T$ is given, in quadrature form, by

$$\begin{bmatrix} b_{out,r}(t) \\ b_{out,i}(t) \end{bmatrix} = \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & \sqrt{\gamma} \end{bmatrix} \begin{bmatrix} q(t) \\ p(t) \end{bmatrix} + \begin{bmatrix} b_{in,r}(t) \\ b_{in,i}(t) \end{bmatrix}. \tag{31.12}$$

This output equation is of the form $y(t) = Cx(t) + Dw(t)$. Note that $B = JC^T J$ and $D = I$.

31.3 Approaches to Quantum Estimation and Control

31.3.1 Estimation

Quantum estimation can be divided into a few broad categories: quantum state tomography (QST), quantum process tomography (QPT), and quantum parameter estimation (QPE). In QST, the density matrix ρ is estimated. In QPT, a matrix known as the *process matrix* is estimated from which OSR elements can be recovered. In QPT, uncertain parameters in a Hamiltonian model are estimated.

In both QST and QPT, the measurements are linear in the parameters to be estimated. Moreover, both the quantum state (density matrix) and the process matrix are constrained by the physics to convex sets. Approaches to both have naturally gravitated toward the well-established methods of least-squares and ML, for example, [45,48]. The resulting estimation problem is a convex optimization problem, and thus in principle, is tractable [49]. Unfortunately, however, the dimension of the parameter space for QST, and especially for QPT, can be prohibitive: for Hilbert space dimension n, QST scales with scaling n^2 and QPT with n^4. For q-qubits,* $n = 2^2$, and hence scaling for both is exponential in the number of qubits. Although this places a burden on computation, it also places the same burden on resources, for example, the number of applied inputs and measurement devices as well as the number of experiments to achieve a desired accuracy. A number of approaches have been developed to alleviate this scaling burden. Of note are the various forms of ancilla†-assisted QPT (see [50] for a review), the use of symmetrization to estimate selected process properties [51], and approaches which use prior modeling to simplify process matrix parameters [52]. With ancilla assistance the scaling power is reduced but is still exponential. Furthermore, ancilla-assisted methods may require entangled inputs which are very sensitive to noise and decoherence.

Recently, the use of compressed sensing (CS) methods of estimation [53–55] have been applied to QPT [56,57]. CS predicts a scaling of measurment resources of order $s \log N$ where s is the sparsity level of the N-dimensional signal to be estimated. In addition, the CS procedure requires solving a convex optimization problem. For QPT with q-qubits, $N = (2^q)^4$, and hence, CS heralds a scaling on the order of sq. As we will show later, and as posited in [56], for an initially well-designed system whose dynamics are close to a desired unitary (a primary goal in quantum computation) the process matrix, in a basis corresponding to the ideal unitary, is almost sparse, that is, an s-sparse estimate exists, which produces an estimation error below any desired level, modulo measurement noise.

QPE refers to estimating parameters in a model of the quantum system, typically a Hamiltonian model. An important subset of QPE is quantum metrology where quite often the information of interest is contained in a *single* parameter which cannot be measured directly, for example, estimation of the phase difference between the arms of an optical interferometer, or the transition frequency of an atomic clock. For example, Figure 31.9 shows a schematic representation of the classical Mach–Zehnder interferometer for phase estimation.

For single parameter (phase) estimation the limit of theoretical accuracy in the ideal noise-free case has been examined in depth, for example, [58–63]. These studies reveal that special preparation of the instrumentation—the probe—can achieve an asymptotic variance smaller than the Cramér–Rao lower bound, the so-called quantum Cramér–Rao bound, or the quantum Fisher information (QFI). Specifically, the unique quantum property of *entanglement* can increase the parameter estimation convergence from the classical limit of $1/\sqrt{N}$ to the Heisenberg limit $1/N$, which arises from the uncertainty principle [64]. In the latter case N refers to the dimension of an entangled state. The entangled state can be very sensitive to noise and decoherence, thus inhibiting the attainment of the theoretical QFI. In addition to this

* A qubit is the quantum analog of the classical information bit. Speceficlally a two-level state $|\psi\rangle = a|0\rangle + b|1\rangle$ with $|a|^2 + |b|^2 = 1$ which, unlike its classical counterpart, is in a superposition of both "0" and "1."

† Ancilla states are quantum states (and channels) intentionally added to the system states to enhance performance, for example, for quantum error correction quantum metrology.

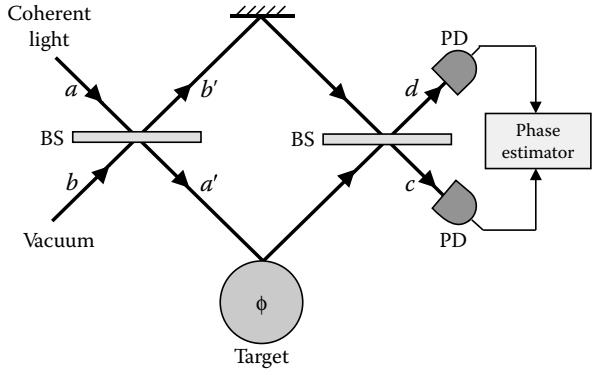

FIGURE 31.9 Classical Mach–Zehnder interferometer. A coherent light beam is split into two parts. The phase difference ϕ between the two optical arms is estimated by analyzing photon statistics of the two output beams. (BS: beam-splitter, PD: photodetector)

sensitivity, the QFI may also be unreachable simply because the instruments are limited, that is, not all states can be prepared and not all measurement schemes are possible.

In general, Hamiltonians are of course useful to understand most physical phenomena. If a quantum system is to be used to simulate the dynamics of another quantum system (Feynman's original idea for Quantum Computer applications), then the ability to accurately and efficiently characterize the quantum process will be crucial. It has been said that adaptive control experiments demonstrate that we have made a quantum computer: the only problem is that we do not know what equation is being solved, because we do not know H, the Hamiltonian [5,8,65]. Besides, H is actually manipulated by the external field, that is, the external field creates a "stationary" new Hamiltonian while the driving field is on. Although not discussed here, CS has been recently applied to the Hamiltonian estimation [66].

31.3.2 Control

How can we go about controlling a quantum system? Let us suppose the system Hamiltonian H governing Schrodinger's Equation 31.1 depends on an external control variable u, that is, $H = H_0 + H_1 u$, so that Schrodinger's equation takes the form

$$\dot{U} = -i(H_0 + H_1 u)U, \quad 0 \le t \le t_f, \quad U(0) = I. \tag{31.13}$$

Clearly, this equation is *bilinear*, since it involves the product of the control variable u and the unitary U at each time instant. The simplest type of control problem for the system Equation 31.13 is to choose an open-loop control signal $u(t)$ so that the unitary at a final time t_f, $U(t_f)$, equals or is close to a desired unitary, U_{des}. This problem is completely deterministic, and beginning with the pioneering paper [22], a large literature has accumulated studying this type of system using methods from the nonlinear control theory, and applying the results to a range of problems (e.g., [16,67], and in particular recent work on *dynamic decoupling* [68,69]). Since the evolution of quantum states depends on the unitary, one has control over the fundamental statistical behavior of the underlying physical system.

So is quantum control simply a form of classical nonlinear control? The answer is no if we wish to use *feedback* to control the quantum system. The advances that have been made since the early 1980s concerning feedback control of quantum systems depend on more sophisticated models. The development of these more advanced models was stimulated by rapid and significant progress in a key

area of quantum physics, namely, quantum optics. In due course, the lessons learned from the feedback control of quantum optical systems will influence the development of feedback control in other physical contexts. To the best of our knowledge, the earliest publication on quantum feedback control was the paper [10] by Belavkin, which discussed open quantum models, filtering, and optimal measurement feedback control. Later, Belavkin developed a general theory of *quantum filtering* [12], which includes the *stochastic master equation* from quantum optics [70]. A significant milestone in the development of quantum control theory is the work of Wiseman and Milburn in the early 1990s, [38,71,72]. An excellent text on quantum measurement and control has recently been published [73].

A wide range of control problems may be formulated in terms of *quantum noise* models for open quantum systems, of which the cavity model Equation 31.9 is an example, [47,74]. In particular, optimal control problems may be formulated by specifying a suitable cost function [17,23,75,76]. In general, these problems are difficult to solve analytically, though it is possible in principle to apply dynamic programming methods. However, as for classical stochastic systems, there is a class of *measurement feedback* problems for which closed-form solutions are available, namely, linear Gaussian quantum stochastic systems [17,75,77]. While the underlying unitary equation is bilinear in the control variable, special features of this class of linear quantum systems mean that the evolution of certain observables is given by *linear* equations, and these equations preserve Gaussian states—this is the reason for the computational tractability. In Section 31.5, we present these models and formulate and solve several control problems.

One may also formulate *coherent feedback* optimal control problems [25,32]. It turns out that the H^∞ problem is tractable because of the linear Gaussian model and the key fact is that for this problem, resolution of the physical constraints in the coherent controller can be decoupled from the optimization [25]. This is described in Section 31.5.5. However, the coherent feedback LQG problem does not appear to be computationally simple [32].

31.3.3 Adaptive and Learning Control

In quantum chemistry, adaptive and learning control has been sucessfully applied in the laboratory in hundreds of experiments (Figure 31.10) [65,78]. These are direct adaptive control systems with no model posed for the system; only a performance measure is available and the control parameters are adjusted to improve the performance. The adjustment "directions," however, clearly must depend on the shape of the "control parameter landscape," otherwise, it would not be possible to know how to make the adjustment. In general it would be expected that the control landscape would have many local optimal values. Thus, without some knowledge of the landscape or an exhaustive search, it might be difficult to find a global optimum. Surprisingly, however, an in-depth analysis of the landscape for the control of quantum systems shows that this is not the case [78,79]. There it is shown that for unconstrained time-varying controls, if the system is controllable, then *all* the local maximums are global, that is, the outcome probability at every local maximum is unity and all other extrema give the minimum probability of zero. However, when there are control constraints the landscape may then exhibit structure that was not evident in the freely floating original set.

The fact that the experiments work at all is rather amazing. A guess is that an examination of the landscapes will show considerable detail, much of which is likely to be false structures arising from having highly constrained controls. From a positive perspective, as more bandwidth becomes available, the control landscape becomes less complicated and more regular in the sense that more of the local optimum values provide performance close to the global optimum.

31.4 Quantum Estimation

In the next few subsections we shall briefly review the algorithms for quantum estimation. Many of these are based on the intrinsic convexity of quantum mechanical variables.

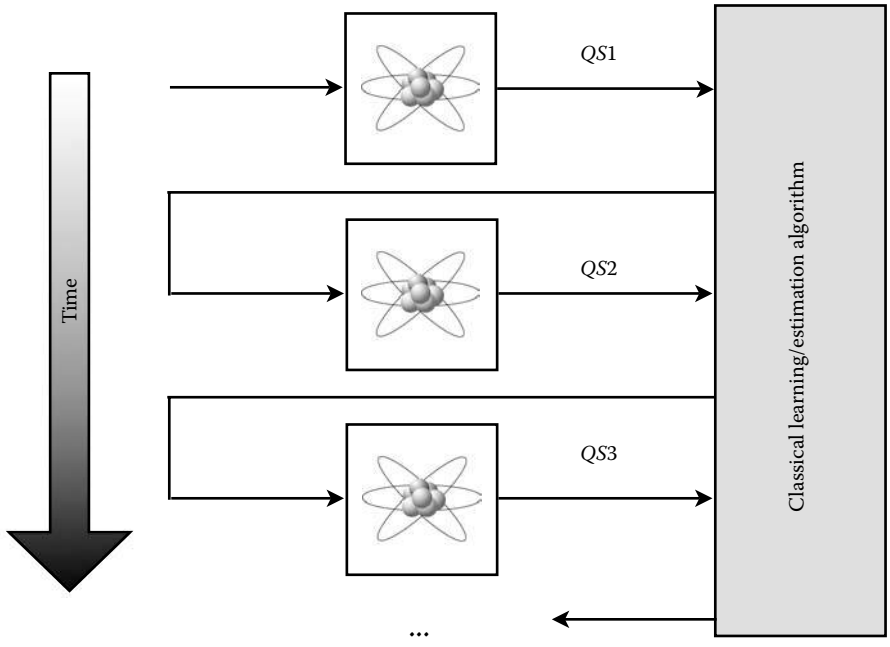

FIGURE 31.10 Iterative learning control. The same scheme is used for estimation from repeated identical experiments. A fresh quantum system is used in each iteration, *QS*1, *QS*2, *QS*3, ...

Applications include: ML estimation and optimal experiment design (OED) for state and process tomography and Hamiltonian parameter estimation, quantum state detection, and quantum error correction. The great advantage of convex optimization is that a globally optimal solution can be found efficiently and reliably.

ML estimation problems include state (density) estimation, ML estimation of the the distribution of known input states, ML estimation of the OSR elements for QPT, and ML estimation of Hamiltonian parameters. Associated with these estimation problems is an OED, invoked by the Cramer–Rao Inequality, which can determine the system configurations to maximize the estimation accuracy.

Designing a detector which is maximally sensitive to specific quantum states can be formulated as a convex optimization problem in the matrices of the POVM, which characterize the measurement apparatus. For example, maximizing the posterior probability of detection is a quasiconvex optimization problem in the POVM elements.

Designing a quantum information error correcting procedure can be cast as a biconvex optimization problem, iterating between encoding and recovery, each being a semidefinite program. For a given encoding operator the problem is convex in the recovery operator. For a given method of recovery, the problem is convex in the encoding scheme. This allows the derivation of codes which are more robust than the standard codes with respect to a range of uncertainty in the error system.

31.4.1 Quantum State Tomography

31.4.1.1 Collecting Data

For quantum state estimation, data is most often collected from identical experiments in each configuration γ repeated ℓ_γ times. The setup for data collection for QST is shown schematically in Figure 31.11 for configuration γ.

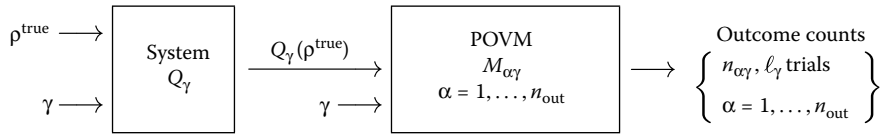

FIGURE 31.11 Data collection in configuration γ for QST.

Here, $\rho^{\text{true}} \in \mathbf{C}^{n\times n}$ is the true, the unknown state to be estimated, $n_{\alpha\gamma}$ is the number of times outcome α is obtained from the ℓ_γ experiments, and $\{M_{\alpha\gamma}\}$ are the POVM elements of the measurement apparatus. The *data set* thus consists of all the outcome counts

$$D = \left\{ n_{\alpha\gamma} \,\middle|\, \alpha = 1, \ldots, n_{\text{out}}, \gamma = 1, \ldots, n_{\text{cfg}} \right\}. \tag{31.14}$$

If $p^{\text{true}}_{\alpha\gamma}$ is the true probability of obtaining outcome α when the system is in configuration γ with state input ρ^{true}, then,

$$\mathbb{P}\, n_{\alpha\gamma} = \ell_\gamma p^{\text{true}}_{\alpha\gamma} \tag{31.15}$$

where the expectation $\mathbb{P}(\cdot)$ taken with respect to the underlying quantum probability distributions with respect to ρ^{true}. We pose the following *model* of the system

$$p_{\alpha\gamma}(\rho) = \text{Tr}\, M_{\alpha\gamma} Q_\gamma(\rho) \tag{31.16}$$

where $p_{\alpha\gamma}(\rho)$ is the outcome probability of measuring α when the system is in configuration γ with input state ρ belonging to the set of density matrices

$$\left\{ \rho \in \mathbf{C}^{n\times n} \mid \rho \geq 0,\ \text{Tr}\, \rho = 1 \right\}. \tag{31.17}$$

If Q_γ is modeled as an OSR with elements $\{K_{\gamma k}\}$, then the model probability outcomes are linear in the input state, that is,

$$p_{\alpha\gamma}(\rho) = \text{Tr}\, O_{\alpha\gamma}\rho, \quad O_{\alpha\gamma} = \sum_{k=1}^{\kappa_\gamma} K^\dagger_{\gamma k} M_{\alpha\gamma} K_{\gamma k}. \tag{31.18}$$

Moreover, the set $\{O_{\alpha\gamma}\}$ is a POVM. If Q_γ is modeled as a unitary system, then

$$Q_\gamma(\rho) = U_\gamma \rho U^\dagger_\gamma, \quad U^\dagger_\gamma U_\gamma = I_n \implies O_{\alpha\gamma} = U^\dagger_\gamma M_{\alpha\gamma} U_\gamma. \tag{31.19}$$

The set O_γ is still a POVM with a single element, $K_\gamma = U_\gamma$.

Figure 31.12 shows a schematic representation of an experimental setup for QST performed in the Clarendon Lab at Oxford. In this case, for the probability outcomes $p_{\alpha\gamma}$, the outcomes $\alpha \in \{0, 1\}$ and the configurations $\gamma \in \{\Omega, T\}$ were selected from frequency and time of the fluoresence signal [80].

31.4.1.2 Maximum Likelihood

The ML approach to quantum state estimation presented in this section, as well as observing that the estimation is convex, can be found in [81,82] and the references therein. Using convex programming methods, such as an interior-point algorithm for computation, was not exploited in these references.

If the experiments are independent, then the probability of obtaining the data (Equation 3.14) is a product of the individual model probabilities (Equation 3.16). Consequently, for an *assumed* initial state ρ, the model predicts that the probability of obtaining the data set (Equation 3.14) is given by, $\mathbf{P}\{D, \rho\} = \prod_{\alpha,\gamma} p_{\alpha\gamma}(\rho)^{n_{\alpha\gamma}}$. The data are thus captured in the outcome counts $\{n_{\alpha\gamma}\}$, whereas the model terms have a ρ-dependence. The ML estimate of ρ is obtained by finding a ρ in the set (Equation 3.17) which maximizes

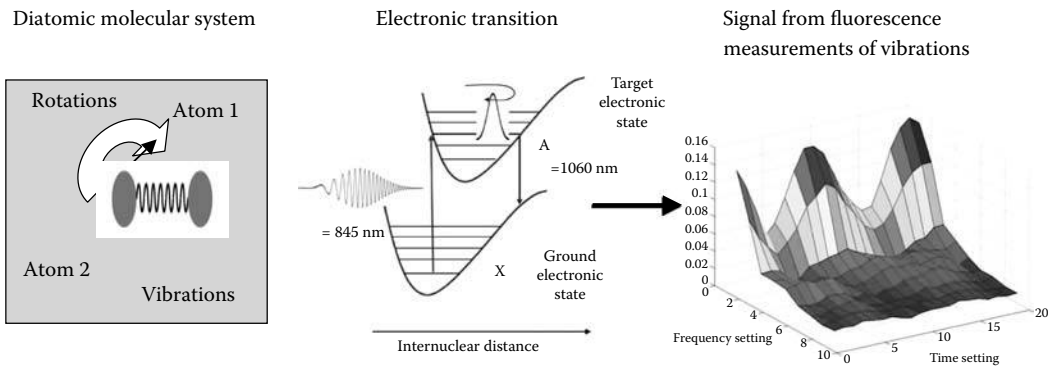

FIGURE 31.12 QST in the lab. State tomography of vibrational wavepackets in diatomic molecules.

the $\mathbf{P}\{D, \rho\}$, or equivalently, minimizes the *negative log-likelihood function*, $L(D, \rho) = -\log \mathbf{P}\{D, \rho\}$. The ML state estimate, ρ^{ML}, is obtained as the solution to the optimization problem

$$\begin{array}{ll} \text{minimize} & L(D, \rho) = -\sum_{\alpha,\gamma} n_{\alpha\gamma} \log \mathrm{Tr}\, O_{\alpha\gamma}\rho \\ \text{subject to} & \rho \geq 0, \quad \mathrm{Tr}\, \rho = 1. \end{array} \tag{31.20}$$

$L(D, \rho)$ is a positively weighted sum of log-convex functions of ρ, and hence, is a log-convex function of ρ. The constraint that ρ is a density matrix forms a convex set in ρ. Hence, Equation 31.20 is in a category of a class of well-studied log-convex optimization problems, for example, [49].

31.4.1.3 Least Squares

In a typical application, the number of trials per configuration, ℓ_γ, is sufficiently large so that the *empirical estimate* of the outcome probability, is a good estimate of the true outcome probability $p^{\mathrm{true}}_{\alpha\gamma}$

$$p^{\mathrm{emp}}_{\alpha\gamma} = \frac{n_{\alpha\gamma}}{\ell_\gamma} \approx p^{\mathrm{true}}_{\alpha\gamma}. \tag{31.21}$$

This leads to the least squares (LS) state estimate ρ^{LS} as the solution to the constrained weighted LS problem

$$\begin{array}{ll} \text{minimize} & \sum_{\alpha,\gamma} \left[p^{\mathrm{emp}}_{\alpha\gamma} - \mathrm{Tr}\, O_{\alpha\gamma}\rho\right]^2 \\ \text{subject to} & \rho \geq 0, \quad \mathrm{Tr}\, \rho = 1. \end{array} \tag{31.22}$$

This is clearly a convex optimization problem. For large ℓ_γ, the LS solution and ML solutions (Equation 31.20) are nearly the same.

31.4.1.4 Optimal Experiment Design

In this section, we describe the experiment design problem for quantum state estimation. The objective is to select the number of experiments per configuration, the elements of the vector $\ell = [\ell_1 \cdots \ell_{n_{\mathrm{cfg}}}]^T \in \mathbf{R}^{n_{\mathrm{cfg}}}$, so as to minimize the error between the state estimate, $\widehat{\rho}(\ell)$, and the true state ρ^{true}. Specifically, we would like to solve for ℓ from

$$\begin{array}{ll} \text{minimize} & \mathbb{P}\, \|\widehat{\rho}(\ell) - \rho^{\mathrm{true}}\|^2_{\mathrm{frob}} \\ \text{subject to} & \sum_\gamma \ell_\gamma = \ell_{\mathrm{expt}}, \quad \text{integer } \ell_\gamma \geq 0,\ \gamma = 1, \ldots, n_{\mathrm{cfg}} \end{array} \tag{31.23}$$

where ℓ_{expt} is the desired number of total experiments. This is a difficult, if not insoluble problem for several reasons. Firstly, the solution depends on the estimation method which produces $\widehat{\rho}(\ell)$. Second, the problem is integer combinatorial, because ℓ is a vector of integers. Finally, the solution depends on ρ^{true},

the very state to be estimated. Fortunately, all these issues can be circumvented. We first eliminate the dependence on the estimation method. The following result is established in [83] using the *Cramér–Rao Inequality* [84].

31.4.1.4.1 State Estimation Variance Lower Bound

For $\ell = [\ell_1 \cdots \ell_{n_{\text{cfg}}}]$ experiments per configuration, suppose $\widehat{\rho}(\ell)$ is an unbiased estimate of ρ^{true}, that is, $\mathbb{P}\,\widehat{\rho}(\ell) = \rho^{\text{true}}$. Then the estimation error variance satisfies,

$$\begin{aligned} &\mathbb{P}\,\|\widehat{\rho}(\ell) - \rho^{\text{true}}\|_{\text{frob}}^2 \geq V(\ell, \rho^{\text{true}}) = \text{Tr}\left[\sum_{\gamma=1}^{n_{\text{cfg}}}\ \ell_\gamma G_\gamma(\rho^{\text{true}})\right]^{-1} \\ &G_\gamma(\rho^{\text{true}}) = C_{\text{eq}}^T\left[\sum_\alpha\ (\mathbf{vec}\ O_{\alpha\gamma})(\mathbf{vec}\ O_{\alpha\gamma})^\dagger / p_{\alpha\gamma}(\rho^{\text{true}})\right] C_{\text{eq}} \in \mathbf{R}^{n^2-1\times n-1^2} \end{aligned} \tag{31.24}$$

where $C_{\text{eq}} \in \mathbf{R}^{n^2\times n^2-1}$ arises from the equality constraint $\text{Tr}\ \rho = 1$.

The experiment design problem can be expressed by the following optimization problem in the vector of integers ℓ:

$$\begin{array}{ll} \text{minimize} & V(\ell, \rho^{\text{true}}) \\ \text{subject to} & \sum_\gamma\ \ell_\gamma = \ell_{\text{expt}},\ \text{integer}\ \ell_\gamma \geq 0,\ \gamma = 1, \ldots, n_{\text{cfg}} \end{array} \tag{31.25}$$

where ℓ_{expt} is the desired number of total experiments. The good news is that the objective, $V(\ell, \rho^{\text{true}})$, is convex in ℓ [49, §7.5]. Unfortunately, there are still two impediments: (1) restricting ℓ to a vector of integers makes the problem combinatorial; (2) the lower-bound function $V(\ell, \rho^{\text{true}})$ depends on the true value, ρ^{true}. These difficulties can be alleviated to some extent. For (1) we can use the convex relaxation described in [49, §7,5]. For (2) we can solve the relaxed experiment design problem with either a set of "what-if" surrogates ρ^{surr} for ρ^{true}. These can be used to start and then "bootstrap" to more precise values by iterating between state estimation and experiment design.

Following the procedure in [49, §7.5], introduce the variables $\lambda_\gamma = \ell_\gamma/\ell_{\text{expt}}$, each of which is the fraction of the total number of experiments performed in configuration γ. As all the ℓ_γ and ℓ_{expt} are non-negative integers, each λ_γ is non-negative and *rational*, specifically an integer multiple of $1/\ell_{\text{expt}}$, and in addition, $\sum_\gamma \lambda_\gamma = 1$. If λ_γ is only otherwise constrained to the non-negative reals, then this has the effect of relaxing the constraint that ℓ_γ are integers. The *relaxed* experiment design problem is

$$\begin{array}{ll} \text{minimize} & V(\lambda, \rho^{\text{surr}}) = \text{Tr}\left[\sum_\gamma\ \lambda_\gamma G_\gamma(\rho^{\text{surr}})/\ell_{\text{expt}}\right]^{-1} \\ \text{subject to} & \sum_\gamma\ \lambda_\gamma = 1, \quad \lambda_\gamma \geq 0,\ \gamma = 1, \ldots, n_{\text{cfg}}. \end{array} \tag{31.26}$$

This is a convex optimization problem in $\lambda \in \mathbf{R}^{n_{\text{cfg}}}$. Let λ^{opt} denote the optimal solution to Equation 31.26. Since the problem no longer depends on ℓ_{expt}, λ^{opt} can be viewed as a distribution of experiments per configuration. Clearly, there is no guarantee that $\ell_{\text{expt}}\lambda^{\text{opt}}$ is a vector of integer multiples of $1/\ell_{\text{expt}}$. A practical choice for obtaining a vector of integer multiples of $1/\ell_{\text{expt}}$ is, $\ell_{\text{expt}}^{\text{round}} = \textbf{round}\left\{\ell_{\text{expt}}\lambda^{\text{opt}}\right\}$. If ℓ^{opt} is the (unknown) integer vector solution to Equation 31.25, then we have the relation:

$$V(\ell_{\text{expt}}^{\text{round}}, \rho^{\text{surr}}) \geq V(\ell^{\text{opt}}, \rho^{\text{surr}}) \geq V(\ell_{\text{expt}}\lambda^{\text{opt}}, \rho^{\text{surr}}). \tag{31.27}$$

The optimal objective is thus bounded above and below by known values obtained from the relaxed optimization. The gap within which the optimal solution falls can be no worse than the difference between $V(\ell_{\text{expt}}^{\text{round}}, \rho^{\text{surr}})$ and $V(\ell_{\text{expt}}\lambda^{\text{opt}}, \rho^{\text{surr}})$, which can be computed solely from λ^{opt}. If the gap is sufficiently small, then for all practical purposes the "optimal" solution is λ^{opt}.

31.4.1.4.2 Numerical Example—Single-Photon State Tomography

A schematic representation of an apparatus for state tomography of a single photon specified by the quantum state (density matrix) ρ is shown in Figure 31.13. The set up has two photon-counting detectors, A, B. There are two continuous variable settings for the quarter-wave (q) plates and half-wave plates (h).

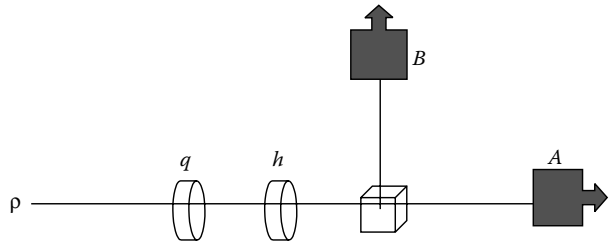

FIGURE 31.13 Single-photon detector.

(These are devices which can be adjusted to change the polarization of light.) For any setting of the angle parameters (q, h), one of the detectors in each arm registers a photon. The objective is to determine the optimal setting of these parameters and the number of experiments per setting for the estimation of the state ρ using the photon counts from the two detectors as data. As the photon sources are not completely efficient, the input quantum state actually consists of either one or zero photons. The detectors register 0 or 1 depending on whether a photon is incident on them or not.

Assume that the incoming state is *always* one photon, never none and that each detector has an efficiency η, $0 \le \eta \le 1$ and a nonzero dark count probability, δ, $0 \le \delta \le 1$. Hence, $1-\eta$ is the probability of no detection and $1-\delta$ is the probability of no dark count. Under these conditions there are four possible outcomes at detectors A,B denoted by the double index $\alpha \in \{10, 01, 00, 11\}$. Optimal experiment distributions are obtained for a pure-state and a mixed-state input

$$\rho_{\text{pure}} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \psi_0 \psi_0^{\dagger}, \quad \psi_0 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \rho_{\text{mixd}} = \begin{bmatrix} 0.6 & -0.2i \\ 0.2i & 0.4 \end{bmatrix}. \tag{31.28}$$

We compute the distributions $\lambda_{\text{pure}}^{\text{opt}}$, $\lambda_{\text{mixd}}^{\text{opt}}$ for each input state with and without "noise" arising from detector efficiency η and dark count probability δ. With no noise ($\eta = 1, \delta = 0$). With noise ($\eta = 0.75, \delta = 0.05$). For all cases and noise conditions we used the wave-plate settings

$$\begin{aligned} h_i &= (i-1)(5^\circ), \quad i = 1, \ldots, 10 \\ q_i &= (i-1)(5^\circ), \quad i = 1, \ldots, 10. \end{aligned} \tag{31.29}$$

Both angles are set from 0 to 45° in 5° increments. This yields a total of $n_{\text{cfg}} = 10^2 = 100$ configurations corresponding to all the wave-plate combinations. Figure 31.14 shows the optimal distributions λ^{opt} versus configurations $\gamma = 1, \ldots, 100$ for all four test cases: two input states with and without noise. Observe that the optimal distributions are *not* uniform, but are concentrated near the same particular wave plate settings.

It is interesting to note that the OED except for the pure state with no noise, and the other distributions display a "spread" of experiments. This gives rise to a more robust estimation in the sense that these OED distributions can handle larger deviations from the nominal (surrogate) design.

To check the gap between the relaxed optimum λ^{opt} and the unknown integer optimum we appeal to Equation 31.27. The following table shows that these distributions are a good approximation to the

ℓ_{expt}	$\dfrac{V(\ell_{\text{expt}}\lambda_{\text{pure}}^{\text{opt}}, \rho_{\text{pure}})}{V(\ell_{\text{expt}}^{\text{round}}(\rho_{\text{pure}}), \rho_{\text{pure}})}$	$\dfrac{V(\ell_{\text{expt}}\lambda_{\text{mixd}}^{\text{opt}}, \rho_{\text{mixd}})}{V(\ell_{\text{expt}}^{\text{round}}(\rho_{\text{mixd}}), \rho_{\text{mixd}})}$
100	0.9797	0.7761
1000	0.9950	0.9735
10,000	0.9989	0.9954

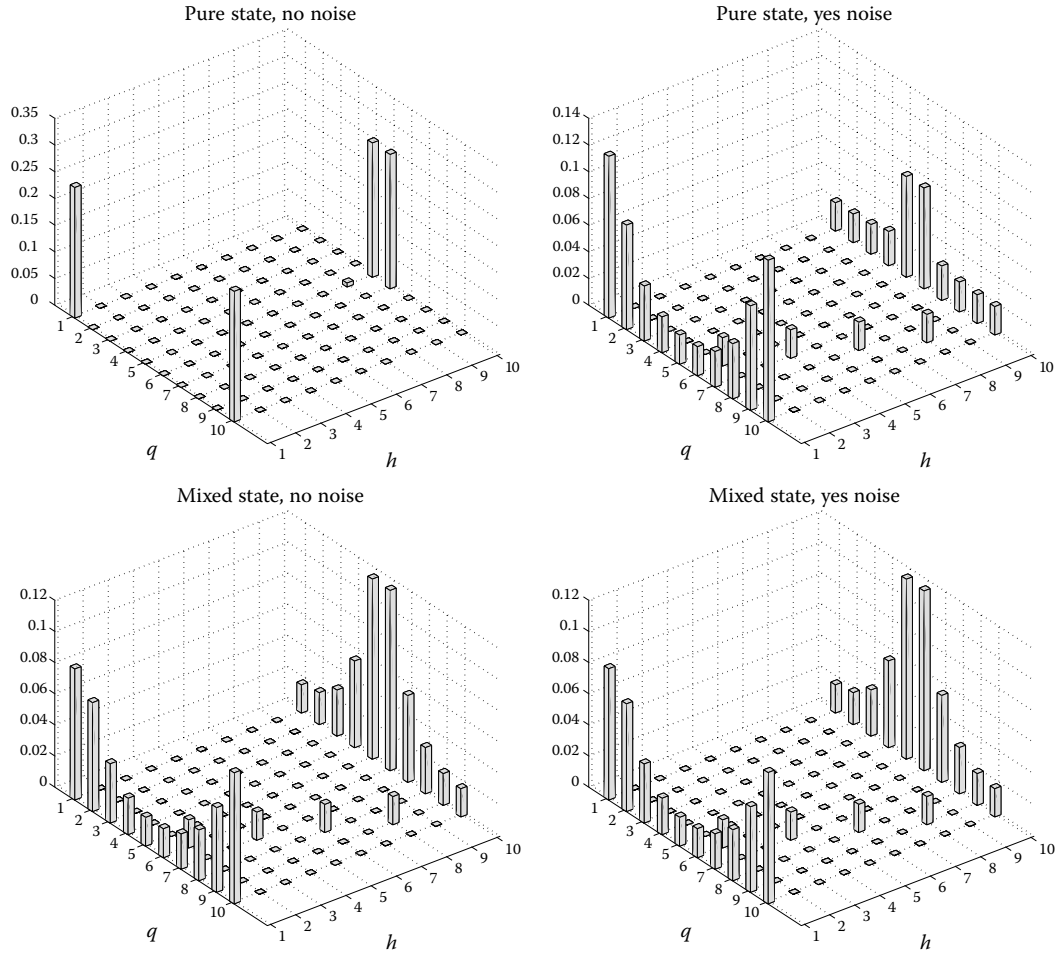

FIGURE 31.14 OED wave-plate distributions.

unknown optimal integer solution for even not so large ℓ_{expt} for the two state cases with no noise. Similar results were obtained for the noisy case.

31.4.2 Quantum Process Tomography

QPT refers to the use of measured data to estimate the OSR elements of the quantum process [45,48]. QPT is thus a means to characterize the dynamics of almost any quantum system. For quantum information systems in particular, QPT is a means to determine if the system is performing as desired. QPT may also be used to estimate the specific system errors which can then be alleviated by optimal error correction tuned to these errors [6].

31.4.2.1 Collecting Data

The setup for collecting data for QPT is similar to that of Figure 31.11 as shown in Figure 31.15.

The main difference between state tomography (Figure 31.11) and process tomography (Figure 31.15) is that here the input state is known and prepared at specific values, ρ_γ, depending on the configuration (γ), whereas the Q-system, which is to be estimated, does not depend on the configuration. Figure 31.16

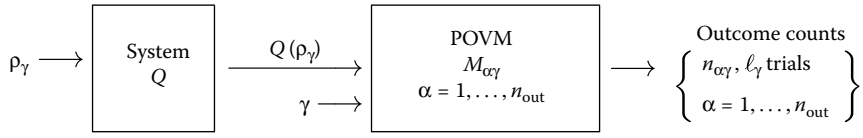

FIGURE 31.15 System/POVM.

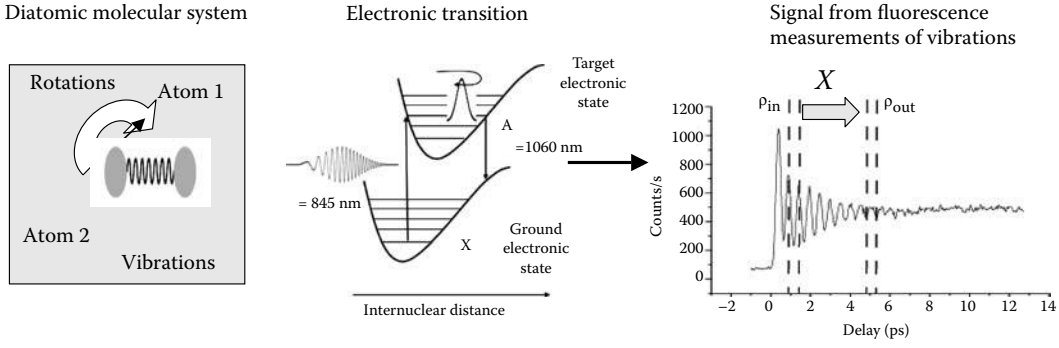

FIGURE 31.16 QPT in the lab. Process tomography of vibrational wavepackets in diatomic molecules.

shows a schematic reperesentation of a QPT setup in the Clarendon lab at Oxford. The "X" shown is the process matrix (defined below), which is to be estimated from the fluorescence data [52].

31.4.2.2 OSR Model

As for QST, if Q is modeled as an OSR with elements $\{K_k\}$, then the model probability outcomes are quadratic functions of the OSR elements, that is,

$$p_{\alpha\gamma}(\rho) = \text{Tr}\, O_{\alpha\gamma}\rho, \quad O_{\alpha\gamma} = \sum_{k=1}^{\kappa} K_k^\dagger M_{\alpha\gamma} K_k, \quad \sum_{k=1}^{\kappa} K_k^\dagger K_k = I. \tag{31.30}$$

In addition, the number of OSR elements is not always known. To circumvent these difficulties, the standard approach is to express the OSR elements in a matrix basis for $\mathbf{C}^{n\times n}$, that is,

$$\left\{ B_i \in \mathbf{C}^{n\times n} \;\middle|\; i = 1, \ldots, n^2 \right\} \;\Rightarrow\; K_k = \sum_{i=1}^{n^2} x_{ki} B_i, \;\; k = 1, \ldots, \kappa. \tag{31.31}$$

The n^2 coefficients $\{x_{ki}\}$ are complex scalars. Introduce the matrix $X \in \mathbf{C}^{n^2\times n^2}$, often referred to as the *process matrix*, with elements

$$X_{ij} = \sum_{k=1}^{\kappa} x_{ki}^* x_{kj}, \quad i, j = 1, \ldots, n^2. \tag{31.32}$$

The trace-preserving condition then becomes

$$\sum_{i,j=1}^{n^2} X_{ij}\, B_i^\dagger B_j = I_n \tag{31.33}$$

which is a linear constraint in X. The outcome probabilities (Equation 31.30) now become

$$p_{\alpha\gamma}(X) = \text{Tr}\, X R_{\alpha\gamma}, \quad [R_{\alpha\gamma}]_{ij} = \text{Tr}\, B_j \rho_\gamma B_i^\dagger O_{\alpha\gamma}, \;\; i, j = 1, \ldots, n^2. \tag{31.34}$$

QPT is then estimated $X \in \mathbf{C}^{n^2\times n^2}$ from the data set D (Equation 31.14) subject to the quadratic equality (Equation 31.32) and linear equality (Equation 31.33). By relaxing the quadratic equality constraint (Equation 31.32) to the semidefinite constraint $X \geq 0$, we can obtain a relaxed ML estimate by solving for X from

$$\begin{array}{ll} \text{minimize} & L(D,X) = -\sum_{\alpha,\gamma}\ n_{\alpha\gamma} \log \operatorname{Tr} XR_{\alpha\gamma} \\ \text{subject to} & X \geq 0, \quad \sum_{ij}\ X_{ij}\, B_i^\dagger B_j = I_n. \end{array} \tag{31.35}$$

Similarly, a relaxed LS estimate is the solution of

$$\begin{array}{ll} \text{minimize} & V(D,X) = \sum_{\alpha,\gamma}\ \left[p_{\alpha\gamma}^{\text{emp}} - \operatorname{Tr} XR_{\alpha\gamma}\right]^2 \\ \text{subject to} & X \geq 0, \quad \sum_{ij}\ X_{ij}\, B_i^\dagger B_j = I_n. \end{array} \tag{31.36}$$

Both these problems are essentially in the same form as Equations 31.20 and 31.22, respectively. Hence both are convex optimization problems with the optimization variables here being the elements of the matrix X. Since $X = X^\dagger \in \mathbf{C}^{n^2\times n^2}$, it can be parameterized by n^4 real variables. Accounting for the n^2 real linear equality constraints, the number of free (real) variables in X is thus $n^4 - n^2$. This can be quite large even for a relatively small number of qubits, for example, for $q = [1, 2, 3, 4]$ qubits, $n = 2^q = [2, 4, 8, 16]$, and $n^4 - n^2 = [12, 240, 4032, 65280]$. This exponential (in qubit) growth is the main drawback in using this approach.

The process matrix X can be transformed back to an OSR via a singular value decomposition (SVD). Specifically, let $X = VSV^\dagger$ with unitary $V \in \mathbf{C}^{n^2\times n^2}$ and $S = \operatorname{diag}(s_1 \ldots s_{n^2})$ with the singular values ordered so that $s_1 \geq s_2 \geq \cdots \geq s_{n^2} \geq 0$. Then the coefficients in *this* basis representation of the OSR elements are

$$x_{ki} = \sqrt{s_k}\ V_{ik}^*, \quad k, i = 1, \ldots, n^2. \tag{31.37}$$

As we now can recover an OSR from the relaxed optimizations (Equation 31.35 or Equation 31.36), we have actually found optimal solutions, that is, the relaxed solutions are optimal.

Unfortunately, as already mentioned, the dimension of the parameter space ($n^4 - n^2$) can severely strain resources to the point of impracticality. To see this more clearly, let the linear relation in Equation 31.34 between the $n_{\text{out}}n_{\text{cfg}}$ model probability outcomes and the n^4 elements of the process matrix be represented by an $n_{\text{out}}n_{\text{cfg}} \times n^4$ matrix $\mathcal{G}$, that is,

$$\mathbf{vec}(P) = \mathcal{G}\mathbf{vec}(X) \tag{31.38}$$

where $\mathbf{vec}(P), \mathbf{vec}(X)$ are vectors formed from the p_{ik} and elements of X, respectively. Accounting for the n^2 linear constraints in Equation 31.33, X can be recovered from either Equation 31.35 or Equation 31.36 to within any desired accuracy by using a sufficiently large number of repetitive experiments (ℓ_{expt}), provided that $\mathbf{rank}(\mathcal{G}) \geq n_{\text{out}}n_{\text{cfg}} \geq n^4 - n^2$. Therefore, the experimental resources, $n_{\text{out}}n_{\text{cfg}}$, must also scale exponentially with the number of qubits. Again, this is the bane of QPT for even modest size systems.

31.4.2.3 Sparse Process Matrix

Is is often the case that the process matrix is sparse or *almost sparse*, that is, it consists of a small number of significant elements. In some cases, a known sparsity pattern can arise from the underlying dynamics, thereby inherently increasing QPT efficiency [85]. In most cases, however, the sparsity pattern is not known. In this more common case we can apply the methods of CS [53–55]. Specifically, for a class of incomplete linear measurement equations ($y = Ax,\ A \in \mathbf{R}^{m\times N},\ m \ll N$), constrained ℓ_1-norm minimization (minimize $\|x\|_{\ell_1}$ subject to $y = Ax$), a convex optimization problem can perfectly estimate the sparse variable x. These methods are also robust to measurement noise and for almost sparse variables.

For a quantum information system the ideal quantum logic gates are unitaries, that is, $Q(\rho) = U\rho U^\dagger$. Let $\{\overline{B}_\alpha \in \mathbf{C}^{n^2\times n^2}, \alpha = 1, \ldots, n^2\}$ denote the "Natural-Basis," that is, each basis matrix has a single nonzero element of one. In this basis, the process matrix associated with the ideal unitary channel has the rank-1 form, $X_{\text{nat}} = xx^\dagger$ with $x \in \mathbf{C}^{n^2},\ x^\dagger x = n$. A SVD gives $X_{\text{nat}} = V\operatorname{diag}(n, 0, \ldots, 0)V^\dagger$ with $V \in \mathbf{C}^{n^2\times n^2}$ a

unitary. An equivalent process matrix can be formed from the SVD in what is referred to here as the "SVD-Basis," $\{B_\alpha = \sum_{\alpha'=1}^{n_S^2} V_{\alpha'\alpha}\overline{B}_{\alpha'}, \alpha = 1, \ldots, n^2\}$. The equivalent process matrix for a unitary channel, in this basis, denoted by X_{svd}, is maximally sparse with a single nonzero element, specifically, $(X_{\text{nat}})_{11} = n$. The actual channel which interacts with an environment will be a perturbation of the ideal unitary. If the noise source is small, then the process matrix in the nominal basis will be almost sparse.

As an example, consider a system which is ideally a two-qubit ($n = 4$) quantum memory, thus $U = I_4$. Suppose the actual system is a perturbation of identity by independent bit-flip errors in each channel occurring with probability p_{bf}. For $p_{\text{bf}} = 0.05$ and $p_{\text{bf}} = 0.2$, the respective channel fidelities are about 0.90 and 0.64, which for quantum information processing would need to be discovered by QPT and then corrected for the device to ever work. Referring to Figure 31.17, in the Natural-Basis, the ideal 16×16 process matrix has 16 nonzero elements out of 256, all of magnitude one. Using the SVD-Basis, the corresponding process matrix as shown in Figure 31.17b has a *single* nonzero element of magnitude

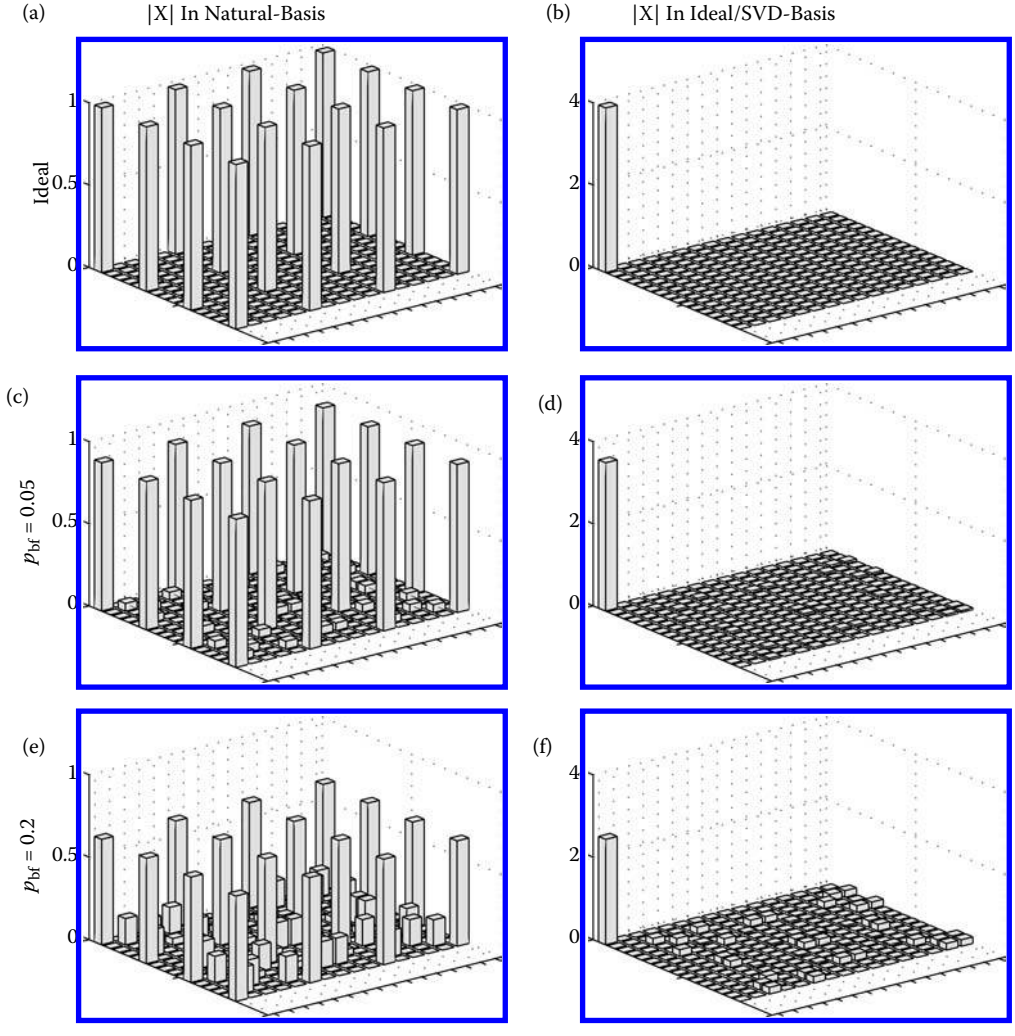

FIGURE 31.17 Absolute values of the elements of the process matrix $X \in \mathbf{C}^{16\times16}$ for: (a) Ideal (unitary) in Natural-Basis; (b) Ideal (unitary) in SVD-Basis; (c) Actual ($p_{\text{bf}} = 0.05$) in Natural-Basis; (d) Actual ($p_{\text{bf}} = 0.05$) in SVD-Basis; (e) actual ($p_{\text{bf}} = 0.2$) in Natural-Basis; (f) Actual ($p_{\text{bf}} = 0.2$) in SVD-Basis.

$n = 4$—it is clearly maximally sparse. Figure 31.17c–f, respectively, shows the effect of the two p_{bf} levels in the two basis sets. In the SVD-basis, Figure 31.17d and f show that the actual (noisy) process matrices are almost sparse.

31.4.2.4 QPT Via CS

A known heuristic for minimizing sparsity without knowing the sparsity pattern, and also accruing the benefit of using fewer resources, is to minimize the ℓ_1-norm of the vector of variables [49,53,55]. For QPT, the equivalent ℓ_1 norm is defined here as the sum of the absolute values of the elements of the process matrix. Specifically, an estimate of X can be obtained by solving the following convex optimization problem

$$\begin{array}{ll} \text{minimize} & \|\mathbf{vec}(X)\|_{\ell_1} = \sum_{\alpha,\beta=1}^{n^2} |X_{\alpha\beta}| \\ \text{subject to} & V(X) = \|\mathbf{vec}(P^{\text{emp}}) - \mathcal{G}\mathbf{vec}(X)\|_{\ell_2} \le \sigma, \quad X \ge 0, \quad \sum_{ij} X_{ij} B_i^\dagger B_j = I_n. \end{array} \tag{31.39}$$

As shown in [57], if $\mathcal{G}$ is a random matrix which satisfies a "concentration inequality," and if $V(X) \le \sigma$, then with high probability the error between the optimal estimate $X^\star$ from Equation 31.39 and the true process matrix X_{true} scales as

$$\|\mathbf{vec}(X^\star - X_{\text{true}})\|_{\ell_2} = \mathcal{O}\left(\frac{1}{\sqrt{s}}\|\mathbf{vec}(X_{\text{true}}(s) - X_{\text{true}})\|_1\right) + \mathcal{O}(\delta) \tag{31.40}$$

provided that $n_{\text{out}} n_{\text{cfg}} = \mathcal{O}(s \log(n^4/s))$. Here, $X_{\text{true}}(s)$ is the best s-sparse approximation of X_{true}; the former of course is not known. Thus, if X_{true} is truly s-sparse and there is no noise ($\sigma = 0$), then the true process matrix is perfectly recovered with high probability.

In practice, the parameter σ in Equation 31.39 is used to regulate the trade-off between fitting X to the data by minimizing $V(X)$ versus minimizing the sparsity of X via the ℓ_1-norm. Selecting σ is often done by averaging $V(X)$ over a series of surrogates for X obtained from anticipated scenarios.

31.4.2.4.1 Numerical Example—QPT of Noisy Two-Qubit Memory

In Each of the examples, to follow the procedure for QPT is: (1) solve Equation 31.36 with a complete measurement set to obtain X_{ℓ_2}; (2) set $\sigma = 1.3\ V(X_{\ell_2})$; (3) solve Equation 31.39 for X_{ℓ_1}.

For the systems from the example in Figure 31.17, the inputs and measurements are selected from the set of two-qubit states: $|a\rangle$, $|+\rangle = (|a\rangle + |b\rangle)/\sqrt{2}$, $|-\rangle = (|a\rangle - i|b\rangle)/\sqrt{2}$ with $a, b = 1, \ldots, 16$. Specifically, the available set of states are the 16 columns of the matrices

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad \text{and} \quad \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -i & 0 & 0 & 1 & 1 & 0 \\ 0 & -i & 0 & -i & 0 & 1 \\ 0 & 0 & -i & 0 & -i & -i \end{bmatrix}. \tag{31.41}$$

Considering that only coincident input/measurement counts [86], the relevant probability outcomes (Equation 31.34) are

$$\begin{aligned} p_{ab}(X) &= g_{ab}^\dagger X g_{ab}, \quad X \in \mathbf{C}^{16\times 16} \\ (g_{ab})_\alpha &= \phi_a^\dagger \Gamma_\alpha \phi_b, \quad \alpha = 1, \ldots, 16 \end{aligned} \tag{31.42}$$

with ϕ_a, ϕ_b $(a, b) \in \{1, \ldots, 16\}$ the selected columns of Equation 31.41.

Figure 31.18 shows the error in estimating the process matrix $\Delta X = X_{\text{true}} - X_{\text{est}}$ as measured by the RMS matrix norm $\|\Delta X\|_{\text{rms}} = (1/n)(\text{Tr}\,\Delta X^\dagger \Delta X)^{1/2}$ versus the number of experiments per input selected from the set (Equation 31.41).* The results shown are from simulations described in the caption.

* The number of experiments per input/measurement configuration here is chosen uniformly. An optimal (nonuniform) choice which minimizes the Cramér–Rao lower bound can be cast as a convex optimization problem [83].

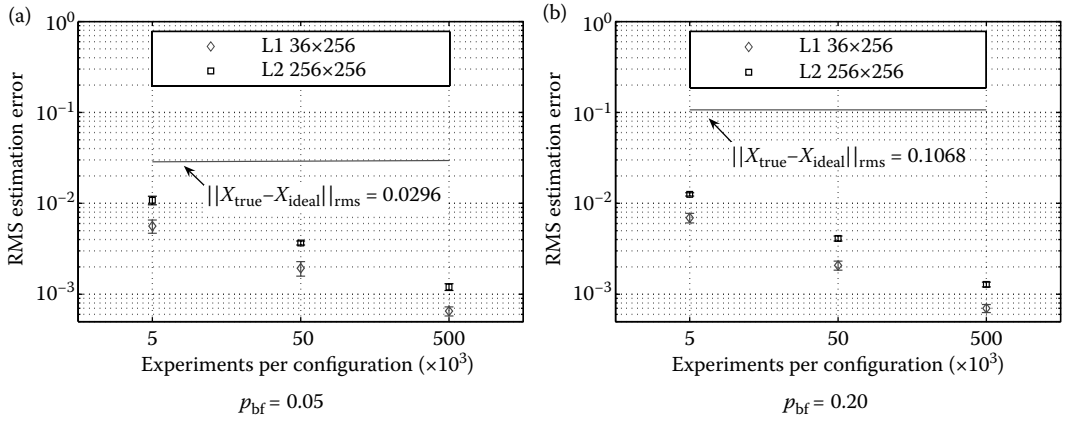

FIGURE 31.18 RMS estimation error $\|X_{\text{true}} - X_{\text{est}}\|_{\text{rms}}$ versus number of experiments per configuration: selected columns of Equation 31.41. Error bars show the deviation from 50 runs at each setting. *ℓ_2-minimization* (□): $X_{\text{est}} = X_{\ell_2}$ is from Equation 31.36 using all 16 input/output combinations. This gives a matrix $\mathcal{G} \in \mathbf{C}^{256\times256}$ as defined in Equation 31.38 which is full rank, that is, $rank(\mathcal{G}) = 256$. *ℓ_1-minimization* (◇): $X_{\text{est}} = X_{\ell_1}$ is from Equation 31.39 using six inputs and six measurements obtained from the columns of the second matrix in Equation 31.41. This gives $\mathcal{G} \in \mathbf{C}^{36\times256}$ which is full rank, that is, $rank(\mathcal{G}) = 36$.

The benefit of ℓ_1-minimization compared to the standard ℓ_2-minimization is seen most clearly with small amounts of data from highly incomplete measurements. For example, for $p_{\text{bf}} = 0.05$ [Figure 31.18a], at 50×10^3 experiments per input for the 6-input/6-output configuration ($\mathcal{G} \in \mathbf{C}^{36\times256}$) the ℓ_1 RMS estimation error is 0.0019. Compare this to the ℓ_2 error of 0.0012 at 500×10^3 experiments per input for the 16-input/16-output configuration ($\mathcal{G} \in \mathbf{C}^{256\times256}$). The latter improvement can be attributed mostly to the 10-fold increase in the number of experiments per input. The additional resources to achieve this are significant, that is, 16 inputs for ℓ_2 versus 6 for ℓ_1, and additionally, an increase in the *total* number of experiments from $6 \times 50 \times 10^3$ to $16 \times 500 \times 10^3$. It is certainly not intuitive that to estimate the 240 parameters of the process matrix, the clearly incomplete set of measurements using only 36 outcomes (◇ in Figure 31.18) could produce results not only similar to, but also for each number of experiments per input, even better than the full input case with all 256 combinations of inputs and measurements (□ in Figure 31.18). As seen, the ℓ_1 error is about half the ℓ_2 error. Also, reweighting reduced the (unweighted) ℓ_1 error by 1/2 to 1/3.

Comparing the estimation errors with the error between the actual and ideal (solid lines in Figure 31.18) suggests that at least 50×10^3 experiments per input are needed to achieve a sufficient post-QPT error correction toward the ideal unitary. Figure 31.18 also reveals that the estimation errors are very similar for both levels of bit-flip error, $p_{\text{bf}} \in \{0.05, 0.20\}$. This is explained by the Cramér–Rao bound, which defines the asymptotic error of any unbiased estimator, that is, the RMS decays as $\Delta/\sqrt{N}$. Here, Δ is effectively the error between the empirical and actual probability outcomes, which by definition is of the order one; this provides a reasonable fit to the data in Figure 31.18.

31.4.2.5 Experiment Design for QPT

The setup for OED here is entirely analogous to that of state estimation (QST). Let X^{surr} be a surrogate for the true process matrix, X^{true}. The associated (relaxed) OED problem is

$$\begin{array}{ll} \text{minimize} & V(\lambda, X^{\text{surr}}) = \text{Tr}\left[\sum_\gamma \lambda_\gamma G_\gamma(X^{\text{surr}})\right]^{-1} \\ \text{subject to} & \sum_\gamma \lambda_\gamma = 1, \;\; \lambda_\gamma \geq 0, \;\; \gamma = 1, \ldots, n_{\text{cfg}} \end{array} \tag{31.43}$$

where

$$G_\gamma(X^{\text{surr}}) = C_{\text{eq}}^\dagger \left[\sum_\alpha \frac{a_{\alpha\gamma} a_{\alpha\gamma}^\dagger}{p_{\alpha\gamma}(X^{\text{surr}})} \right] C_{\text{eq}}, \quad a_{\alpha\gamma} = \mathbf{vec}\, R_{\alpha\gamma} \in \mathbf{C}^{n^4} \tag{31.44}$$

and $C_{\text{eq}} \in \mathbf{C}^{n^4 \times n^4 - n^2}$ is part of the unitary matrix $W = [C\ C_{\text{eq}}] \in \mathbf{C}^{n^4 \times n^4}$ in the SVD of the $n^2 \times n^4$ matrix

$$[a_1 \ldots a_{n^4}] = U \left[\sqrt{n} I_{n^2} \ \ 0_{n^2 \times n^4 - n^2} \right] W^\dagger \tag{31.45}$$

with $a_k = \mathbf{vec}(B_i^\dagger B_j) \in \mathbf{C}^{n^2}$ for $k = i + (j-1)n^2,\ i, j = 1, \ldots, n^2$. The columns of C_{eq}, that is, the last $n^4 - n^2$ columns of W, are a basis for the null-space of $[a_1 \ldots a_{n^4}]$.

31.4.3 Hamiltonian Parameter Estimation

31.4.3.1 ML Hamiltonian Parameter Estimation

The quantum system is modeled by a finite dimensional Hamiltonian matrix $H(t, \theta) \in \mathbf{C}^{n \times n}$, having a known dependence on time t, $0 \le t \le t_f$, and on an unknown parameter vector $\theta \in \mathbf{R}^{n_\theta}$. The model density matrix depends on θ and the initial (prepared and known) state drawn from the set of states $\left\{ \rho_\beta^{\text{init}} \in \mathbf{C}^{n \times n} \mid \beta = 1, \ldots, n_{\text{in}} \right\}$. Thus, the density matrix associated with initial state ρ_β^{init} is $\rho_\beta(t, \theta) \in \mathbf{C}^{n \times n}$ which evolves according to

$$i\hbar \dot{\rho}_\beta = [H(t, \theta), \rho_\beta],\ \rho_\beta(0, \theta) = \rho_\beta^{\text{init}}. \tag{31.46}$$

Equivalently,

$$\rho_\beta(t, \theta) = U(t, \theta) \rho_\beta^{\text{init}} U(t, \theta)^* \tag{31.47}$$

where $U(t, \theta) \in \mathbf{C}^{n \times n}$ is the unitary propagator associated with $H(t, \theta)$ which satisfies

$$i\hbar \dot{U} = H(t, \theta) U,\ U(0, \theta) = I_n. \tag{31.48}$$

At each of the n_{sa} sample times at a time interval of duration t_{f}, measurements are recorded from identical repeated experiments. Specifically, let $\{ t_\tau \mid \tau = 1, \ldots, n_{\text{sa}} \}$ denote the sample times relative to the start of each experiment. Let $n_{\alpha\beta\tau}$ be the number of times the outcome α is recorded at t_τ with initial state ρ_β^{init} from $\ell_{\beta\tau}$ experiments. The data set thus consists of all the outcome counts

$$D = \left\{ n_{\alpha\beta\tau} \mid \alpha = 1, \ldots, n_{\text{out}},\ \beta = 1, \ldots, n_{\text{in}},\ \tau = 1, \ldots, n_{\text{sa}} \right\}. \tag{31.49}$$

The *configurations* previously enumerated and labeled by $\gamma = 1, \ldots, n_{\text{cfg}}$ are in this case combinations of input states ρ_β^{init} and sample times τ, thus $n_{\text{cfg}} = n_{\text{in}} n_{\text{sa}}$. For the POVM M_α, the model outcome probability per configuration pair $(\rho_\beta^{\text{in t}},\ t_\tau)$ is

$$\begin{aligned} p_{\alpha\beta\tau}(\theta) &= \text{Tr}\, M_\alpha \rho_\beta(t_\tau, \theta) = \text{Tr}\, O_{\alpha\tau}(\theta) \rho_\beta^{\text{init}} \\ O_{\alpha\tau}(\theta) &= U(t_\tau, \theta)^* M_\alpha U(t_\tau, \theta). \end{aligned} \tag{31.50}$$

The ML estimate, $\theta^{\text{ML}} \in \mathbf{R}^{n_\theta}$, is obtained as the solution to the optimization problem

$$\begin{array}{ll} \text{minimize} & L(D, \theta) = -\sum_{\alpha,\beta,\tau} n_{\alpha\beta\tau} \log \text{Tr}\, O_{\alpha\tau}(\theta) \rho_\beta^{\text{init}} \\ \text{subject to} & \theta \in \Theta \end{array} \tag{31.51}$$

where Θ is a set of constraints on θ. For example, it may be known that θ is restricted to a region near a nominal value, for example, $\Theta = \{ \theta \mid \|\theta - \theta_{\text{nom}}\| \le \delta \}$. Although this latter set is convex, unfortunately, the likelihood function, $L(D, \theta)$, is not guaranteed to be convex in θ. It is possible, however, that it is convex in the restricted region Θ, for example, if δ is sufficiently small.

31.4.3.2 Experiment Design for Hamiltonian Parameter Estimation

Despite the fact that Hamiltonian parameter estimation is not convex, the (relaxed) experiment design problem is convex. A direct application of the Cramér–Rao bound to the likelihood function in Equation 31.51 results in the following.

31.4.3.2.1 Hamiltonian Parameter Estimation Variance Lower Bound

For $\ell = [\ell_1 \ldots \ell_{n_{\text{cfg}}}]$ experiments per configuration $(\rho_\beta^{\text{init}}, t_\tau)$, suppose $\widehat{\theta}(\ell) \in \mathbf{R}^{n_\theta}$ is an unbiased estimate of $\theta^{\text{true}} \in \mathbf{R}^{n_\theta}$. Under these conditions, the estimation error variance satisfies

$$\mathbb{P}\, \|\widehat{\theta}(\ell) - \theta^{\text{true}}\|^2 \geq V(\ell, \theta^{\text{true}}) = \text{Tr}\, G(\ell, \theta^{\text{true}})^{-1} \tag{31.52}$$

where

$$\begin{aligned} G(\ell, \theta^{\text{true}}) &= \sum_{\beta,\tau} \ell_{\beta\tau} G_{\beta\tau}(\theta^{\text{true}}) \in \mathbf{R}^{n_\theta \times n_\theta} \\ G_{\beta\tau}(\theta^{\text{true}}) &= \sum_{\alpha} \left[\frac{\left[\nabla_\theta\, p_{\alpha\beta\tau}(\theta)\right]\left[\nabla_\theta\, p_{\alpha\beta\tau}(\theta)\right]^T}{p_{\alpha\beta\tau}(\theta)} - \nabla_{\theta\theta}\, p_{\alpha\beta\tau}(\theta) \right]\Bigg|_{\theta=\theta^{\text{true}}} \in \mathbf{R}^{n_\theta \times n_\theta}. \end{aligned} \tag{31.53}$$

The relaxed experiment design problem with respect to the surrogate $\widehat{\theta}$ for θ^{true} is

$$\begin{array}{ll} \text{minimize} & V(\lambda, \widehat{\theta}) = \text{Tr}\left[\sum_{\beta,\tau}\ \lambda_{\beta\tau} G_{\beta\tau}(\widehat{\theta})\right]^{-1} \\ \text{subject to} & \sum_{\beta,\tau}\ \lambda_{\beta\tau} = 1, \quad \lambda_{\beta\tau} \geq 0,\ \forall\, \beta,\ \tau \end{array} \tag{31.54}$$

with optimization variables $\lambda_{\beta\tau}$, the distribution of experiments per configuration $(\rho_\beta^{\text{in t}},\ t_\tau)$. The difference between this and the previous formulation is that there are no equality constraints on the parameters. The gradient $\nabla_\theta\, p_{\alpha\beta\tau}(\theta)$ and Jacobian $\nabla_{\theta\theta}\, p_{\alpha\beta\tau}(\theta)$ are dependent on the parametric structure of the Hamiltonian $H(t, \theta)$.

A two-parameter system was explored in depth in [87]. As described there, and shown in Figure 31.19, the ML estimate for both the optimal and a suboptimal configuration approaches the Fisher information bound (provided by the optimal configuration) as $N \to \infty$, the optimal configuration more rapidly approaches this bound. Furthermore, the mean-squared error (MSE) of the ML estimate is lower for the

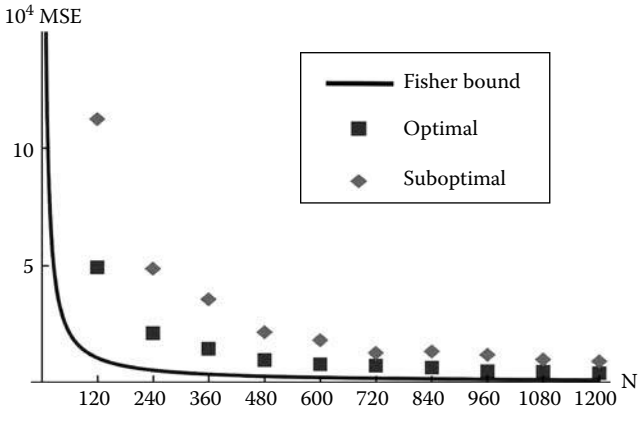

FIGURE 31.19 Plot of the MSE of the MLE estimator for the optimal (squares) and suboptimal (diamonds) configurations. Also shown (solid line) is the Fisher bound for the MSE of *any* estimator as given by the optimal experiment.

optimally configured experiments for all N. To achieve the same MSE, one must roughly perform twice as many experiments with the suboptimal configuration as are required with the optimal configuration for this particular set of guessed and actual parameters.

31.4.3.3 Indirect Adaptive Control

Hamiltonian parameter estimation can be combined with a model-based control law to form an iterative indirect adaptive control. As an example, consider the spin-coherent photon transmitter/receiver system proposed in [88,89]. This device creates quantum logic gates by manipulating electron spin via external potentials (gate voltages) to effect the *g-factors* in the semiconductor material in the presence of an external (rotating) magnetic field. Following [90, III, Ch.12-9] on models of spin systems, an idealized model of the normalized Hamiltonian in the rotating frame of a two-qubit gate under "linear g-factor control" is given by

$$\begin{aligned} H &= H_1 + H_2 + H_{12} \\ H_1 &= \frac{1}{2}\left[\varepsilon_{1z}\omega_0(Z \otimes I_2) + \varepsilon_{1x}\omega_1(X \otimes I_2)\right] \\ H_2 &= \frac{1}{2}\left[\varepsilon_{2z}\omega_0(I_2 \otimes Z) + \varepsilon_{2x}\omega_1(I_2 \otimes X)\right] \\ H_{12} &= \varepsilon_c\omega_c\left[X^{\otimes 2} + Y^{\otimes 2} + Z^{\otimes 2}\right]. \end{aligned}$$

The design goal is to use the 5 controls (ε_{1z}, ε_{1x}, ε_{2z}, ε_{2x}, ε_c) to make the Bell transform logic gate [45]

$$U_{\text{bell}} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}.$$

One of the many possible decompositions of the Bell transform is the following:

$$U_{\text{bell}} = (U_{\text{had}} \otimes I_2)\sqrt{U_{\text{swap}}}\,(X^{-1/2} \otimes X^{1/2})\sqrt{U_{\text{swap}}}\,(I_2 \otimes X).$$

Each operation in this sequence uses only the single qubit and swap "gates" produced by simultaneously pulsing the 5 controls as shown in Figure 31.20:

ε_{1z}	ε_{1x}	ε_{2z}	ε_{2x}	ε_c	Δt	gate
0	0	0	1	0	$\frac{\pi}{\omega_1}$	$-iI_2 \otimes X$
0	0	0	0	1	$\frac{\pi}{8\omega_c}$	$e^{-i\frac{\pi}{8}}\sqrt{U_{\text{swap}}}$
0	0	0	1	0	$\frac{\pi}{2\omega_1}$	$e^{-i\frac{\pi}{4}}I_2 \otimes X^{1/2}$
0	1	0	0	0	$\frac{3\pi}{2\omega_1}$	$e^{-i\frac{3\pi}{4}}X^{-1/2} \otimes I_2$
0	0	0	0	1	$\frac{\pi}{8\omega_c}$	$e^{-i\frac{\pi}{8}}\sqrt{U_{\text{swap}}}$
$\frac{\omega_{\text{had}}}{\omega_0\sqrt{2}}$	$\frac{\omega_{\text{had}}}{\omega_1\sqrt{2}}$	0	0	0	$\frac{\pi}{\omega_{\text{had}}}$	$-iU_{\text{had}} \otimes I_2$

FIGURE 31.20 Pulse control table.

The resulting gate at the final time, t_f, is U_{bell} to within a scalar phase

$$U(t_f) = e^{-i\frac{\pi}{4}}\, U_{\text{bell}}, \quad t_f = \left[\frac{3}{\omega_1} + \frac{1}{4\omega_c} + \frac{1}{\omega_{\text{had}}}\right]\pi.$$

Suppose the only unknown parameter is ω_1. Consider the following basic iteration between estimation and control:

$$\begin{array}{ll} \text{Control design} & \varepsilon^{(i)} = \bar{\varepsilon}(\widehat{\omega}_1^{(i)}), \quad t_f{}^{(i)} = t_f(\widehat{\omega}_1^{(i)}) \\ \text{Estimation} & \widehat{\omega}_1^{(i+1)} = \arg\min_{\omega_1} \mathbb{P}\, L(\omega_1,\, \varepsilon^{(i)}) \end{array}$$

where the control design function $\bar{\varepsilon}(\widehat{\omega}_1^{(i)})$ represents the pulse design from the above table (Figure 31.20), and where the average likelihood function follows from the description in Section 31.4.3.1 with the following parameters:

$$\begin{array}{ll} \text{Single initial state} & \rho^{\text{init}} = |0\rangle\langle 0| \;\; (\beta = 1) \\ \text{POVM} & M_1 = |0\rangle\langle 0|, \quad M_2 = |1\rangle\langle 1| \;\; (n_{\text{out}} = 2) \\ \text{Sample times} & \text{either } \{t_f(\widehat{\omega}_1), \quad (n_{\text{sa}} = 1)\} \text{ or } \left\{t_f(\widehat{\omega}_1)/2, \quad t_f(\widehat{\omega}_1), \; (n_{\text{sa}} = 2)\right\}. \end{array}$$

Using Hamiltonian parameters ($\omega_0^{\text{true}} = 1$, $\omega_1^{\text{true}} = 0.01$, $\omega_c^{\text{true}} = 0.01$), Figure 31.21 shows $\mathbb{P}\; L(\widehat{\omega}_1, n_{\text{sa}} = 1)$ versus $\widehat{\omega}_1/\omega_i^{\text{true}}$ for sequences of adaptive iterations using the estimate $\widehat{\omega}_1$ obtained from a local hill-climbing algorithm, that is, the local maximum of the average likelihood function is obtained. The estimation is followed by a control using the estimated value in the pulse control table. In the two cases shown, the algorithm converges to the true value.

Although not shown, the algorithm does not converge from all initial values of $\widehat{\omega}_1$. Figure 31.21c shows $\|U(t_f(\widehat{\omega}_1),\; \bar{\varepsilon}(\widehat{\omega}_1)) - U_{\text{des}}\|_{\text{frob}}$ versus estimate $\widehat{\omega}_1/\omega_1^{\text{true}}$ with the control from the table. The function is clearly not convex. The region of convergence for $n_{\text{sa}} = 1$ is $0.9 \le \widehat{\omega}_1/\omega_1^{\text{true}} \le 1.3$, for $n_{\text{sa}} = 2$, $0.6 \le \widehat{\omega}_1/\omega_1^{\text{true}} \le 2.5$. In neither case will the algorithm converge for $\widehat{\omega}_1/\omega_1^{\text{true}} < 0.6$. These results, of course, are specific to this example and cannot be generalized. To reiterate, conditions for convergence, region of attraction, and so on, are only partially understood, in general, for this type of iteration [91].

31.5 Optimal Quantum Feedback Control

31.5.1 Quantum Linear Systems

In the remaining sections, we look at feedback control for a special class of quantum systems. The models are expressed in the Heisenberg picture, where they are described by equations that look formally like the standard classical linear systems models. However, these equations are for noncommuting quantum operators and are driven by noncommuting quantum noise sources. The matrices occurring in the equations are not arbitrary—they must satisfy certain constraints if they are to describe quantum systems. Specifically, we consider quantum linear systems describing an assembly of open harmonic oscillators, such as an interconnection of cavities. These systems are described by quantum stochastic differential equations with quantum w and classical u inputs, and an output y available to the controller

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t), \quad x(0) = x, \\ y(t) &= Cx(t)dt + Dw(t) \end{aligned} \tag{31.55}$$

where A, B_1, B_2, C, and D are, respectively, real $2n \times 2n$, $2n \times 2p$, $2n \times m$, $2p \times 2n$, and $2p \times 2p$ matrices, and $x(t) = (x_1(t), \ldots, x_{2n}(t))^T$ is a vector of system variables. The initial system variables $x(0) = x$ are

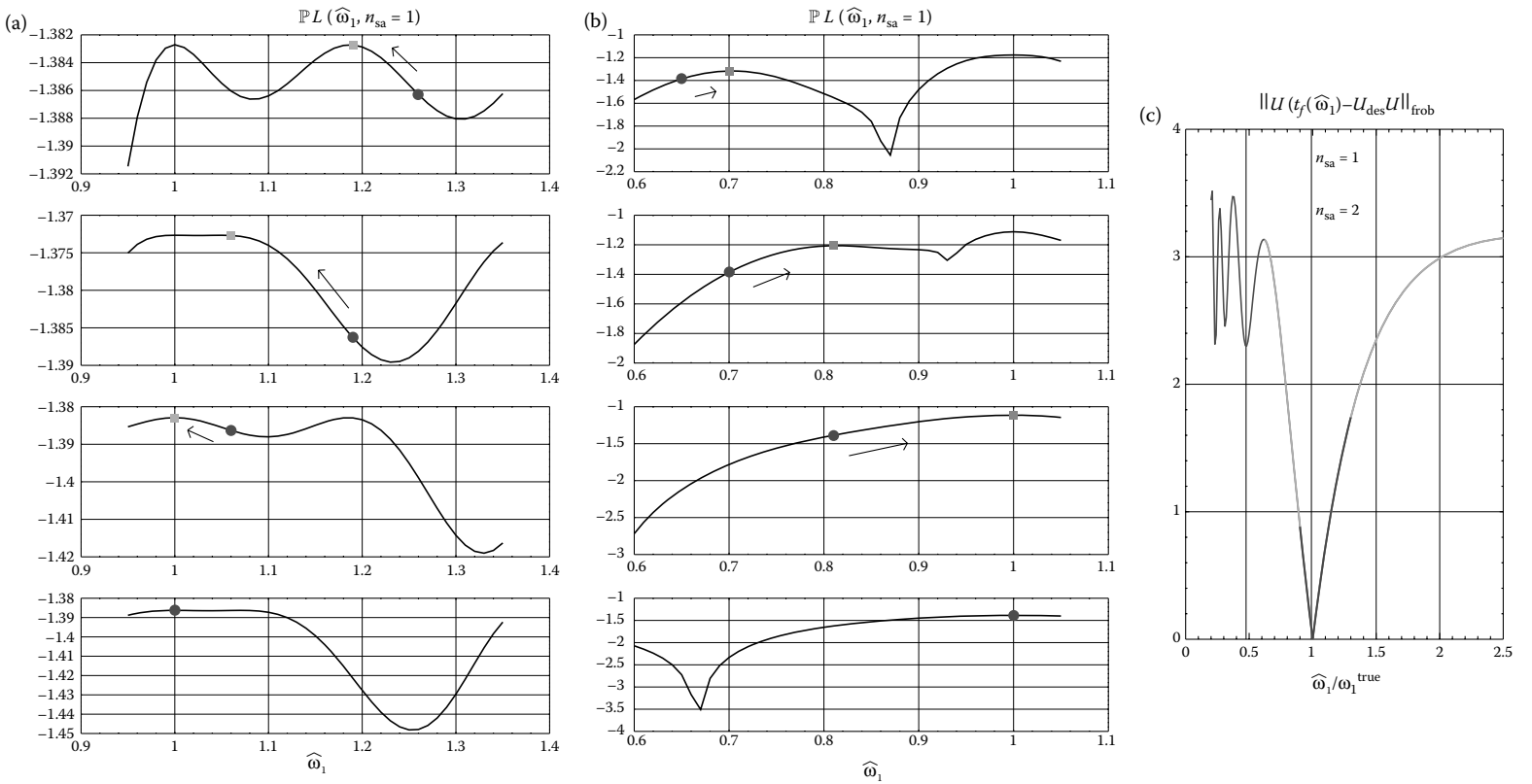

FIGURE 31.21 Plots (a) and (b): Iterative adaptation for $n_{sa} = 1$ at $t_f(\widehat{\omega}_1)$ for two starting values of $\widehat{\omega}_1$. Plot (c): Regions of convergence.

Gaussian with state ρ, and satisfy the commutation relations

$$[x_j, x_k] = i\Theta_{jk}, \quad j,k = 1,\ldots,n \tag{31.56}$$

where Θ is the real antisymmetric matrix

$$\Theta = J_n = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \tag{31.57}$$

where I_n is the $n \times n$ identity matrix. The signal $w(t)$ is a vector of self-adjoint stochastic processes with covariance

$$\langle w(t)w(t')^T \rangle = F\delta(t-t') \tag{31.58}$$

where F is the nonnegative Hermitian matrix $F = I_{2p} + iJ_{2p}$. The input $u(t)$ is a vector of classical processes (self-adjoint).

As mentioned above, the linear system (Equation 31.55) looks formally like a classical linear system. The essential difference is that the vectors $x(t)$, $w(t)$, and $y(t)$ have components that do not commute. For arbitrary matrices A, B_1, B_2, C, and D, the linear system (Equation 31.55) need not necessarily correspond to a quantum system, and so it is important to know when Equation 31.55 does in fact describe a quantum physical system.

The open quantum systems that give rise to Heisenberg equations of the type (Equation 31.55) are those for which the S-system (recall Section 31.2.3) is a collection of harmonic oscillators (Section 31.2.5), and the E-system is the free field (Section 31.2.6). The unitary dynamics of the SE-system means that we require $x(t) = U^\dagger(t)xU(t)$ and $y(t) = U^\dagger(t)w(t)U(t)$, where $U(t) = U_{SE}(t)$ is a unitary operator solving Schrodinger's equation (in Stratonovich form)

$$\dot{U}(t) = \{Lb_{in}^\dagger(t) - b_{in}(t)L^\dagger - iH\}U(t), \quad U(0) = I \tag{31.59}$$

where $w(t) = (b_{in,r}^T(t), b_{in,i}^T(t))^T$, $L = Mx$, $H = \frac{1}{2}(x^T Rx + x^T Su + u^T S^T x)$, and R is real symmetric and S is real. If the system is specified by these parameters, then the matrices A, B_1, B_2, C, and D are given by

$$\begin{aligned} &A = J_n(R + \frac{1}{2i}(M^\dagger M - M^T M^\#)), \quad B_1 = J_n(i(M^T - M^\dagger), (M^T + M^\dagger)), \quad B_2 = J_n S, \\ &C = \begin{bmatrix} M + M^\# \\ -i(M - M^\#) \end{bmatrix}, \quad D = I \end{aligned} \tag{31.60}$$

and $D = I$.

As alluded to in Sections 31.2.5 and 31.2.7 [25], Equations 31.55 describe a quantum physical system in this sense if and only if the matrices A, B_1, B_2, C, and D are real and satisfy the following conditions:

$$AJ_n + J_n A^T + B_1 J_p B_1^T = 0 \tag{31.61}$$

$$B_1 = \frac{1}{2}J_n C^T J_p, \tag{31.62}$$

$$D = I. \tag{31.63}$$

Indeed, if A, B_1, B_2, C, and D satisfy these conditions, then $R = \frac{1}{2}(-J_n A + A^T J_n)$, $M = \frac{1}{2}[I \; iI]C$, and $S = -J_n B_2$.

The optical cavity described in Section 31.2.7 is an example of a linear quantum system.

31.5.2 Quantum Filtering

It is often of interest to monitor an observable $y_0(t)$ of the output field $y(t)$, as this information may be used to make inferences about the system, or for controlling the system, see Figure 31.22.

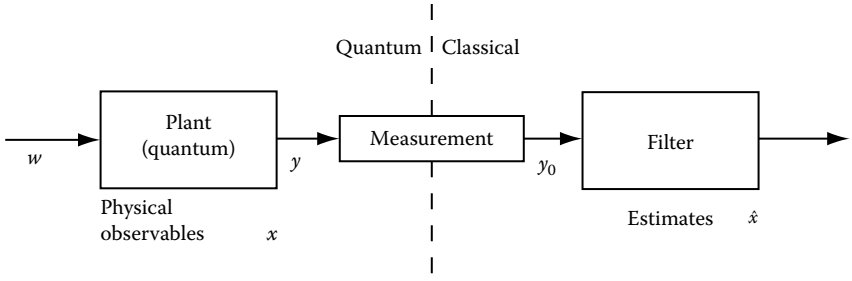

FIGURE 31.22 Filtering a measurement signal $y_0(\cdot)$.

In quantum optics, such measurements are made using photodetection systems; for instance, homodyne detection is used to (approximately) measure the real quadrature $y_0(t) = [I_p\ 0]y(t)$, the photocurrent produced by the detector. Note that $[y_{0,j}(s), y_{0,k}(t)] = 0$ for all j, k and times s, t, and so the components of $y_0(s)$, $0 \le s \le t$ form a commuting collection of observables. The measurement signal is given by

$$y_0(t) = C_0 x(t) + w_0(t) \tag{31.64}$$

where $C_0 = [I_p\ 0]C$ and $w_0(t) = [I_p\ 0]w(t)$ is a standard classical Wiener process.

The measurement signal $y_0(\cdot)$ may be filtered to produce estimates $\widehat{x}(t)$ of the system variables $x(t)$ using the *quantum filter*, developed by Belavkin [12,13,42], and independently by Carmichael [70] (who used the terms stochastic master equation and quantum trajectories), building on the theory of open quantum systems developed during the 1970s and 1980s. The quantum filter, which computes the conditional state π_t for a quantum system, was developed for a more general class of open quantum systems than the linear systems considered here, and can be viewed as a quantum generalization of the equations of classical nonlinear filtering. In the case of linear gaussian quantum systems, the quantum filter reduces to a Kalman filter.

Now $x(t)$ commutes with $y_0(s)$, $0 \le s \le t$, and so the quantum conditional expectation $\widehat{x}(t) = \mathbb{P}[x(t) \mid y_0(s), 0 \le s \le t] = \mathrm{Tr}[\pi_t x]$ is well defined, [12,13,42,75]. The time evolution of $\widehat{x}(t)$ is given by the quantum filter

$$\dot{\widehat{x}}(t) = A\widehat{x}(t) + B_2 u(t) + (Y(t)C_0^T + B_1[I\ 0]^T)(y_0(t) - C_0\widehat{x}(t)) \tag{31.65}$$

where

$$\dot{Y}(t) = AY(t) + Y(t)A^T + B_1B_1^T - (Y(t)C_0^T + B_1[I\ 0]^T)(Y(t)C_0^T + B_1[I\ 0]^T)^T. \tag{31.66}$$

The conditional mean $\widehat{x}(t)$ and symmetrized conditional covariance $Y(t) = \frac{1}{2}\mathbb{P}[x(t)x^T(t) + (x(t)x^T(t))^T]$ parameterize the conditional state π_t.

Example: Monitoring an Atom in a Cavity

Consider the problem of measurement feedback control of an atom trapped inside an optical cavity, as shown in Figure 31.7 [17]. A magnetic field and a second optical beam create a spatially dependent force within the cavity which provides a trapping mechanism which can be adjusted by classical control signals. The control objective is to cool and confines the atom, that is, to reduce the atom's momentum and to spatially localize the atom. Here, we present the quantum filter for this problem, and look at control aspects in the following sections.

A linear quadratic Gaussian (LQG) approximation gives a linear quantum model for the trapped atom with parameters

$$R = \begin{bmatrix} m\omega^2 & 0 \\ 0 & \frac{1}{m} \end{bmatrix}, \quad S = \begin{bmatrix} 0 & -l_2 \\ l_1 & 0 \end{bmatrix}, \quad M = [\sqrt{2k}\ 0]$$

determining the Hamiltonian and coupling operator (recall Section 31.5.1). Here, m is the mass of the atom, ω is the natural frequency, and k is a coupling strength parameter. The matrices A, B_1, B_2, C, and D are given by

$$A = \begin{bmatrix} 0 & \frac{1}{m} \\ -m\omega^2 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 \\ 0 & -\sqrt{2k} \end{bmatrix}, \quad B_2 = \begin{bmatrix} l_1 & 0 \\ 0 & l_2 \end{bmatrix}, \quad C = \begin{bmatrix} 2\sqrt{2k} & 0 \\ 0 & 0 \end{bmatrix}, \quad D = I.$$

The vector $x = (q, p)^T$ is formed from the position q and momentum p observables for the atom. It can be seen that the uncontrolled mean motion is oscillatory.

The measurement signal $y_0(\cdot)$ is given by Equation 31.64, where $C_0 = [l_p\ 0] C = [2\sqrt{2k}\ 0]$. The vector of estimates $\widehat{x} = [\widehat{q}, \widehat{p}]^T$ evolves according to the quantum (Kalman) filter

$$\begin{aligned} \dot{\widehat{q}}(t) &= \frac{\widehat{p}(t)}{m} + l_1 u_1(t) + 2\sqrt{2k}\, Y_{11}(t)(y_0(t) - 2\sqrt{2k}\,\widehat{q}(t)) \\ \dot{\widehat{p}}(t) &= -m\omega^2 \widehat{q}(t) + l_2 u_2(t) + 2\sqrt{2k}\, Y_{12}(t)(y_0(t) - 2\sqrt{2k}\,\widehat{q}(t)) \end{aligned} \tag{31.67}$$

where the components of the conditional covariance

$$Y(t) = \begin{bmatrix} Y_{11}(t) & Y_{12}(t) \\ Y_{12}(t) & Y_{22}(t) \end{bmatrix}$$

satisfy

$$\begin{aligned} \dot{Y}_{11}(t) &= \frac{2Y_{12}(t)}{m} - 8kY_{11}^2(t) \\ \dot{Y}_{12}(t) &= \frac{Y_{22}(t)}{m} - m\omega^2 Y_{11}(t) - 8kY_{11}(t)Y_{12}(t) \\ \dot{Y}_{22}(t) &= -2m\omega^2 Y_{12}(t) + 2k - 8kY_{12}^2(t). \end{aligned} \tag{31.68}$$

31.5.3 Quantum Measurement Feedback LQG Control

Kalman's LQG control is one of the most significant developments in control theory [92]. In this section, we explain how measurement feedback LQG [measurement feedback (MF)-quantum linear quadratic Gaussian (QLQG)] works for the quantum linear system (Equation 31.55) [17,75]. An experimental demonstration of the application of LQG control in quantum optics is described in [20].

We begin by stating what we mean by a measurement feedback controller K for the quantum linear system (Equation 31.55). Such a controller K is to process the measurement signal $y_0(\cdot)$ (discussed in Section 31.5.2) in a causal manner to produce the classical control signal $u(\cdot)$. Thus, for each time $t \geq 0$, $u(t) = K_t(y_0(s), 0 \leq s \leq t)$, Figure 31.23.

The LQG performance criterion is defined by

$$J(K) = \mathbb{P}\left[\int_0^T \left[\frac{1}{2}x^T(t)Px(t) + \frac{1}{2}u^T(t)Qu(t)\right]dt + \frac{1}{2}x^T(T)X_T x(T)\right] \tag{31.69}$$

where P, Q, and X_T are symmetric nonnegative definite matrices. The problem is to minimize $J(K)$ over the class of measurement feedback controllers.

To solve this problem, we recall that the quantum conditional expectation $\mathbb{P}[x^T(t)Px(t) \mid y_0(s), 0 \leq s \leq t]$ is well-defined since $x(t)$ commutes with the commutative family $y_0(s), 0 \leq s \leq t$ of observables,

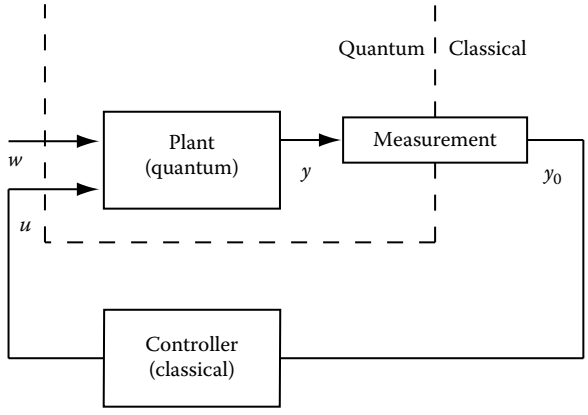

FIGURE 31.23 Measurement feedback control for a quantum linear system.

and, moreover, satisfies the fundamental property $\mathbb{P}[\mathbb{P}[x^T(t)Px(t) \mid y_0(s), 0 \le s \le t]] = \mathbb{P}[x^T(t)Px(t)]$. Furthermore, $\mathbb{P}[x^T(t)Px(t) \mid y_0(s), 0 \le s \le t] = \widehat{x}^T(t)P\widehat{x}(t) + \mathrm{Tr}[PY(t)]$. We may then re-express $J(K)$ as an equivalent classical control problem in terms of the quantum filter:

$$J(K) = \mathbf{P}\left[\int_0^T \left[\frac{1}{2}\widehat{x}^T(t)P\widehat{x}(t) + \frac{1}{2}u^T(t)Qu(t)\right]dt + \frac{1}{2}\widehat{x}^T(T)X_T\widehat{x}(T)\right] + \alpha \tag{31.70}$$

where α is a constant independent of the controller.

This equivalent problem may be solved using standard methods from the classical stochastic control theory [93]. The optimal control is given by

$$u^\star(t) = K_t^\star(y_0(s), 0 \le s \le t) = -Q^{-1}B_2^T X(t)\widehat{x}(t) \tag{31.71}$$

where

$$-\dot{X}(t) = X(t)A + A^T X(t) + P - X(t)B_2Q^{-1}B_2^T X(t), \quad X(T) = X_T. \tag{31.72}$$

Note that the solution to this MF-QLQG problem is identical to the standard classical LQG solution (Figure 31.24).

This controller has the *separation structure*, with filter dynamics the quantum filter for the conditional state π_t (in terms of $\widehat{x}(t)$ and $Y(t)$), which serves as an information state [93].

Example: Controlling an Atom in a Cavity

Consider again the problem of cooling and confining the trapped atom, as discussed in Section 31.5.2. In [17], LQG control was used to solve this problem. If we select the cost matrices $P = R$,

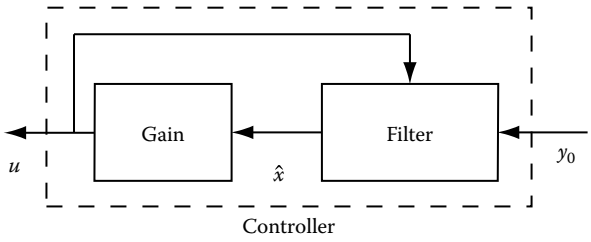

FIGURE 31.24 Structure of the optimal MF-QLQG controller (Equations 31.65 and 31.71).

$Q = \kappa^2 R$, and let $l_1 = l_2 = 1$, we find that the optimal steady-state feedback control (Equation 31.71) is given by

$$u^\star(t) = -\begin{bmatrix} \frac{1}{\kappa} & 0 \\ 0 & \frac{1}{\kappa} \end{bmatrix} \begin{bmatrix} \widehat{q}(t) \\ \widehat{p}(t) \end{bmatrix}. \tag{31.73}$$

With this feedback in place, the conditional dynamics (Equation 31.67) has asymptotically stable mean motion, as required by successful cooling and confinement. Steady-state fluctuations persist and may be characterized by the steady-state covariances.

31.5.4 Quantum Measurement Feedback LEQG Control

The classical *linear exponential quadratic Gaussian* (LEQG), or *risk-sensitive*, control problem was formulated in [94] by taking the exponential of the integral in the cost criterion before taking the expectation. It was not possible to solve the measurement feedback LEQG problem using the Kalman filter in general, and the solution tuned out to need a filter that includes terms from the cost [95–97]. This surprising result has important implications and is related to game theory and robust control. The measurement feedback risk-sensitive problem was extended to the quantum context [23,76,77]. In the section, we discuss the measurement feedback quantum LEQG (MF-QLEQG) problem. As we shall see, the solution is not the same as the classical LEQG solution.

The LQG performance criterion (Equation 31.69) is defined as the expectation of an integral of terms that do not commute. In order to define the exponential of the integral, we introduce the observable $R(t)$ defined as the solution to the differential equation

$$\dot{R}(t) = \frac{\mu}{2}\left[\frac{1}{2}x^T(t)Px(t) + \frac{1}{2}u^T(t)Qu(t)\right]R(t), \quad R(0) = I \tag{31.74}$$

where $\mu > 0$ is a real risk parameter. Thus, $R(t)$ is a time-ordered exponential. The quantum MF-QLEQG performance criterion is defined to be

$$J^\mu(K) = \mathbb{P}\left[R^\dagger(T)e^{\mu\frac{1}{2}x^T(T)X_T x(T)}R(T)\right]. \tag{31.75}$$

As explained in [23,77], $J^\mu(K)$ can be re-expressed as a classical expectation of a suitably defined information state π_t^μ parameterized by quantities $\widehat{x}^\mu(t)$ and $Y^\mu(t)$ given by the equations

$$\dot{\widehat{x}}^\mu(t) = (A + \mu Y^\mu(t)P)\widehat{x}^\mu(t) + B_2 u(t) + (Y^\mu(t)C_0^T + B_1[I\ 0]^T)(y_0(t) - C_0\widehat{x}^\mu(t)) \tag{31.76}$$

and

$$\begin{aligned}\dot{Y}^\mu(t) = {} & AY^\mu(t) + Y^\mu(t)A^T + \mu Y^\mu(t)PY^\mu(t) + B_1B_1^T - \frac{\mu}{4}JPJ \\ & - (Y^\mu(t)C^T + B_1[I\ 0]^T)(Y^\mu(t)C^T + B_1[I\ 0]^T)^T.\end{aligned} \tag{31.77}$$

Equation 31.77 for the matrix $Y^\mu(t)$ contains terms depending on the cost matrix P. The term $-\frac{\mu}{4}JPJ$ arises from the fact that terms in the integrand in Equation 31.74 do not commute, and does not arise in the analogous classical LEQG problem [95]. The optimal control is given by

$$u^{\mu,\star}(t) = K_t^{\mu,\star}(y_0(s), 0 \le s \le t) = -Q^{-1}B_2^T X^\mu(t)(I - \mu Y^\mu(t)X^\mu(t))^{-1}\widehat{x}^\mu(t) \tag{31.78}$$

where

$$-\dot{X}^\mu(t) = X^\mu(t)A + A^T X^\mu(t) + P - X^\mu(t)(B_2Q^{-1}B_2^T - \mu B_1B_1^T)X^\mu(t), \quad X^\mu(T) = X_T. \tag{31.79}$$

This controller also has the separation structure (Figure 31.24 with $\widehat{x}^\mu(t)$ and $Y^\mu(t)$ replacing $\widehat{x}(t)$ and $Y(t)$).

The MF-QLEQG information state π_t^μ parameterized by $\widehat{x}^\mu(t)$ and $Y^\mu(t)$ appears to be new to quantum physics. The traditional conditional state π_t used in quantum physics can be regarded as a subjective state providing a description of the knowledge obtained by monitoring the system. The risk-sensitive state π_t^μ can be viewed as a subjective state containing knowledge obtained by monitoring but modified by the cost—it includes both knowledge and purpose, extending Bohr's interpretation of quantum mechanics in the feedback context.

Example: Monitoring an Atom in a Cavity

We again return to the trapped atom considered in Sections 31.5.2 and 31 5.3. Using the same cost matrices as above, we find that the quantum risk sensitive (or MF-QLEQG) filter is given by

$$\begin{aligned}\dot{\widehat{q}}^\mu(t) &= \frac{\widehat{p}^\mu(t)}{m} + l_1 u_1(t) + 2\sqrt{2k}\, Y_{11}(t)(y_0(t) - 2\sqrt{2k}\,\widehat{q}^\mu(t)) \\ \dot{\widehat{p}}^\mu(t) &= -m\omega^2 \widehat{q}^\mu(t) + l_2 u_2(t) + 2\sqrt{2k}\, Y_{12}(t)(y_0(t) - 2\sqrt{2k}\,\widehat{q}^\mu(t))\end{aligned} \tag{31.80}$$

where the components of the matrix

$$Y^\mu(t) = \begin{bmatrix} Y_{11}^\mu(t) & Y_{12}^\mu(t) \\ Y_{12}^\mu(t) & Y_{22}^\mu(t) \end{bmatrix}$$

satisfy

$$\begin{aligned}\dot{Y}_{11}^\mu(t) &= \frac{2Y_{12}^\mu(t)}{m} + (\mu m\omega^2 - 8k)(Y_{11}^\mu(t))^2 + \frac{\mu}{m}(Y_{12}^\mu(t))^2 + \frac{\mu}{4m} \\ \dot{Y}_{12}^\mu(t) &= \frac{Y_{22}^\mu(t)}{m} - m\omega^2 Y_{11}^\mu(t) + (\mu m\omega^2 - 8k) Y_{11}^\mu(t) Y_{12}^\mu(t) + \frac{\mu}{m} Y_{22}^\mu(t) Y_{12}^\mu(t) \\ \dot{Y}_{22}^\mu(t) &= -2m\omega^2 Y_{12}^\mu(t) + 2k + (\mu\omega^2 - 8k)(Y_{12}^\mu(t))^2 + \frac{\mu}{m}(Y_{22}^\mu(t))^2 + \frac{\mu m\omega^2}{4}.\end{aligned} \tag{31.81}$$

The optimal feedback is given by Equation 31.78.

In [77], a simulation study was carried out to compare the robustness characteristics of the MF-QLQG and MF-QLEQG controllers. Both controllers were designed on the basis of the nominal model, as described above. The physical system was assumed to be subject to a field with unmodeled coherent magnitude β. A plot of the mean MF-QLQG performance as a function of the uncertainty parameter β is shown in Figure 31 25. The vertical axis shows the integral of quadratic costs function averaged with respect to the true model. The horizontal axis is the uncertainty parameter β. When $\beta = 0$, the nominal and actual models coincide, and the MF-QLQG has lower cost, as expected (MF-QLQG is defined to minimize this cost). However, as β increases, we see that the MF-QLEQG controller achieves better performance, indicated by the curve with smaller concavity. The MF-QLEQG controller's performance degrades less rapidly and MF-QLQG with increasing uncertainty. This is consistent with expectations for a robust controller: good performance under nominal conditions, and acceptable performance in other than nominal conditions.

31.5.5 Quantum Coherent Feedback H^∞ Control

In the proceeding sections, the controllers K were classical systems driven by a classical measurement signal $y_0(\cdot)$ producing a classical control signal u. We now consider controllers K that are quantum systems driven by a fully quantum signal y_1 derived from one or more output channels of a quantum plant G. Such a quantum controller K produces a quantum signal u that is connected to an input channel

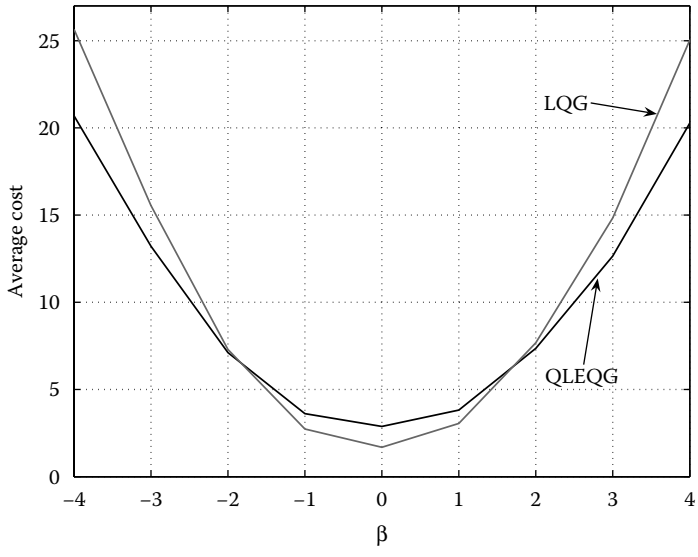

FIGURE 31.25 Performance (vertical axis) of the MF-QLQG and MF-QLEQG controllers under uncertainty (horizontal axis, β).

of the plant G, Figure 31.26. The mapping $K : y_1(\cdot) \mapsto u(\cdot)$ is a causal quantum operation, so that in general the feedback loop formed by the interconnection of P and K can preserve quantum coherence.

The plant G to be controlled is a quantum linear system of the type discussed in Section 31.5.1 expressed in the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_0 v(t) + B_1 w(t) + B_2 u(t) \\ y_1(t) &= C_1 x(t) + D_1 w(t) \end{aligned} \tag{31.82}$$

where v and w are independent and commuting quantum noises with covariances $\langle v(t)v^T(t')\rangle = F_v\delta(t - t')$ and $\langle w(t)w^T(t')\rangle = F_w\delta(t - t')$, and u is a quantum input signal. The quantum output signal y_1 is of the form $y_1 = P_f y$, for a suitable matrix P_f. Note that quantum signals like u and y_1 have self-adjoint components (in our quadrature representation), but the components do not in general commute. Thus, the output information $y_1(s), 0 \le s \le t$ is a collection of noncommuting observables in general, in contrast to the commuting observables $y_0(s), 0 \le s \le t$ discussed in Section 31.5.2. At present, there are no known

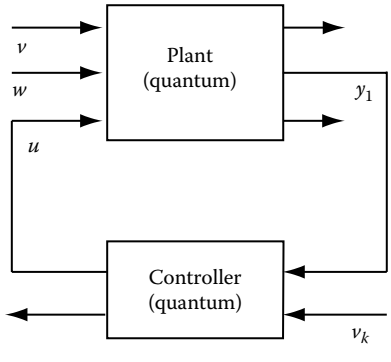

FIGURE 31.26 Coherent feedback control for a quantum linear system. The quantum fields u and y_1 are not measured, and the controller is also a quantum system.

noncommutative generalizations of the quantum filter, and indeed systematic and optimal design of fully quantum feedback systems is a major research issue. In this section, we fix the form of the controller K and determine the parameters of the controller using the performance and physical realizability criteria.

The controller K is taken to be a quantum linear system of the form

$$\begin{aligned}\dot{\xi}(t) &= A_K\xi(t) + B_{K1}v_K(t) + B_{K2}y_1(t)\\ u(t) &= C_K\xi(t) + D_Kv_K(t)\end{aligned} \tag{31.83}$$

where the matrices A_K, B_{K1}, B_{K2}, C_K, and D_K are to be determined, and v_K is a quantum noise process with covariance $\langle v_K(t)v_K^T(t')\rangle = (I+iJ)\delta(t-t')$. The vector of self-adjoint variables ξ is to have commutation relations $\xi(t)\xi^T(t) - (\xi(t)\xi^T(t))^T = i\Theta_K$, where Θ_K is a skew-symetric matrix to be determined.

Before we can specify the performance criteria, we interconnect the plant and controller to form the closed-loop system

$$\dot{\eta}(t) = \widetilde{A}\eta(t) + \widetilde{B}\widetilde{w}(t) \tag{31.84}$$

where $\widetilde{w}(t) = (v^T(t), w^T(t), v_K^T(t))^T$,

$$\widetilde{A} = \begin{bmatrix} A & BC_K \\ B_{K2}D & A_K \end{bmatrix},\ \widetilde{B} = \begin{bmatrix} B_0 & B_1 & B_2D_K \\ 0 & B_{K2}D & B_{K1} \end{bmatrix}. \tag{31.85}$$

We now specify an H^∞ or L^2 gain performance objective for the closed-loop system. Suppose that the quantum signal w has the form $w(t) = \beta_w(t) + n_w(t)$, where n_w is a quantum white-noise process with covariance $\langle n_w(t)n_w^T(t')\rangle = F_w\delta(t-t')$. Write $\beta_z = C_1x + D_{12}C_K\xi$. Given a gain bound $g > 0$ and $\epsilon > 0$, we wish to find a controller K such that for the closed-loop system (Equation 31.84) the following L^2 gain inequality is satisfied:

$$\mathbb{P}\left[\int_0^T (\beta_z^T(t)\beta_z(t) + \epsilon\eta^T(t)\eta(t))dt\right] \le (g^2-\epsilon)\mathbb{P}\left[\int_0^T \beta_w^T(t)\beta_w(t)dt\right] + \mu_1 + \mu_2 t. \tag{31.86}$$

We make the standard classical assumptions [98] (1) $D_{12}^TD_{12} = E_1 > 0$, (2) $D_{21}D_{21}^T = E_2 > 0$, (3) The matrices $\begin{bmatrix} A-j\omega I & B_2 \\ C_1 & D_{12}\end{bmatrix}$ and $\begin{bmatrix} A-j\omega I & B_1 \\ C_2 & D_{21}\end{bmatrix}$ have full rank for all $\omega \ge 0$.

The controller is expressed in terms of the following pair of algebraic Riccati equations:

$$\begin{aligned}&(A - B_2E_1^{-1}D_{12}^TC_1)^TX + X(A - B_2E_1^{-1}D_{12}^TC_1) + X(B_1B_1^T - g^2B_2E_1^{-1}B_2')X\\ &\quad + g^{-2}C_1^T(I - D_{12}E_1^{-1}D_{12}^T)C_1 = 0;\end{aligned} \tag{31.87}$$

$$\begin{aligned}&(A - B_1D_{21}^TE_2^{-1}C_2)Y + Y(A - B_1D_{21}^TE_2^{-1}C_2) + Y(g^{-2}C_1^TC_1 - C_2^TE_2^{-1}C_2)Y\\ &\quad + B_1(I - D_{21}^TE_2^{-1}D_{21})B_1^T = 0.\end{aligned} \tag{31.88}$$

The solutions to these Riccati equations are required to satisfy the following conditions: (1) $A - B_2E_1^{-1}D_{12}^TC_1 + (B_1B_1^T - g^2B_2E_1^{-1}B_2')X$ is a stability matrix, (2) $A - B_1D_{21}^TE_2^{-1}C_2 + Y(g^{-2}C_1^TC_1 - C_2^TE_2^{-1}C_2)$ is a stability matrix, and (3) the matrix XY has a spectral radius strictly less than one.

In [25], it was shown that if the Riccati equations 31.87 and 31.88 have solutions satisfying the conditions mentioned, then a controller of the form Equation 31.83 solves the H^∞ control problem under consideration if its system matrices are constructed from the Riccati solutions as follows:

$$\begin{aligned}A_K &= A + B_2C_K - B_KC_2 + (B_1 - B_KD_{21})B_1^TX;\\ B_{K2} &= (I - YX)^{-1}(YC_2^T + B_1D_{21}^T)E_2^{-1};\\ C_K &= -E_1^{-1}(g^2B_2^TX + D_{12}^TC_1).\end{aligned} \tag{31.89}$$

By [25, Theorem 5.5], the controller parameters B_{K1}, D_K, and the controller noise v_K may be chosen to give a controller that is physically realizable.

An experimental demonstration of quantum H^∞ control is reported in [29].

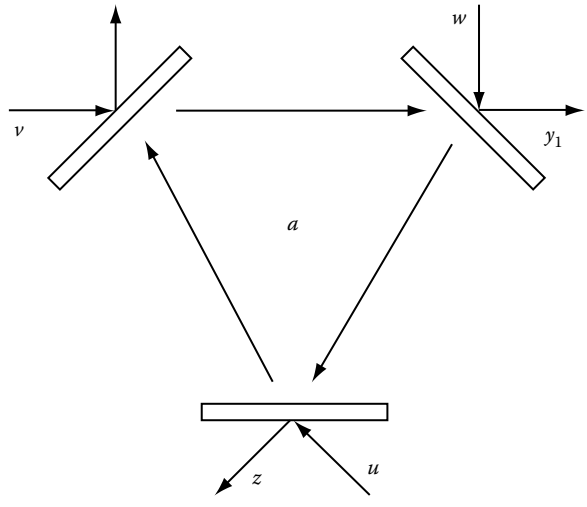

FIGURE 31.27 An optical cavity (plant).

Example: Coherent Control of an Optical Cavity

We consider an optical cavity resonantly coupled to three optical channels v, w, and u as in Figure 31.27. The control objective is to attenuate the effect of the disturbance signal w on the output z—physically this means to dim the light emerging from z resulting from light shone in at w.

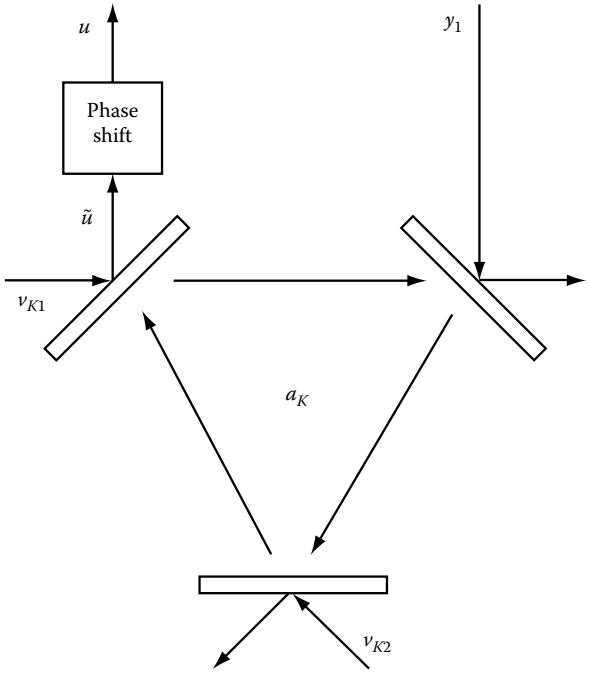

FIGURE 31.28 An optical cavity quantum realization of the controller ($\Theta_K = J$) for the plant shown in Figure 31.27.

The matrices defining the plant are

$$A = -\frac{\gamma}{2}I; \quad B_0 = -\sqrt{\kappa_1}I; \quad B_1 = -\sqrt{\kappa_2}I; \quad B_2 = -\sqrt{\kappa_3}I;$$
$$C_1 = \sqrt{\kappa_3}I; \quad D_{12} = I; \quad C_2 = \sqrt{\kappa_2}I\, D_{21} = I,$$

where $\kappa_1 = 2.6$, $\kappa_2 = \kappa_3 = 0.2$.

With a disturbance attenuation constant of $g = 0.1$, it was found that the Riccati equations 31.87 and 31.88 have stabilizing solutions satisfying the conditions stated above, namely $X = Y = 0_{2\times 2}$. The controller matrices determined from the Riccati equations are

$$A_K = -1.1I, \quad B_{K2} = -0.447I, \quad C_K = -0.447I.$$

Solving the physical realization conditions give a physical system

$$\dot{\xi}(t) = A_K\xi(t) + [B_{K1} \quad B_K][v_K^T \quad y_1^T]^T$$
$$d\widetilde{u}(t) = -C_K\xi(t)dt + [I_{2\times 2} \quad 0_{2\times 4}][v_K^T \quad y_1^T]^T$$
$$\widetilde{u}(t) = K_s u(t)$$

where $B_{K1} = [-0.447I \quad -1.342]$, $v_K(t) = (v_{K11}(t), v_{K12}(t), v_{K21}(t), v_{K22}(t))^T$ are independent quantum noises, and K_s is a 180° phase shift (Figure 31.28).

Acknowledgement

This work was partially supported by research grants from the Australian Research Council and AFOSR Grant FA2386-09-1-4089 AOARD 094089.

References

1. ARDA quantum computing roadmap. Available at http://qist.lanl.gov, 2004.
2. P. Dowling and G. J. Milburn. Quantum technology: The second quantum revolution. *Proc. R. Soc. London*, A 361:1655, 2003.
3. G. R. Fleming and M. A. Ratner. Grand challenges in basic energy sciences. *Physics Today*, 61-7:28–33, July 2008.
4. A. S. Fletcher, P. W. Shor, and M. Z. Win. Optimum quantum error recovery using semidefinite programming. *Phys. Rev. A*, 75:012338, arXiv.org: 0606035[quant-ph], 2007.
5. J. M. Geremia and H. Rabitz. Teaching lasers to optimally identify molecular hamiltonians. *Phys. Rev. Lett.*, 89, 263902, 2002.
6. R. L. Kosut, A. Shabani, and D. A. Lidar. Robust quantum error correction via convex optimization. *Phys. Rev. Lett.*, 100:020502, arXiv.org: 0703274[quant-ph], 2008.
7. M. Reimpell and R. F. Werner. Iterative optimization of quantum error correcting codes. *Phys Rev Lett* , 94:080501, 2005.
8. M. Q. Phan and H. Rabitz. Learning control of quantum-mechanical systems by laboratory identification of effective input–output maps. *Chem. Phys.*, 217:389–400, 1997.
9. M. A. Armen, K. J. Au, J. K. Stockton, A. C. Doherty, and H. Mabuchi. Adaptive homodyne measurement of optical phase. *Phys. Rev. A*, 89(13):133602, 2002.
10. V. P. Belavkin. Optimal measurement and control in quantum dynamical systems. Preprint 411, Institute of Physics, Nicolaus Copernicus University, Torun, 1979.
11. V. P. Belavkin. On the theory of controlling observable quantum systems. *Automat. Remote Control*, 44(2):178–188, 1983.
12. V. P. Belavkin. Quantum stochastic calculus and quantum nonlinear filtering. *J. Multivariate Anal.*, 42:171–201, 1992.

13. L. Bouten, R. Van Handel, and M. R. James. An introduction to quantum filtering. *SIAM J. Control Optim.*, 46(6):2199–2241, 2007.
14. L. Bouten, R. Van Handel, and M. R. James. A discrete invitation to quantum filtering and feedback control. *SIAM Rev.*, 51(2):239–316, 2009.
15. M. P. A Branderhorst, P. Londero, P. Wasylczyk, C. Brif, R. L. Kosut, H. Rabitz, and I. A. Walmsley. Coherent control of decoherence. *Science*, 320:638–643, 2008.
16. D. D'Alessandro. *Introduction to Quantum Control and Dynamics*. Chapman & Hall/CRC, 2008.
17. A. C. Doherty and K. Jacobs. Feedback-control of quantum systems using continuous state-estimation. *Phys. Rev. A*, 60:2700, 1999.
18. J. Gough and M. R. James. Quantum feedback networks: Hamiltonian formulation. *Commun. Math. Phys.*, 287(DOI: 10.1007/s00220-008-0698-8):1109–1132, 2009.
19. J. Gough and M. R. James. The series product and its application to quantum feedforward and feedback networks. *IEEE Trans. Automatic Control*, 54(11):2530–2544, 2009.
20. S. Z. Sayed Hassen, M. Heurs, E. H. Huntington, I. R. Petersen, and M. R. James. Frequency locking of an optical cavity using linear quadratic Gaussian integral control. *J. Phys. B: At. Mol. Opt. Phys.*, 42:175501, 2009.
21. T.-S. Ho and H. Rabitz. Why do effective quantum controls appear easy to find? *J. Photochem. Photobiol. A: Chem.*, 180:226–240, 2006.
22. G. M. Huang, T. J. Tarn, and J. W. Clark. On the controllability of quantum-mechanical systems. *J. Math. Phys.*, 24(11):2608–2618, 1983.
23. M. R. James. A quantum Langevin formulation of risk-sensitive optimal control. *J. Opt. B: Semiclassical Quantum*, Special Issue on Quantum Control, 7(10):S198–S207, 2005.
24. M. R. James and J. Gough. Quantum dissipative systems and feedback control design by interconnection. *IEEE Trans Auto. Control*, to appear, arXiv.org: 0707.1074[quant-ph], 2010.
25. M. R. James, H. Nurdin, and I. R. Petersen. H^∞ control of linear quantum systems. *IEEE Trans Auto. Control*, 53(8):1787–1803.
26. N. Khaneja, R. Brockett, and S. J. Glaser. Time optimal control in spin systems. *Phys Rev A*, 63:032308, 2001.
27. N. Khaneja, S. J. Glaser, and R. Brockett. Sub-Riemannian geometry and time optimal control of three spin systems: Quantum gates and coherence transfer. *Phys Rev A*, 65:032301, 2002.
28. N. Khaneja, B. Luy, and S. Glaser. Boundary of quantum evolution under decoherence. *Proc. Nat. Acad. Sci., USA*, 100(23):13162–13166, 2003.
29. H. Mabuchi. Coherent-feedback quantum control with a dynamic compensator. *Phys. Rev. A*, 78(3):032323, 2008.
30. M. Mirrahimi and R. van Handel. Stabilizing feedback controls for quantum systems. *SIAM J. Control. Optim.*, 46:445–467, 2007.
31. H. Nurdin, M. R. James, and A. C. Doherty. Network synthesis of linear dynamical quantum stochastic systems. *SIAM J. Control Optim.*, 48(4):2686–2718, 2009.
32. H. Nurdin, M. R. James, and I. R. Petersen. Coherent quantum LQG control. *Automatica*, 45:1837–1846, 2009.
33. H. A. Rabitz, M. M. Hsieh, and C. M. Rosenthal. Quantum optically controlled transition landscapes. *Science*, 303:1998–2001, 2004.
34. L. K. Thomsen and H. M. Wiseman. Atom-laser coherence and its control via feedback. *Phys. Rev. A*, 65:063607, 2002.
35. I. Walmsley and H. Rabitz. Quantum physics under control. *Phys. Today*, 50:43–49, August 2003.
36. S. D. Wilson, A. R. R. de Carvalho, J. J. Hope, and M. R. James. Effects of measurement back-action in the stabilisation of a Bose–Einstein condensate through feedback. *Phys. Rev. A*, 76:013610, 2007.
37. H. Wiseman. Adaptive phase measurements of optical modes: Going beyond the marginal q distribution. *Phys. Rev. Lett.*, 75:4587–4590, 1995.
38. H. Wiseman and G. J. Milburn. Quantum theory of optical feedback via homodyne detection. *Phys. Rev. Lett.*, 70(5):548–551, 1993.
39. N. Yamamoto, H. Nurdin, M. R. James, and I. R. Petersen. Avoiding entanglement sudden-death via feedback control in a quantum network. *Phys. Rev. A*, 78:042339, 2008.
40. M. Yanagisawa and M. R. James. Atom-laser coherence via multiloop feedback control. *Phys Rev A*, 79:023620, 2009.
41. M. Yanagisawa and H. Kimura. Transfer function approach to quantum control-part I: Dynamics of quantum feedback systems. *IEEE Trans. Automatic Control*, 48:2107–2120, 2003.
42. V. P. Belavkin. Quantum continual measurements and *a posteriori* collapse on CCR. *Commun. Math. Phys.*, 146:611–635, 1992.

43. H. J. Carmichael. Quantum trajectory theory for cascaded open systems. *Phys Rev Lett.*, 70(15):2273–2276, 1993.
44. E. Merzbacher. *Quantum Mechanics*, 3rd ed. Wiley, New York, 1998.
45. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.
46. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.
47. C. W. Gardiner and P. Zoller. *Quantum Noise*. Springer, Berlin, 2000.
48. G. M. D'Ariano and P. Lo Presti. Quantum tomography for measuring experimentally the matrix elements of an arbitrary quantum operation. *Phys. Rev. Lett.*, 86:4195, 2001.
49. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
50. M. Mohseni, A. T. Rezakhani, and D. A. Lidar. Quantum process tomography: Resource analysis of different strategies. *Phys. Rev. A*, 77:032322, 2008.
51. J. Emerson, M. Silva, O. Moussa, C. Ryan, M. Laforest, J. Baugh, D. G. Cory, and R. Laflamme. Symmetrised characterisation of noisy quantum processes. *Science*, 317:1893, 2007.
52. M. P. A. Branderhorst, I. A. Walmsley, and R. L. Kosut. Quantum process tomography of decoherence in diatomic molecules. In *European Conference on Lasers and Electro-Optics*, Munich, June 2007.
53. E. J. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.*, 59(8):1207–1223, August 2006.
54. E. J. Candes, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 14:877–905, October 2008.
55. D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4), April 2006.
56. R. L. Kosut. Quantum process tomography via ℓ_1-norm minimization. arXiv.org: 0812.4323v1[quant-ph], 2008.
57. A. Shabani, R. L. Kosut, and H. Rabitz. Compressed quantum process tomography. arXiv.org: 0910.5498[quant-ph], 2009.
58. S. Boixo, S. T. Flammia, C. M. Caves, and J. M. Geremia. Generalized limits for single-parameter quantum estimation. *Phys Rev Lett.*, 98:090401, 2007.
59. S. Braunstein and C. Caves. Statistical distance and the geometry of quantum states. *Phys Rev Lett*, 72:3439, 1994.
60. V. Giovannetti, S. Lloyd, and L. Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96:010401, January 2006.
61. B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde. Entanglement-free Heisenberg-limited phase estimation. *Nature*, 450:393–396, 2007.
62. A. S. Holevo. *Probabilistic and Statistical Aspects of Quantum Theory*. North-Holland, Amsterdam, 1982.
63. M. Sarovar and G. J. Milburn. Optimal estimation of one parameter quantum channels. *J. Phys. A: Math. Gen.*, 39:8487–8505, 2006.
64. C. Brif and A. Mann. Nonclassical interferometry with intelligent light. *Phys. Rev. A*, 54:4505, 1996.
65. H. Rabitz. Making molecules dance: Optimal control of molecular motion. In A. D. Bandrauk, ed., *Atomic and Molecular Processes with Short Intense Pulses*. Plenum Publishing Corporation, New York, 1988.
66. A. Shabani, M. Mohseni, S. Lloyd, R. L. Kosut, and H. Rabitz. Efficient estimation of many-body quantum hamiltonians via random measurements. arXiv.org: 1002.1330[quant-ph], 2010.
67. M. D. Grace, J. Dominy, R. L. Kosut, C. Brif, and H. Rabitz. Environment-invariant measure of distance between evolutions of an open quantum system. *New J. Phys.*, 12:015001, 2010.
68. G. S. Uhrig. Keeping a quantum bit alive by optimized π-pulse sequences. *Phys. Rev. Lett.*, 98:100504, 2007.
69. L. Viola, E. Knill, and S. Lloyd. Dynamical decoupling of open quantum systems. *Phys. Rev. Lett.*, 82:2417, 1999.
70. H. Carmichael. *An Open Systems Approach to Quantum Optics*. Springer, Berlin, 1993.
71. H. Wiseman. Quantum theory of continuous feedback. *Phys Rev A*, 49(3):2133–2150, 1994.
72. H. M. Wiseman and G. J. Milburn. All-optical versus electro-optical quantum-limited feedback. *Phys Rev A*, 49(5):4110–4125, 1994.
73. W. M. Wiseman and G. J. Milburn. *Quantum Measurement and Control*. Cambridge University Press, Cambridge, UK, 2010.
74. K. R. Parthasarathy. *An Introduction to Quantum Stochastic Calculus*. Birkhauser, Berlin, 1992.

75. S. C. Edwards and V. P. Belavkin. Optimal quantum feedback control via quantum dynamic programming. arXiv.org:0506018[quant-ph], University of Nottingham, 2005.
76. M. R. James. Risk-sensitive optimal control of quantum systems. *Phys. Rev. A*, 69:032108, 2004.
77. C. D'Helon, A. C. Doherty, M. R. James, and S. D. Wilson. Quantum risk-sensitive control. In *Proc. 45th IEEE Conference on Decision and Control*, IEEE Publication, San Diego, pp. 3132–3137, December 2006.
78. H. Rabitz, T. S. Ho, M. Hsieh, R. Kosut, and M. Demiralp. Topology of optimally controlled quantum mechanical transition probability landscapes. *Phys. Rev. A*, 74:012721, 2006.
79. H. Rabitz, M. Hsieh, and C. Rosenthal. Quantum optimally controlled transition landscapes. *Science*, 303, 2004.
80. M. P. A. Branderhorst, I. A. Walmsley, R. L. Kosut, and H. Rabitz. Optimal experiment design for quantum state tomography of a molecular vibrational mode. *J. Phys. B: At. Mol. Opt. Phys.*, 41, 2008.
81. M. G. A. Paris, G. M. D'Ariano, and M. F. Sacchi. Maximum likelihood method in quantum estimation. arXiv.org: 0101071v1[quant-ph], 16 Jan 2001.
82. F. Verstraete, A. C. Doherty, and H. Mabuchi. Sensitivity optimization in quantum parameter estimation. *Phys. Rev. A*, 64:032111, 2001.
83. R. L. Kosut, I. A. Walmsley, and H. Rabitz. Optimal experiment design for quantum state and process tomography and Hamiltonian parameter estimation. arXiv.org: 0411093[quant-ph], Nov 2004.
84. H. Cramér. *Mathematical Methods of Statistics*. Princeton Press, Princeton, NJ, 1946.
85. M. Mohseni and A. T. Rezakhani. Dynamical evolution of superoperator for identification and control of quantum hamiltonian systems. arXiv.org: 0805.3188[quant-ph], 2008.
86. J. L. O'Brien, G. J. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White. Quantum process tomography of a controlled-not gate. *Phys. Rev. Lett.*, 93:080502, 2004.
87. K. C. Young, M. Sarovar, R. L. Kosut, and K. B. Whaley. Optimal quantum multi-parameter estimation as applied to dipole- and exchange-coupled qubits. *Phys. Rev. A*, 79:062301, 2009.
88. R. Vrijen, E. Yablonovitch, K. Wang, H. W. Jiang, A. Balandin, V. Roychowdhury, T. Mor, and D. DiVincenzo. Electron-spin-resonance transistors for quantum computing in silicon-germanium heterostructures. *Phys. Rev. A*, 62:1050–2947, 2000.
89. E. Yablonovitch, H. W. Jiang, H. Kosaka, H. D. Robinson, D. S. Rao, and T. Szkopek. Optoelectronic quantum telecommunications based on spins in semiconductors. *Proc. IEEE*, 91(5), May 2003.
90. R. P. Feynman, R. B. Leighton, and M. Sands. *The Feynman Lectures on Physics*. Addison-Wesley, Reading, MA, 1963–1965.
91. H. Hjalmarsson, M. Gevers, and F. De Bruyne. For model-based control design, closed loop identification gives better performance. *Automatica*, 32(12):1659–1673, 1996.
92. B. D. O. Anderson and J. B. Moore. *Linear Optimal Control*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
93. P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice-Hall, Englewood Cliffs, NJ, 1986.
94. D. H. Jacobson. Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games. *IEEE Trans. Automatic Control*, 18(2):124–131, 1973.
95. A. Bensoussan and J. H. van Schuppen. Optimal control of partially observable stochastic systems with an exponential-of-integral performance index. *SIAM J. Control Optimi.*, 23:599–613, 1985.
96. M. R. James, J. S. Baras, and R. J. Elliott. Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems. *IEEE Trans. Automatic Control*, 39:780–792, 1994.
97. P. Whittle. Risk-sensitive linear/ quadratic/ Gaussian control. *Adv. App. Probab.*, 13:764–777, 1981.
98. M. Green and D. J. N. Limebeer. *Linear Robust Control*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

32

Motion Control of Marine Craft

Tristan Perez
The University of Newcastle and Norwegian University of Science and Technology

Thor I. Fossen
Norwegian University of Science and Technology

Marine craft (surface vessels, underwater vehicles, and offshore rigs) perform operations that require tight motion control. During the past three decades, there has been an increasing demand for higher accuracy and reliability of marinecraft motion control systems. Today, these control systems are an enabling factor for single and multicraft marine operations. This chapter provides an overview of the main characteristics and design aspects of motion control systems for marine craft. In particular, we discuss the architecture of the control system, the functionality of its main components, the characteristics of environmental disturbances, control objectives, and essential aspects of modeling and motion control design.

32.1 System Architecture and Control Objectives

The purpose of a *marine craft motion control system* is to act on the craft using force actuators such that the craft follows a desired motion pattern despite environmental forces. The essential components of such

a control system are depicted in Figure 32.1. The main element is the *marine craft*, which incorporates sensors that provide information related to motion (position, velocity, acceleration), and actuators that produce forces to control the motion. The other system components can be grouped into three main subsystems, namely, guidance, navigation, and control:

Guidance system: The guidance system provides information on where the craft should go and how it should get there. The guidance system generates feasible desired reference trajectories described in terms of position, velocity, and acceleration. The trajectory may be generated by algorithms that use the craft's actual and desired position and, oftentimes, a mathematical model the craft dynamic response to control forces. Information about missions, operator decisions, weather, other local vessels, and fleet operations have a bearing on guidance.

Navigation system: The navigation system provides information on where the craft is and how it is sailing (speed and heading). The navigation system collects motion information from the various onboard sensors, such as global navigation satellite systems (GPS, Galileo, GLONASS), speed log, compass, RADAR, and accelerometers; it performs signal quality checking; and it transforms the measurements to a common reference frame used by the control and guidance systems.

Control system: The control system processes information from the navigation and guidance systems and provides actuator commands to control the motion. The control system uses motion-related signals to infer the state of the craft and disturbance forces. This processing often involves an *observer*–model-based filtering. The *motion controller* generates appropriate commands for the actuators so as to reduce the difference between the actual and desired craft trajectories. For some craft, the actuator configuration is such that the same desired control action can be delivered using different combinations of actuator commands. This provides increased reliability to actuator faults. In this case, it is common to incorporate a control allocation function, and let the motion

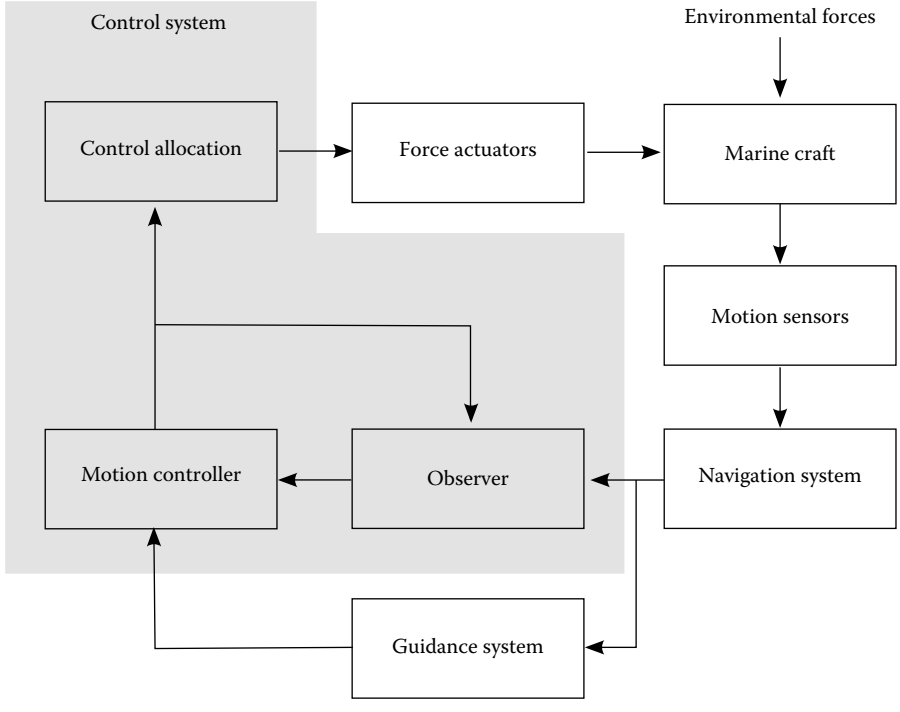

FIGURE 32.1 Marine craft motion control system.

controller be designed to generate desired force commands for the craft's degrees of freedom of interest instead of direct actuator commands. The *control allocation* function then maps the desired control forces into actuator commands. Should an actuator fail, the control allocation reconfigures the remaining healthy actuators; and thus, the effect of the fault may not require changes in the motion controller. This provides a first level of actuator-fault tolerance.

For marine craft, the environment consists of waves, wind, and ocean currents. The environment induces forces on the craft that are considered disturbances to the motion control system. These forces are conceptually separated into three components:

- Low frequency
- Wave frequency
- High frequency

where the descriptions *low* and *high* are relative to the wave frequency. Waves produce a pressure change on the craft's hull surface, which in turn induces forces. These pressure-induced forces have an oscillatory component that depends linearly on the wave elevation. Hence, these forces have the same frequency as that of the waves and are therefore referred to as *wave-frequency forces*. Wave forces also have a component that depends nonlinearly on the wave elevation. These nonlinear-force components have a mean component or wave drift force, a low-frequency oscillatory component, and a high-frequency oscillatory component. The low-frequency wave forces cause the vessel to drift and the oscillatory components can excite resonant modes in the horizontal motion of vessels that are at mooring or under positioning control. The high-frequency wave forces are normally too high to be considered in motion control of marine craft, but these forces may contribute to structural vibration of the hull. For further details about wave loads and their effects on craft motion, see Faltinsen (1990).

Wind and ocean currents also induce forces due to pressure variation on the craft structure. Wind forces have a mean component and an oscillatory component due to gusts. In marine craft motion control, only the mean wind forces are compensated since the frequency of gusts is often outside the bandwidth of the vessel response—this, however, depends on the size of the craft. Current-induced forces affect marine craft requiring positioning control and at mooring. These forces have a mean and an oscillatory low-frequency component due to hull vortex shedding. Low-frequency forces on marine craft, therefore, include the effect of waves, wind and current.

Due to the characteristics of the environmental forces mentioned above, the following problems are considered:

- Control only the low-frequency motion.
- Control only the wave-frequency motion.
- Control both low- and wave-frequency motion.

In low-to-medium sea states, the frequency of oscillations of the linear wave forces do not normally affect the operational performance of the craft. Hence, controlling only low-frequency motion avoids correcting the motion for every single wave, which can result in unacceptable operational conditions for the propulsion system due to power consumption and potential wear of the actuators. Applications that require the control of only the low-frequency motion include dynamic positioning, heading autopilots, and thruster-assisted position mooring. Dynamic positioning refers to the use of the propulsion system to regulate the horizontal position and heading of the craft. In thruster-assisted position mooring, the propulsion system is used to reduce the mean loading on the mooring lines. Additional applications that require the control of only the low-frequency motion include slow maneuver of surface vessels that arise, for example, from following underwater remotely operated vehicles.

Operations that require the control of only the wave-frequency motions include heave compensation for deploying loads at the sea floor as well as ride control of passenger vessels, where reducing roll and pitch motion helps avoid motion sickness (Perez, 2005). As the sea state develops, the waves grow in size

and their frequency reduces; and thus, the wave-induced forces start to appear within the bandwidth of the motion control system. In this situation, the control system objective may be to control both low- and wave-frequency motion. This is particularly so for positioning of offshore vessels and oil rigs, for which low-frequency motion is controlled in low sea states and both low- and wave-frequency motion is controlled in severe sea states (Fossen, 2002). Another example of a control problem that requires controlling both low- and wave-frequency motion is that of using the rudder for simultaneous course keeping and roll motion reduction in surface vessels—see Perez (2005) and references therein.

32.2 Marine Craft Rigid-Body Dynamics

32.2.1 Kinematics

To describe the motion of a marine craft, we consider two reference frames: *Earth-fixed* and *body-fixed*. Marine craft move at a relatively low speed, and hence considering the Earth to be an inertial frame is a good approximation. Associated with the Earth frame, we consider a local geographical coordinate system with origin o_n fixed to the mean water surface and positive directions along the North, East, and down. This system is abbreviated *NED* and denoted $\{n\}$. The craft is considered to be a rigid body, and thus a reference frame. Associated with the body frame there is a coordinate system with origin at a point o_b fixed to the craft and positive directions forward, starboard (right-hand side of the craft when looking toward the front), and down. Such a system is denoted by $\{b\}$. These frames are illustrated in Figure 32.2. The location of $\{b\}$ can vary for different control applications.

The position of the craft is given by the relative position of o_b with respect to o_n. The components of this vector are North, East, and Down positions and are denoted

$$\mathbf{p}_{b/n}^{n} \triangleq [N,\ E,\ D]^T.$$

The lower script b/n indicates that the position refers to that of o_b with respect to o_n. The upper script n indicates that components correspond to expressing the vector in $\{n\}$.

The orientation of the vessel is given by the Euler angles which correspond to three consecutive single rotations that take $\{n\}$ into the orientation of $\{b\}$. These rotations are $\{n\} \overset{\psi/z_n}{\rightarrow} \{n'\}$, $\{n'\} \overset{\theta/y'_n}{\rightarrow} \{n''\}$, and $\{n''\} \overset{\phi/x''_n}{\rightarrow} \{b\}$, where the upper script indicates the angle rotated about the axis of rotation, and the coordinate system on the right of the arrow is the result of the rotation. With the rotations thus defined,

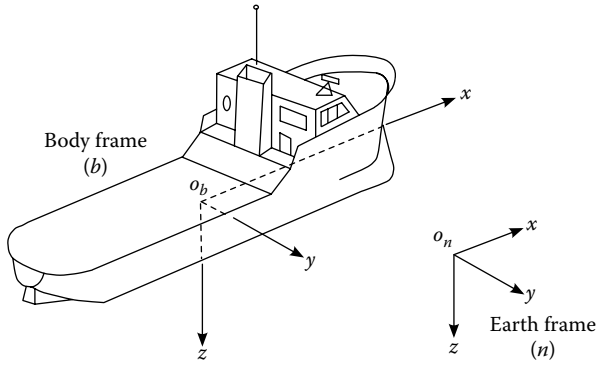

FIGURE 32.2 Reference frames and coordinate systems used for marine craft motion description.

the angles are called ψ–yaw, θ–pitch, and ϕ–roll. The *Euler angle vector* is given by

$$\boldsymbol{\Theta} \triangleq [\phi, \theta, \psi]^T.$$

The *generalized-position vector* (position-orientation) is defined by

$$\boldsymbol{\eta} \triangleq \begin{bmatrix} \mathbf{p}_{b/n}^{n} \\ \boldsymbol{\Theta} \end{bmatrix} = [N,\ E,\ D, \phi, \theta, \psi]^T. \tag{32.1}$$

The velocities are expressed in terms of body-fixed coordinates and are denoted by the *generalized velocity vector* (linear-angular),

$$\boldsymbol{\nu} \triangleq \begin{bmatrix} {}^{n}\dot{\mathbf{p}}_{b/n}^{b} \\ \boldsymbol{\omega}_{b/n}^{b} \end{bmatrix} = [u,\ v,\ w,\ p,\ q,\ r]^T. \tag{32.2}$$

The linear-velocity vector ${}^{n}\dot{\mathbf{p}}_{b/n}^{b} = [u,\ v,\ w]^T$ is the time derivative of the position vector as seen from the frame $\{n\}$, and the components correspond to expressing the vector in $\{b\}$. These components are the surge, sway, and heave velocities, respectively. The vector $\boldsymbol{\omega}_{b/n}^{b} = [p,\ q,\ r]^T$ is the angular velocity of the body with respect to the $\{n\}$ frame with components corresponding to expressing the vector in $\{b\}$. These components are the roll, pitch, and yaw rates respectively. Figure 32.3 shows the positive convention for the generalized velocities, and Table 32.1 summarizes the notation.

The trajectory of the craft is given by the time evolution of the generalized positions $\boldsymbol{\eta}$ defined in Equation 32.1. The time derivative of the positions is related to the body-fixed generalized velocities via a kinematic transformation,

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\, \boldsymbol{\nu}, \tag{32.3}$$

where

$$\mathbf{J}(\boldsymbol{\eta}) = \begin{bmatrix} \mathbf{R}_{n}^{b}(\boldsymbol{\Theta}) & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\Theta}) \end{bmatrix}.$$

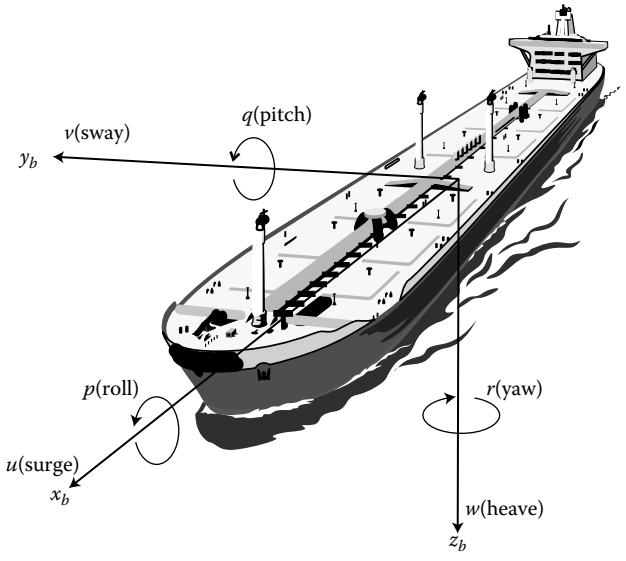

FIGURE 32.3 Positive convention of generalized velocities.

TABLE 32.1 Summary of Marine Craft Motion Variables

Variable	Name	Frame	Units
N	North position	Earth-fixed	m
E	East position	Earth-fixed	m
D	Down position	Earth-fixed	m
ϕ	Roll angle	–	rad
θ	Pitch angle	–	rad
ψ	Yaw angle	–	rad
u	Surge speed	Body-fixed	m/s
v	Sway speed	Body-fixed	m/s
w	Heave speed	Body-fixed	rad/s
p	Roll rate	Body-fixed	rad/s
q	Pitch rate	Body-fixed	rad/s
r	Yaw rate	Body-fixed	rad/s
$\mathbf{p}_{b/n}^{n} = [N,\ E,\ D]^T$	Position vector	Earth-fixed	
${}^{n}\dot{\mathbf{p}}_{b/n}^{b} = [u,\ v,\ w]^T$	Linear-velocity vector	Body-fixed	
$\mathbf{\Theta} = [\phi, \theta, \psi]^T$	Euler-angle vector	–	
$\boldsymbol{\omega}_{b/n}^{b} = [p, q, r]^T$	Angular-velocity vector	Body-fixed	
$\boldsymbol{\eta} = [(\mathbf{p}_{b/n}^{n})^T, \mathbf{\Theta}^T]^T$	Generalized position vector	–	
$\boldsymbol{\nu} = [({}^{n}\dot{\mathbf{p}}_{b/n}^{b})^T, (\boldsymbol{\omega}_{b/n}^{b})^T]^T$	Generalized velocity vector	Body-fixed	

The rotation matrix $\mathbf{R}_b^n(\mathbf{\Theta})$ is given by

$$\mathbf{R}_b^n(\mathbf{\Theta}) = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ s_\psi c_\theta & c_\psi c_\phi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + s_\psi c_\phi s_\theta \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix},$$

where $s_x \equiv \sin(x)$ and $c_x \equiv \cos(x)$. Note that rotation matrices are orthogonal, that is, $\mathbf{R}(\mathbf{\Theta})^{-1} = \mathbf{R}(\mathbf{\Theta})^T$ and also $\det \mathbf{R}(\mathbf{\Theta}) = 1$. The transformation from the angular velocity expressed in the body-fixed coordinates to the time derivatives of the Euler angles is given by

$$\mathbf{T}(\mathbf{\Theta}) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \dfrac{s_\phi}{c_\theta} & \dfrac{c_\phi}{c_\theta} \end{bmatrix}, \quad t_\theta \equiv \tan(\theta), \quad \cos(\theta) \neq 0.$$

Note that $\mathbf{T}(\mathbf{\Theta})$ is not orthogonal. Also, $\mathbf{T}(\mathbf{\Theta})$ and its inverse are singular for $\theta = \pm\pi/2$—known as the Euler-angle singularity. This is not usually a problem for marine surface vessels, but it may be an issue for underwater vehicles. In such cases, alternative representation of the kinematic transformations without singularities can be obtained in terms of quaternions. For further details about marine craft kinematics, see Fossen (2002).

Note that there is no physical vector whose time-derivative gives $\boldsymbol{\nu}$ (Goldstein, 1980). Note also that in the literature of analytical mechanics, the term generalized velocity usually refers to $\dot{\boldsymbol{\eta}}$; however, in the context of this chapter, we use the term in relation to the body-fixed velocity vector $\boldsymbol{\nu}$.

32.2.2 Equations of Motion

The equations of motion of an unconstrained rigid body can be derived using either vectorial or analytical mechanics. Here, we will outline the second approach, and in particular the work of Kirchhoff, in which a rigid body moving in a fluid and the fluid are teated as a single dynamic system (Lamb, 1932).

The kinetic energy of the craft due to its rotation and translation (without interacting with the fluid) can be expressed in terms of body-fixed generalized velocities (Egeland and Gravdahl, 2002; Fossen, 2002),

$$T = \frac{1}{2}\boldsymbol{\nu}^T \boldsymbol{M}_{RB}\,\boldsymbol{\nu}, \tag{32.4}$$

where the craft *rigid-body generalized mass matrix* is of the form

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbf{I}_{3\times 3} & -m\mathbf{S}(\mathbf{p}^b_{g/b}) \\ m\mathbf{S}(\mathbf{p}^b_{g/b}) & \mathbf{I}^b_b \end{bmatrix},$$

in which, m is the mass of the craft, $\mathbf{p}^b_{g/b}$ is the position of the craft's center of gravity (CG) relative to o_b in $\{b\}$, and $\mathbf{S}(\mathbf{a})$ is, by definition, the skew-symmetric matrix form of any vector $\mathbf{a} = [a_x, a_y, a_z]^T$, that is

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}.$$

The inertia matrix $\mathbf{I}^b_b$ about the point o_b can be expressed using the *parallel-axis theorem*, namely

$$\mathbf{I}^b_b = \mathbf{I}^b_g - m\mathbf{S}(\mathbf{p}^b_{g/b})\mathbf{S}(\mathbf{p}^b_{g/b}) = \begin{bmatrix} I^b_{xx} & -I^b_{xy} & -I^b_{xz} \\ -I^b_{yx} & I^b_{yy} & -I^b_{yz} \\ -I^b_{zy} & -I^b_{zy} & I^b_{zz} \end{bmatrix},$$

where $\mathbf{p}^b_{g/b}$ is position of the CG in $\{b\}$, and $\mathbf{I}^b_g$ is the inertia matrix about CG in $\{b\}$. The inertia matrix about CG can be computed using the following sum over the vessel mass particles m_i:

$$\mathbf{I}^b_g = \sum_i m_i\, \mathbf{S}^T(\mathbf{p}^b_{i/g})\, \mathbf{S}(\mathbf{p}^b_{i/g}),$$

where $\mathbf{p}^b_{i/g}$ represents the position of the mass particle i with respect to the CG.

Let the body-fixed *generalized forces* (forces and moments) be

$$\boldsymbol{\tau} = [X, Y, Z, K, M, N]^T,$$

where

- X, Y, Z are the surge, sway and heave forces respectively.
- K, M, N are the roll, pitch, and yaw moments respectively.

Then, we can use Kirchhoff's equations to derive a dynamic model that relates the forces to the velocities (Kirchhoff, 1869):

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \boldsymbol{\nu}_1}\right) + \mathbf{S}(\boldsymbol{\nu}_2)\frac{\partial T}{\partial \boldsymbol{\nu}_1} = \boldsymbol{\tau}_1, \tag{32.5}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \boldsymbol{\nu}_2}\right) + \mathbf{S}(\boldsymbol{\nu}_1)\frac{\partial T}{\partial \boldsymbol{\nu}_1} + \mathbf{S}(\boldsymbol{\nu}_2)\frac{\partial T}{\partial \boldsymbol{\nu}_2} = \boldsymbol{\tau}_2, \tag{32.6}$$

where,

$$\boldsymbol{\nu}_1 = [u,\, v,\, w]^T, \quad \boldsymbol{\nu}_2 = [p,\, q,\, r]^T,$$
$$\boldsymbol{\tau}_1 = [X,\, Y,\, Z]^T, \quad \boldsymbol{\tau}_2 = [K,\, M,\, N]^T.$$

Substituting Equation 32.4 into Equations 32.5 and 32.6 and manipulating the latter, a general model structure for craft's rigid-body dynamics can be expressed as

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}. \tag{32.7}$$

The first term on the left-hand side of Equation 32.7 is the product of the generalized mass matrix and the acceleration. The second term represents forces due to Coriolis and centripetal accelerations. These

accelerations are due to the rotation of the the body frame relative to the local geographical frame $\{n\}$. *Note these Coriolis forces are different from the Coriolis forces due to the rotation of the Earth.* The latter are ignored since marine craft move at low speeds; and therefore, we can assume that the Earth frame is inertial.

The Coriolis and centripetal acceleration matrix in Equation 32.7 can be expressed in different ways; one of such representations is

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} m\mathbf{S}(\boldsymbol{\nu}_2) & -m\mathbf{S}(\boldsymbol{\nu}_2)\mathbf{S}(\mathbf{p}^b_{g/b}) \\ m\mathbf{S}(\mathbf{p}^b_{g/b})\mathbf{S}(\boldsymbol{\nu}_2) & -\mathbf{S}(\mathbf{I}^b_b\boldsymbol{\nu}_2) \end{bmatrix}. \tag{32.8}$$

For alternative representations of Equation 32.8 see Fossen (2002). Note that

$$\dot{\mathbf{M}}_{RB} = \mathbf{0}, \qquad \mathbf{M}_{RB} = \mathbf{M}^T_{RB}, \qquad \mathbf{C}_{RB} = -\mathbf{C}^T_{RB}.$$

The kinematic transformation Equation 32.3 together with the kinetic model (Equation 32.7) provide a dynamic model for rigid-body motion of the craft without interaction with the fluid. To describe such interaction, we need to separate the generalized forces on the right-hand side of Equation 32.7 into

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{hyd} + \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{env}, \tag{32.9}$$

where $\boldsymbol{\tau}_{hyd}$ describes fluid pressure-induced forces due to the motion of the craft, $\boldsymbol{\tau}_{ctrl}$ the control forces due to actuators, and $\boldsymbol{\tau}_{env}$ the environmental forces due to waves, wind and current.

32.3 Maneuvering Hydrodynamics and Models

The study of marine craft dynamics has traditionally been covered by two main theories: *maneuvering* and *seakeeping*. Maneuvering refers to the study of craft motion in the absence of wave excitation (calm water). Seakeeping, on the other hand, refers to the study of motion when there is wave excitation and while the vessel keeps its course and speed constant (which includes the case of zero speed). Although both areas are concerned with the same issues: study of motion, stability and control, the separation allows making different assumptions that simplify the study hydrodynamic forces in Equation 32.9.

In maneuvering theory (in calm water), the hydrodynamic forces in Equation 32.9 can be expressed as

$$\boldsymbol{\tau}_{hyd} = -\mathbf{M}_A\dot{\boldsymbol{\nu}} - \mathbf{C}_A\left(\boldsymbol{\nu}\right)\boldsymbol{\nu} - \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{g}(\boldsymbol{\eta}). \tag{32.10}$$

The first two terms on the right-hand side of Equation 32.10 can be explained by considering the motion of the craft in an irrotational flow and for ideal fluid (no viscosity). As the craft moves, it changes the momentum of the fluid. The kinetic energy of the ideal fluid due to the motion of the craft can be expressed as

$$T_A = \frac{1}{2}\boldsymbol{\nu}^T\mathbf{M}_A\boldsymbol{\nu}, \tag{32.11}$$

where the constant matrix $\mathbf{M}_A$ is called the matrix of *added mass* coefficients,

$$\mathbf{M}_A = -\begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{p} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix}, \qquad \dot{\mathbf{M}}_A = \mathbf{0}.$$

The notation used for the coefficients is related to the forces. For example,

- The product $X_{\dot{u}}\,\dot{u}$ gives the force in surge due to surge acceleration,
- The product $Y_{\dot{r}}\,\dot{r}$ is the sway force due to the yaw angular acceleration.

Note that not all the coefficients have units of mass, and that they have signs. For example $X_{\dot{u}} < 0$, and the sign of $Y_{\dot{r}}$ depends on the extent of fore–aft symmetry of the submerged hull. Note also that depending on the symmetry of the hull, some of the added mass coefficients can be zero. For vessels maneuvering at low speeds the added mass matrix is positive definite and symmetric. For surface vessels in waves sailing at forward speed symmetry may be lost (Faltinsen, 1990).

Using the fluid kinetic energy (Equation 32.11) in Kirchhoff's equations 32.5 and 32.6, we can obtain the forces on the vessel due to the change in the energy of the ideal fluid (Lamb, 1932) (p. 168). With some elementary algebraic work, this gives the first two terms in Equation 32.10 (Fossen, 2002). The first term represents pressure-induced forces proportional to the accelerations of the craft. The second term corresponds to Coriolis and centripetal forces due to the added mass. The Coriolis-centripetal matrix can be expressed as

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3\times 3} & -\mathbf{S}(\mathbf{A}_{11}\boldsymbol{\nu}_1 + \mathbf{A}_{12}\boldsymbol{\nu}_2) \\ -\mathbf{S}(\mathbf{A}_{11}\boldsymbol{\nu}_1 + \mathbf{A}_{12}\boldsymbol{\nu}_\nu) & -\mathbf{S}(\mathbf{A}_{21}\boldsymbol{\nu}_1 + \mathbf{A}_{22}\boldsymbol{\nu}_2) \end{bmatrix},$$

where

$$\mathbf{A} = \frac{1}{2}(\mathbf{M}_A + \mathbf{M}_A^T).$$

The third term on the right-hand side of Equation 32.10 corresponds to damping forces, which have the following origins:

Potential damping: This damping force is the result of a body passing through a fluid and wake wave making. The word *potential* indicates that this damping force can be obtained from a study of irrotational flow in an ideal fluid (no viscosity). In such a fluid, one can define a potential function of the space, such that its gradient gives the vector field of flow velocity, and the pressure can be computed from the potential function (Lamb, 1932; Newman, 1977; Faltinsen, 1990). As the fluid is displaced, there is a buildup of pressure in front, and a decrease of pressure behind the body. This pressure difference induces a force that opposes the motion.

Skin friction: This viscous effect is caused by the passage of water over the wetted surface of the hull. At low speeds, it is dominated by a linear component, while at higher speeds, nonlinear terms dominate.

Vortex shedding: This damping force arises from the vortex generation due to flow separation (viscous effects), which occurs at the sharp edges commonly found at the bow and stern of a marine craft, or the control surfaces.

Lifting Forces: The hull of a craft can be modeled as a low aspect ratio wing (Blanke, 1981; Ross et al., 2007; Ross, 2008). The forces generated by the hull at some angle of attack due to a maneuver can be resolved into two components: lift and drag. The former acts perpendicular to the direction of motion of the vessel, while the latter acts in the opposite direction of the motion. This is depicted in Figure 32.4. Note that the lift force is not actually a damping force as it is oriented perpendicular to the motion of the vessel, it does not serve to dissipate energy. With an abuse of terminology, all lift forces are generically grouped into a damping term. The drag component that arises due to lift is known as *parasitic drag*, or *lift-induced drag*. This force acts directly in opposition to motion, and is properly referred to as a damping force.

For vessels maneuvering such that the athwart component of the flow velocity is larger than the forward component, viscous damping effects due to skin friction and vortex shedding are generally grouped into what is called *cross-flow drag* (Faltinsen, 1990). This is particular of low-speed maneuvering and positioning applications.

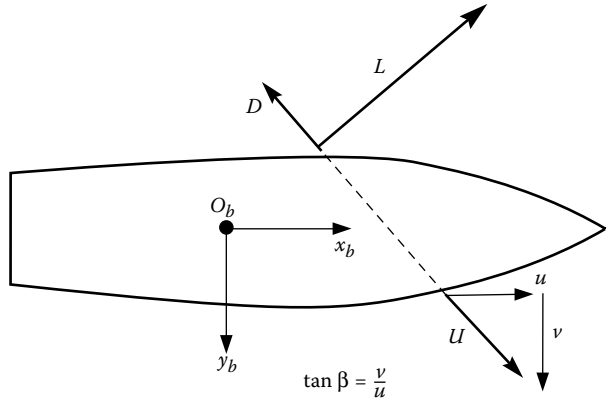

FIGURE 32.4 Lift and drag forces experienced during maneuvering.

Due to the complex interaction of different effects, the hydrodynamic damping forces of maneuvering vessels are generally modeled by a series expansion, and the coefficients of the model are obtained via regression analysis of data measured from scaled model tests or by system identification. Two types of mathematical models are commonly used. The first one consists of representing the complex interactions by a multivariable Taylor expansion with only odd terms generally up to third order. This model was proposed by Abkowitz (1964) taking into consideration port-starboard symmetry, and it has the mathematical elegance of the Taylor terms, but they have no inherent physical meaning. The second model commonly used is an expansion in terms of second-order modulus terms ($|x|x$). This approach was introduced by Fedyaevsky and Sobolev (1964) to capture cross-flow drag effects at large angles of incidence β, see Figure 32.4. It should be mentioned that the differences between these two type of model of are quite fundamental, and the different coefficients are irreconcilable. Ross et al. (2007), see also Ross (2008), used a first-principle approach and considered the different theories that explain the phenomena involved to a great extent and derived a comprehensive model.

The last term in Equation 32.10 represents forces due to gravity and buoyancy. These forces tend to restore the up-right equilibrium of the vessel; and therefore, are called *restoring forces*. These forces depend on the displacement volume of the vessel, its shape and heave, pitch, and roll angles:

$$\mathbf{g}(\boldsymbol{\eta}) = [0,\ 0, Z_g(\boldsymbol{\eta}),\ M_g(\boldsymbol{\eta}),\ K_g(\boldsymbol{\eta}),\ 0]^T.$$

Depending on the symmetry of the vessel, the forces may be coupled. For example there is usually a heave force due to pitch angle, which is a consequence of the fore–aft asymmetry of the hull.

Replacing Equation 32.10 into Equation 32.9 and combining the latter with Equations 32.7 and 32.3, a maneuvering model takes the following form:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{32.12}$$

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{env}. \tag{32.13}$$

If there is ocean current, the model must be modified according to

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{32.14}$$

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{C}_A(\boldsymbol{\nu}_{rc})\boldsymbol{\nu}_{rc} + \mathbf{D}(\boldsymbol{\nu}_{rc})\boldsymbol{\nu}_{rc} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{env}, \tag{32.15}$$

where $\boldsymbol{\nu}_{rc}$ is the craft velocity relative to the current (seen from the body-fixed frame):

$$\boldsymbol{\nu}_{rc} = \boldsymbol{\nu} - \boldsymbol{\nu}_c,$$

where $\boldsymbol{\nu}_c = [u_c, v_c, w_c, 0, 0, 0]^T$. Current forces can be separated into a potential component due to irrotational flow in an ideal fluid and a viscous component (nonideal fluid). The potential part in the model

Equation 32.15 is represented by the Coriolis and centripetal term $\mathbf{C}_A(\boldsymbol{\nu}_{rc})\boldsymbol{\nu}_{rc}$ due to added mass, whereas the viscous part is incorporated in the damping term $\mathbf{D}(\boldsymbol{\nu}_{rc})\boldsymbol{\nu}_{rc}$. Then, $\boldsymbol{\tau}_{env}$ in Equation 32.15 accounts for environmental forces other than ocean currents—for example, wind in the case of surface vessels. We next present an example of a maneuvering model for a surface vessel.

32.3.1 Example: Maneuvering Model of a High-Speed Vehicle–Passenger Trimaran

In this section, we consider an example of a 4-degree-of-freedom maneuvering model of high-speed trimaran adapted from Perez et al. (2007). Figure 32.5 shows a picture of the vessel.

The model considered is given by Equations 32.12 and 32.13 for the degrees of freedom of surge, sway, roll, and yaw, that is,

$$\begin{aligned}\boldsymbol{\eta} &= [x, y, \phi, \psi]^T \\ \boldsymbol{\nu} &= [u, v, p, r]^T \\ \boldsymbol{\tau} &= [X, Y, K, N]^T.\end{aligned}$$

The rigid-body mass and Coriolis-centripetal matrices are given by

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 & -my_g \\ 0 & m & -mz_g & mx_g \\ 0 & -mz_g & I^b_{xx} & -I^b_{xz} \\ -my_g & mx_g & -I^b_{zx} & I^b_{zz} \end{bmatrix},$$

and

$$\mathbf{C}_{RB}(\mathbf{n}) = \begin{bmatrix} 0 & 0 & mz_g r & -m(x_g r + v) \\ 0 & 0 & -my_g p & -m(y_g r - u) \\ -mz_g r & my_g p & 0 & I^b_{yz} r + I^b_{xy} p \\ m(x_g r + v) & m(y_g r - u) & -I^b_{yz} r - I^b_{xy} p & 0 \end{bmatrix},$$

FIGURE 32.5 Austal's hull H260 "Benchijigua Express." (Courtesy of Austal Ships, http://austal.com.)

where m is the mass of the vessel, $\mathbf{p}^b_{g/b} = [x_g, y_g, z_g]^T$ gives position the CG relative to o_b, and I^b_{ik} are the moments and products of inertia about o_b.

The kinematic transformation (Equation 32.12) reduces to

$$\mathbf{J}(\boldsymbol{\eta}) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & \cos(\psi)\cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cos(\phi) \end{bmatrix}.$$

The added mass matrix and the Coriolis-centripetal matrix due to added mass are given by

$$\mathbf{M}_A = \mathbf{M}_A^T = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{p}} & Y_{\dot{r}} \\ 0 & K_{\dot{v}} & K_{\dot{p}} & K_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{p}} & N_{\dot{r}} \end{bmatrix},$$

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & Y_{\dot{v}}v + Y_{\dot{p}}p + Y_{\dot{r}}r \\ 0 & 0 & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & 0 \\ -Y_{\dot{v}}v - Y_p p - Y_{\dot{r}}r & X_{\dot{u}}u & 0 & 0 \end{bmatrix}.$$

The adopted damping terms take into account the lift, drag, and viscous effects.

$$\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D}_{LD}(\boldsymbol{\nu}) + \mathbf{D}_{VIS}(\boldsymbol{\nu}),$$

where

$$\mathbf{D}_{LD}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & X_{rv}\,v \\ 0 & Y_{uv}\,u & 0 & Y_{ur}\,u \\ 0 & K_{uv}\,u & 0 & K_{ur}\,u \\ 0 & N_{uv}\,u & 0 & N_{ur}\,u \end{bmatrix}. \tag{32.16}$$

$$\mathbf{D}_{VIS}(\boldsymbol{\nu}) = \begin{bmatrix} X_{u|u|} & 0 & 0 & 0 \\ 0 & Y_{|v|v}\,|v| + Y_{|r|v}\,|v| & 0 & Y_{|v|r}\,|v| + Y_{|r|r}\,|r| \\ 0 & 0 & K_{p|p|} + Y_p & 0 \\ 0 & N_{|v|v}\,|v| + N_{|r|v}\,|v| & 0 & N_{|v|r}\,|v| + N_{|r|r}\,|r| \end{bmatrix}. \tag{32.17}$$

The lift-drag representation in Equation 32.16 is consistent with taking only the first order terms derived in Ross et al. (2007)—see also Ross (2008), whereas the viscous damping representation in Equation 32.17 follows from Blanke (1981). Finally, the restoring term reduces to

$$\mathbf{g}(\boldsymbol{\eta}) = [0,\ 0,\ M_g(\boldsymbol{\eta}),\ 0]^T.$$

Figure 32.6 shows model validation data for the velocities based on a full scale zig-zag sea trial. The hydrodynamic parameters of the model were partially obtained from computational fluid dynamic software, and partially by optimizing model prediction errors using full-scale trial data. For further details see Perez et al. (2007).

32.4 Seakeeping Hydrodynamics and Models

Seakeeping theory of marine craft motion refers to the study of motion in waves while the vessel keeps a constant course and speed. The equations of motion are described within the linear framework, which allows considering the problem in the frequency domain. The latter, in turn, enables one to compute

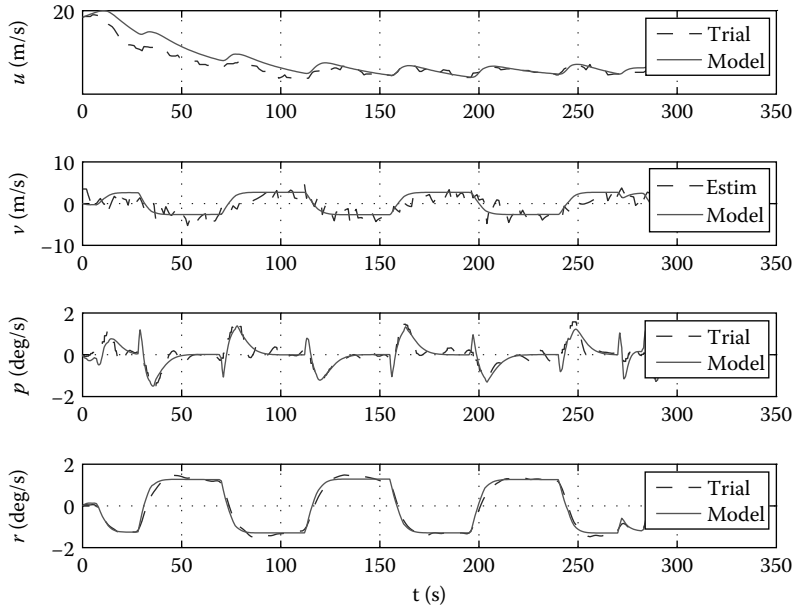

FIGURE 32.6 Model validation against a zig-zag sea trial data for a high-speed trimaran ferry.

wave-to-force and force-to-motion frequency response functions. The waves are described as realizations of stationary stochastic processes. When the vessel frequency responses are combined with the wave elevation power spectral density (PSD), one can compute the response, from which several statistics of motion are derived. The approach just described is commonly used by naval architects to compare different hull forms at a craft-design stage.

The frequency-domain models used in seakeeping theory provide valuable information for marine craft motion control design. In this section, we discuss models for wave elevation, vessel frequency response with particular parameterizations used in naval architecture, and time-domain simulations. Finally, we discuss how the seakeeping models can be combined with maneuvering models for control system design.

32.4.1 Wave Environment

Ocean waves are random in both time and space. These characteristics are often summarized by the term *irregular* in the marine literature. The stochastic description is, therefore, the most appropriate approach to characterize them. The following simplifying assumption regarding the underlying stochastic model are usually made: *The observed sea surface elevation* $\zeta(t)$*, at a certain location and for short periods of time, is considered a realization of a stationary and homogeneous zero mean Gaussian stochastic process.* The period for which ocean waves can be considered stationary can vary between 20 min and 3 h. For deep water, wave elevation tends to present a Gaussian distribution, as the water becomes shallow nonlinear effects dominate, and the waves become non-Gaussian (Ochi, 1998).

Under the stationary and Gaussian assumptions, the sea surface elevation is completely characterized by its PSD $S_{\zeta\zeta}(\omega)$, commonly referred to as the *wave spectrum*,

$$E[\zeta(t)^2] = \int_0^\infty S_{\zeta\zeta}(\omega)\, d\omega.$$

The wave spectrum can be estimated from data records. However, to study the response of marine structures, a family of idealized spectra is commonly used. One commonly used family is the modified

Pierson-Moskowitz or International Towing Tank Conference (ITTC) spectrum—which was recommended by the ITTC in 1978:

$$S_{\zeta\zeta}(\omega) = \frac{A}{\omega^5} \exp\left(\frac{-B}{\omega^4}\right) \quad (\mathrm{m^2 s}). \tag{32.18}$$

The parameters A and B are given by

$$A = \frac{173 H_{1/3}^2}{T_1^4}, \quad B = \frac{691}{T_1^4},$$

where $H_{1/3}$ is the significant wave height (average of the highest one third of the waves) and T_1 is the average wave period. These two parameters are referred to as long-term statistics. There are wave atlases with scatter diagrams of $H_{1/3}$ and T_1 for specific locations on the globe and time of the year. Figure 32.7 shows a plot of the ITTC wave spectrum (Equation 32.18) for 4 m significant wave height and three average wave periods. This type of spectrum is used to describe fully developed seas in deep water. For a thorough introduction to the modeling of ocean waves see Ochi (1998), and for the models used in marine craft motion control see Perez (2005).

When a marine craft is at rest, the frequency at which the waves excite the craft coincides with the wave frequency; and thus, the previous description is valid. However, when the craft moves with a constant forward speed U, the frequency observed from the craft differs from the wave frequency. The frequency experienced by the craft is called the *encounter frequency*. The encounter frequency depends not only on the speed of the craft, but also on the angle the waves approach:

$$\omega_e = \omega - \frac{\omega^2 U}{g} \cos(\chi). \tag{32.19}$$

where, the encounter angle χ defines the sailing condition, namely,

- Following seas ($\chi = 0$ or 360 deg)
- Quartering seas ($0 < \chi < 90$ deg or $270 < \chi < 360$ deg)
- Beam seas ($\chi = 90$ deg—port or 270 deg—starboard)
- Bow seas ($90 < \chi < 180$ deg or $180 < \chi < 270$ deg)
- Head seas ($\chi = 180$ deg)

The encounter frequency captures a Doppler effect. Figure 32.8 shows a schematic representation of the transformation Equation 32.19. From this figure, we can see that when the vessel is sailing in bow or head seas, the wave frequencies are mapped into higher frequencies. In beam seas, however, there is no change and both ω and ω_e are the same. In following and quartering seas, the situation becomes more involved as different wave frequencies can be mapped into the same encounter frequency. In deep water,

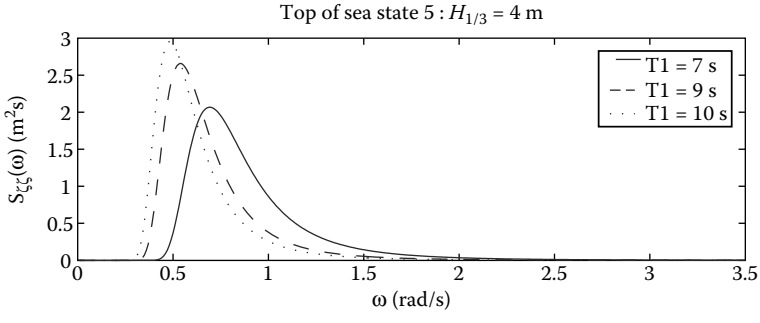

FIGURE 32.7 Example of ITTC PSD for the wave elevation and wave slope for $H_{1/3} = 4$ m and $T_1 = 7$, 9 and 10 s.

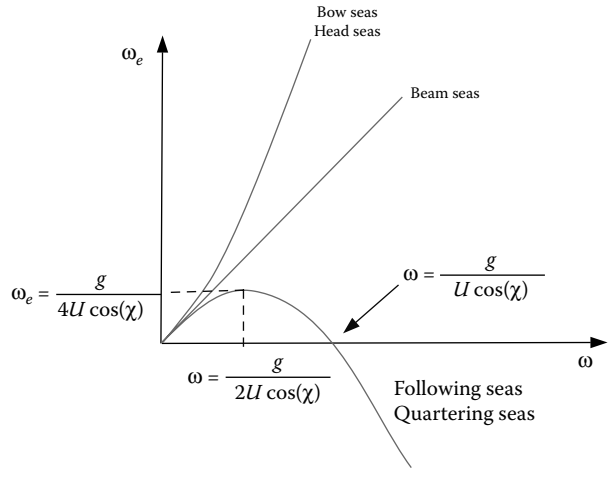

FIGURE 32.8 Transformation from wave to encounter frequency under different sailing conditions for U fixed.

long waves travel faster than short waves. Hence, in following and quartering seas, long waves overtake the vessel whereas short waves are overtaken by the vessel. Indeed, for $0 < \omega < \frac{g}{U\cos(\chi)}$ the waves overtake the vessel. The wave frequency $\omega = \frac{g}{U\cos(\chi)}$, at which $\omega_e = 0$, corresponds to the situation in which the component of the craft velocity in the direction of wave propagation is the same as the wave celerity. In this case, the wave pattern observed from the craft remains stationary and travels along with the craft. Finally, for high-frequency waves, the encounter frequency is negative, meaning that the craft overtakes the waves.

Since the power of any magnitude is invariant with respect to the reference frame from which it is observed, for any PSD the following holds:

$$S(\omega_e)\, d\omega_e = S(\omega)\, d\omega.$$

From this, it follows that

$$S(\omega_e) = \frac{S(\omega)}{\left|\frac{d\omega}{d\omega_e}\right|} = \frac{S(\omega)}{\left|1 - \frac{2\omega U}{g}\cos(\chi)\right|}.$$

For beam seas, the transformation is trivial, that is since $\cos(\pi/2) = 0$, then $S(\omega_e) = S(\omega)$. In bow seas, the encounter spectrum is a spread version of the wave spectrum shifted toward higher frequencies. For quartering and following seas the situation becomes complex since expression (Section 32.4.1) is singular at $\bar{\omega}_w = g/(2U\cos\chi)$ where the denominator vanishes. This is an integrable singularity, and the variance of the process remains the same in both wave- and encounter-frequency domains (Price and Bishop, 1974).

The motion of a marine craft in waves is the result of the wave excitation due to the varying distribution of pressure on the hull. Therefore, the wave excitation, as well as the vessel response, will depend not only on the characteristics of the waves—amplitude and frequency—but also on the *sailing conditions*: encounter angle and speed. The wave spectrum, and thus the wave-induced forces can change significantly with the sailing conditions for a given sea state. These changes have a significant bearing on control system design.

32.4.2 Time-Domain Seakeeping Models

For simplicity, in this and the next section, we will only consider the case of zero forward speed and then comment on the extensions to the forward speed case in Section 32.4.4.

For the zero-speed case, the models (32.3) and (32.7) can be linearized about the equilibrium point ($\bar{\boldsymbol{\eta}} = \mathbf{0}, \bar{\boldsymbol{\nu}} = \mathbf{0}$) and expressed as

$$\delta\dot{\boldsymbol{\eta}} = \delta\boldsymbol{\nu},$$
$$\mathbf{M}_{RB}\delta\dot{\boldsymbol{\nu}} = \delta\boldsymbol{\tau},$$

where $\delta\boldsymbol{\eta}$ represents the generalized perturbation position-orientation vector and $\delta\boldsymbol{\nu}$ the generalized perturbation body-fixed velocity vector.

For a vessel in waves, the generalized pressure force $\delta\boldsymbol{\tau}$ vector can be separated into two components:

$$\delta\boldsymbol{\tau} = \boldsymbol{\tau}_{rad} + \boldsymbol{\tau}_{exc}.$$

The first component corresponds to the radiation forces arising from the change in momentum of the fluid due to the oscillatory motion of the craft at the wave frequency. The radiation forces are the result of the waves being radiated from the hull as a consequence of its motion. The excitation represents the pressure forces due to the incoming waves. Both radiation and excitation forces in sinusoidal waves can be studied using potential theory, that is, irrotational flow of an ideal fluid (Newman, 1977; Faltinsen, 1990).

In seakeeping theory, the forces are computed in a reference frame fixed to the equilibrium position of the vessel—called *equilibrium or seakeeping frame*, to which we associate a coordinate system $\{s\}$. Hence, the forces require a kinematic transformation. For the case where the seakeeping frame is stationary relative to the local Earth frame $\{n\}$ (zero speed case), these kinematic transformations can be neglected under the small angle assumption. Then, for zero speed,

$$\delta\boldsymbol{\eta} \approx \boldsymbol{\eta},$$
$$\delta\boldsymbol{\nu} \approx \boldsymbol{\nu},$$
$$\delta\boldsymbol{\tau} \approx \boldsymbol{\tau}.$$

Cummins (1962) studied the radiation hydrodynamic problem in an ideal fluid and found the following representation for linear hydrodynamic forces:

$$\boldsymbol{\tau}_{rad} = -\bar{\mathbf{A}}\,\dot{\boldsymbol{\nu}} - \int_0^t \mathbf{K}(t-t')\boldsymbol{\nu}(t')\,dt'. \tag{32.20}$$

The first term in Equation 32.20 represents forces due the accelerations of the structure, and $\bar{\mathbf{A}}$ is the constant positive-definite added mass matrix*. The second term represents fluid-memory effects that incorporate the energy dissipation due the radiated waves as a consequence of the motion of the vessel. The kernel of the convolution term, $\mathbf{K}(t)$, is the matrix of *retardation* or *memory* functions (impulse responses).

By renaming the variables, combining terms, and adding the hydrostatic restoring forces due to gravity and buoyancy ($\boldsymbol{\tau}_{hs} = -\mathbf{G}\boldsymbol{\eta}$), we obtain the *Cummins Equation* for zero forward speed:

$$(\mathbf{M} + \bar{\mathbf{A}})\dot{\boldsymbol{\nu}} + \int_0^t \mathbf{K}(t-t')\boldsymbol{\nu}(t')\,dt' + \mathbf{G}\boldsymbol{\eta} = \boldsymbol{\tau}_{exc}. \tag{32.21}$$

Equation 32.21 describes the motion of a vessel at zero speed for any wave excitation $\boldsymbol{\tau}_{exc}(t)$ provided the linearity assumption is satisfied. In the case of forward speed, additional linear terms appear in Equation 32.21. This is further discussed in Section 32.4.4.

* Note that this added mass matrix is different than the one used in maneuvering—Section 32.3. We will explain the difference in Section 32.4.4.

32.4.3 Frequency-Domain Seakeeping Models

When Equation 32.21 is considered in the frequency domain, it can be expressed in the following form (Newman, 1977; Faltinsen, 1990):

$$(-\omega^2[\mathbf{M}+\mathbf{A}(\omega)]-j\omega\mathbf{B}(\omega)+\mathbf{G})\boldsymbol{\eta}(j\omega)=\boldsymbol{\tau}_{exc}(j\omega), \tag{32.22}$$

where $\boldsymbol{\eta}(j\omega)$ and $\boldsymbol{\tau}_{exc}(j\omega)$ are the complex response and excitation variables:

$$\eta_i(t)=\bar{\eta}_i\cos(\omega t+\epsilon_i)\Longrightarrow\eta_i(j\omega)=\bar{\eta}_i\exp(j\epsilon_i)$$
$$\tau_i(t)=\bar{\tau}_i\cos(\omega t+\varepsilon_i)\Longrightarrow\tau_i(j\omega)=\bar{\tau}_i\exp(j\varepsilon_i).$$

The parameters $\mathbf{A}(\omega)$ and $\mathbf{B}(\omega)$ are the frequency-dependent added mass and damping respectively.

Equation 32.22 is also commonly written in a mixed frequency–time–domain form:

$$[\mathbf{M}+\mathbf{A}(\omega)]\ddot{\boldsymbol{\eta}}+\mathbf{B}(\omega)\dot{\boldsymbol{\eta}}+\mathbf{G}\boldsymbol{\eta}=\boldsymbol{\tau}_{exc}. \tag{32.23}$$

This unfortunate form is rooted deeply in the literature of marine hydrodynamics and the abuse of notation of this false time-domain model has been discussed eloquently in the literature (Cummins, 1962). The reader is warned that Equation 32.23 is not a time-domain model, rather a different way or writing Equation 32.22, which is a frequency response function. The corresponding time–domain model is given by Equation 32.21.

Expression 32.22 provides the *frequency response from force to displacement*,

$$\boldsymbol{\eta}(j\omega)=\mathbf{G}(j\omega)\boldsymbol{\tau}_{exc}(j\omega),$$

that is,

$$\mathbf{G}(j\omega)=[-\omega^2(\mathbf{M}_{RB}+\mathbf{A}(\omega))+j\omega\mathbf{B}(\omega)+\mathbf{G}]^{-1}=\begin{bmatrix}G_{11}(j\omega) & \cdots & G_{16}(j\omega)\\ \vdots & & \vdots\\ G_{61}(j\omega) & \cdots & G_{66}(j\omega)\end{bmatrix}.$$

Similarly, there exists a *frequency response from wave elevation to wave-excitation force*,

$$\boldsymbol{\tau}_{exc}(j\omega)=\mathbf{F}(j\omega,\chi)\,\zeta(j\omega),$$

where χ is the angle at which the waves approach the vessel—see Section 32.4.1, and

$$\mathbf{F}(j\omega,\chi)=\begin{bmatrix}F_1(j\omega,\chi) & \cdots & F_6(j\omega,\chi)\end{bmatrix}^T.$$

The latter frequency response is known as the *force response amplitude operator* (FRAO) in the naval architecture literature. By combining the above frequency responses we obtain the wave-to-motion frequency response, which is known as the *motion response amplitude operator* (MRAO),

$$\boldsymbol{\eta}(j\omega)=\mathbf{H}(j\omega,\chi)\zeta(j\omega),$$

where

$$\mathbf{H}(j\omega,\chi)=\mathbf{G}(j\omega)\,\mathbf{F}(j\omega,\chi).$$

Hydrodynamic codes based on potential theory are nowadays readily available for the computation of the frequency-dependant added mass, $\mathbf{A}(\omega)$, and potential damping, $\mathbf{B}(\omega)$, and therefore the force and motion RAO. These codes use information about hull geometry and weight distribution to compute the coefficients and responses for finite set of frequencies.

If we combine the RAO with the wave spectrum, we can compute the force and motion spectra:

$$S_{\tau\tau}(j\omega)=|\mathbf{F}(j\omega,\chi)|^2S_{\zeta\zeta}(j\omega), \tag{32.24}$$
$$S_{\eta\eta}(j\omega)=|\mathbf{H}(j\omega,\chi)|^2S_{\zeta\zeta}(j\omega). \tag{32.25}$$

These spectra can be used to compute wave force and motion statistics and also the time series for simulations.

32.4.4 Time-Domain Model Approximations

By taking the Fourier transform of Equation 32.21 and comparing it with Equation 32.22, it can be shown that

$$\mathbf{A}(\omega) = \bar{\mathbf{A}} - \frac{1}{\omega}\int_0^\infty \mathbf{K}(t)\ \sin(\omega t)\, dt,$$
$$\mathbf{B}(\omega) = \int_0^\infty \mathbf{K}(t)\ \cos(\omega t)\, dt.$$

From these expressions, it follows

$$\bar{\mathbf{A}} = \lim_{\omega\to\infty} \mathbf{A}(\omega). \tag{32.26}$$

Expression 32.26 indicates that $\bar{\mathbf{A}}$ is the infinite-frequency added mass matrix. The models used for maneuvering—see Section 32.3—are low-frequency models. Hence, the added mass $\mathbf{M}_A$ in maneuvering models is related to $\mathbf{A}(0) \neq \mathbf{A}(\infty)$.

It also follows from the Fourier transform that

$$\mathbf{K}(j\omega) = \mathbf{B}(\omega) + j\omega[\mathbf{A}(\omega) - \bar{\mathbf{A}}]. \tag{32.27}$$

As commented in the previous section, hydrodynamic codes can be used to compute $\mathbf{A}(\omega)$, and potential damping, $\mathbf{B}(\omega)$ for a discrete set of frequencies. This information can be used together with Equation 32.27 to obtain rational transfer functions that approximate the convolution integral in Equation 32.21, that is

$$\boldsymbol{\mu} = \int_0^t \mathbf{K}(t-t')\boldsymbol{\nu}(t')\, dt' \quad \approx \quad \mathbf{K}(s) \quad \Leftrightarrow \quad \begin{array}{l} \dot{\mathbf{x}} = \mathbf{A}'\mathbf{x} + \mathbf{B}'\,\boldsymbol{\nu} \\ \boldsymbol{\mu} = \mathbf{C}'\mathbf{x} \end{array} \tag{32.28}$$

From potential theory, it can be shown (Perez and Fossen, 2008b) that

$$\lim_{\omega\to 0} \mathbf{K}(j\omega) = \mathbf{0},$$
$$\lim_{\omega\to\infty} \mathbf{K}(j\omega) = \mathbf{0},$$
$$\lim_{t\to 0^+} \mathbf{K}(t) \neq \mathbf{0},$$
$$\lim_{t\to\infty} \mathbf{K}(t) = \mathbf{0},$$
$$\boldsymbol{\nu} \mapsto \boldsymbol{\mu} \quad \text{is passive.}$$

These properties translate into the following constraints on the rational approximations $\hat{K}_{ik}(s) = P_{ik}(s)/Q_{ik}(s)$:

$$\hat{K}_{ik}(s) \text{ has a zero at } s = 0, \tag{32.29}$$
$$\hat{K}_{ik}(s) \text{ has relative degree 1}, \tag{32.30}$$
$$\hat{K}_{ik}(s) \text{ is stable}, \tag{32.31}$$
$$\hat{K}_{ik}(s) \text{ is at least of order 2}, \tag{32.32}$$
$$\hat{K}_{ik}(s) \text{ is positive real for } i = k. \tag{32.33}$$

The constraints (Equations 32.29 through 32.33) can be easily enforced if one performs the identification in the frequency domain using the nonparametric data (Equation 32.27). Figure 32.9 shows the bow half-hull of a semisubmersible offshore rig used to compute hydrodynamic data with the code WAMIT. These data are part of a demo of the Marine Systems Simulator (MSS) available at www.marinecontrol.org. Figure 32.10 shows the results of frequency-domain identification incorporating constraints for the cou-

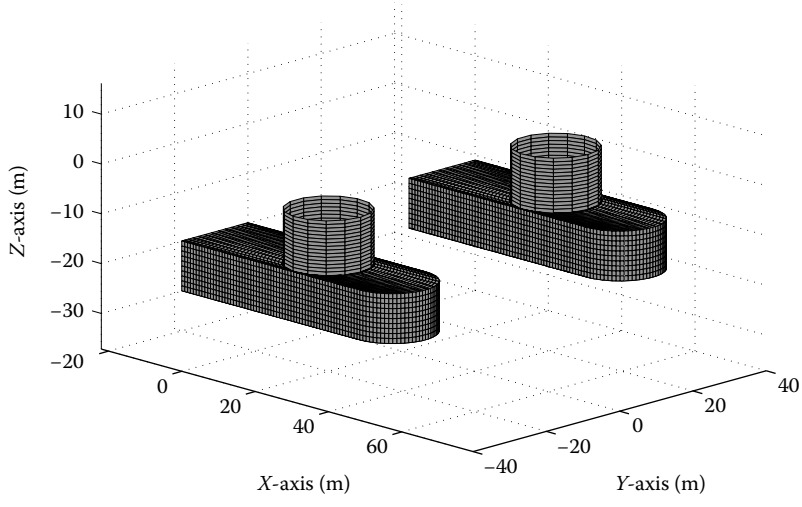

FIGURE 32.9 Hull geometry of a semisubmersible. (Data from www.marinecontrol.org.)

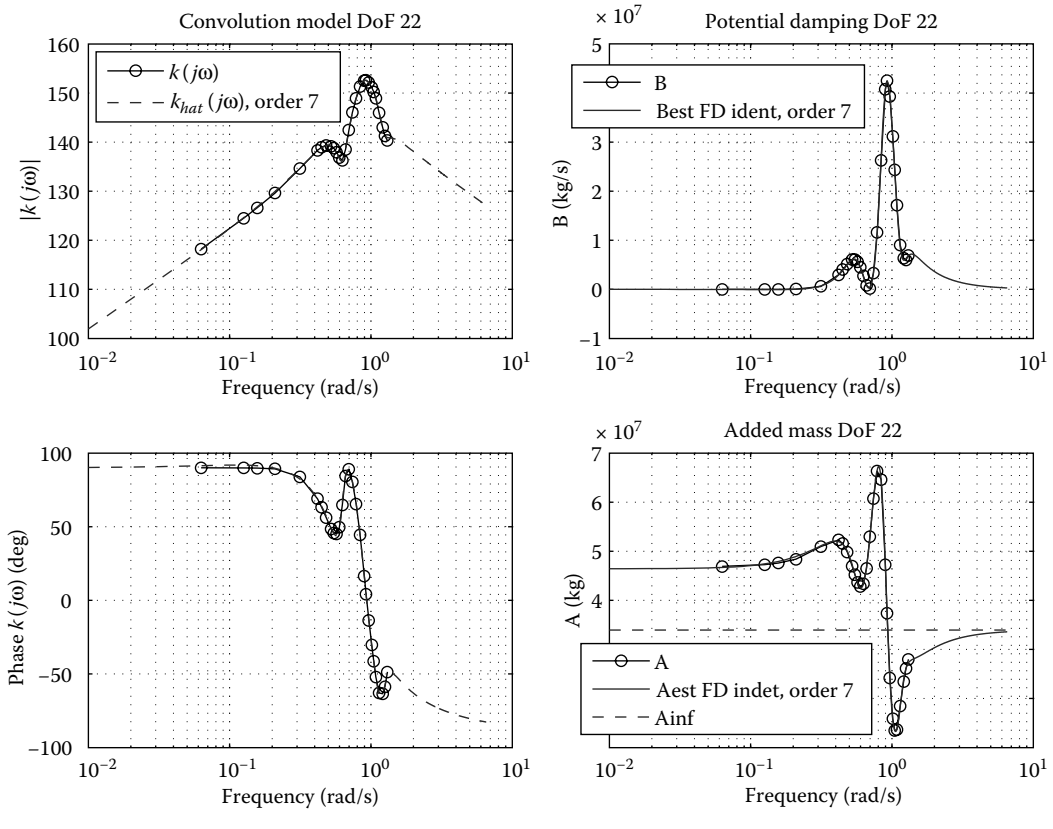

FIGURE 32.10 Data fit corresponding to the coupling 2-2 of a semisubmersible. The left-hand side plots show the magnitude and phase of $K_{22}(j\omega)$ and $\hat{K}_{22}(j\omega)$ for a parametric approximation of order 7. The right-hand side plots show damping and added mass computed by the code and the approximations based on $\hat{K}_{22}(j\omega)$.

plings 2-2 (sway-sway). The left-hand side plots show the magnitude and phase of the convolution frequency response and the right-hand side plots show the potential damping added mass. For further details about identification of radiation force models, see Perez and Fossen (2008a,b 2009).

Time-domain models can be obtained from Equation 32.21 and replacing the convolution by an approximating LTI system Equation 32.28. This model, however, needs to be augmented with viscous damping:

$$(\mathbf{M} + \bar{\mathbf{A}})\,\dot{\boldsymbol{\nu}} + \mathbf{D}_{vis}\,\boldsymbol{\nu} + \boldsymbol{\mu} + \mathbf{G}\,\boldsymbol{\eta} = \boldsymbol{\tau}_{exc}, \tag{32.34}$$

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}'\mathbf{x} + \mathbf{B}'\,\boldsymbol{\nu} \\ \boldsymbol{\mu} &= \mathbf{C}'\mathbf{x}. \end{aligned} \tag{32.35}$$

Potential theory only gives the radiation damping, which reflects the energy carried away by the waves that are generated as a consequence of craft motion—this is captured in the model by Equation 32.35. The radiation damping is only a part of the total damping. Unfortunately, there are no empirical rules to incorporate damping, except for the degree of freedom of roll (Ikeda, 2004). Hence, one needs to use experimental data to compute such damping.

If the vessel has forward speed U, the seakeeping model becomes:

$$(\mathbf{M} + \bar{\mathbf{A}})\,\dot{\boldsymbol{\nu}} + \mathbf{D}_{vis}\,\boldsymbol{\nu} + \mathbf{C}^*\,\boldsymbol{\nu} + \mathbf{D}^*\,\boldsymbol{\nu} + \boldsymbol{\mu} + \mathbf{G}\,\boldsymbol{\eta} = \boldsymbol{\tau}_{exc}, \tag{32.36}$$

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}'\mathbf{x} + \mathbf{B}'\,\delta\boldsymbol{\nu}, \\ \boldsymbol{\mu} &= \mathbf{C}'\mathbf{x}, \end{aligned} \tag{32.37}$$

where $\mathbf{C}^*$ and $\mathbf{D}^*$ are linear additional damping and Coriolis-centripetal terms that appear due to forward speed U and the kinematic transformation between the equilibrium frame in which the forces are computed and the body frame. For further details, see Perez and Fossen (2007) and Fossen (2002).

32.4.5 Time-Domain Wave Excitation

In order to generate realizations of the wave-excitation forces, the spectrum (Equation 32.24) can be used. Indeed, since the wave elevation is Gaussian and considered stationary, and the force response being considered is linear, the response is also Gaussian and stationary. There are different approaches to generate realizations from the spectrum. One approach consists of making a spectral factorization of Equation 32.24 and approximating the realizations as filtered white noise. This approach is commonly used in stochastic control theory. Another approach, commonly used in naval architecture, consists of using a multisine signal. For example, for any component of $\boldsymbol{\tau}$, we can generate realizations via

$$\tau_i(t) = \sum_{n=1}^{N} \bar{\tau}_n \cos(\omega_n t + \varepsilon_n),$$

with N being sufficiently large, where $\bar{\tau}_n$ are constants, and the phases ε_n are independent identically distributed random variables with uniform distribution in $[0,\ 2\pi]$. This choice of random phases ensures that $\tau_i(t)$ is a Gaussian process, and for each realization of the phases, we obtain a realization of the process (St Denis and Pierson, 1953).

The autocorrelation of the process for lag zero satisfies

$$\int_0^\infty S_{\tau\tau}(\omega)\,d\omega \approx \sum_{n=1}^{N} \frac{\bar{\tau}_n^2}{2},$$

from which we can take

$$\bar{\tau}_n = \sqrt{2S_{\tau\tau}(\omega^*)\Delta\omega},$$

where ω^* is chosen randomly within the interval $[\omega_n - \frac{\Delta\omega}{2}, \omega_n + \frac{\Delta\omega}{2}]$.

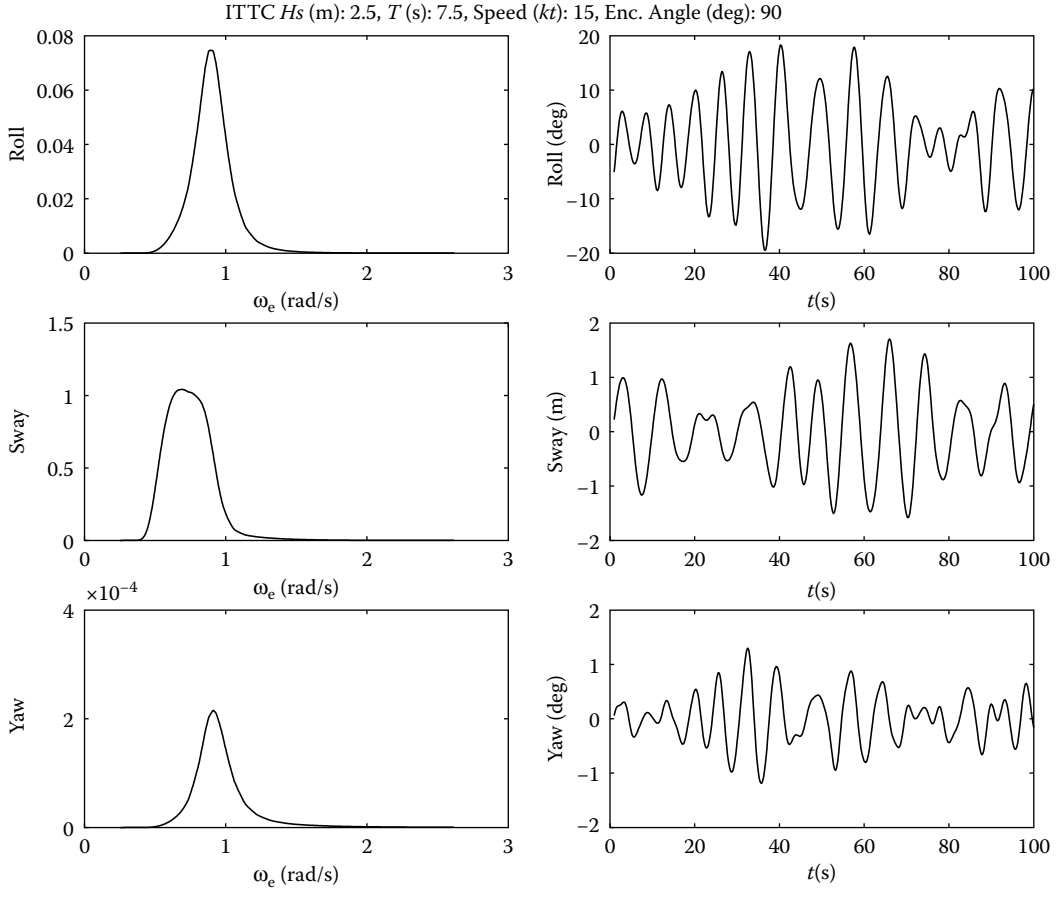

FIGURE 32.11 Roll sway and yaw motion power spectral densities and time series of a navy vessel for beam seas at 15kts. The wave spectrum used is ITTC with $Hs = 2.5$ m and $T = 7.5$ s.

Note that this method can be used not only for wave forces but also for the wave elevation and for the motion of the vessel itself. For example, Figure 32.11 shows the spectra of motion for roll, sway and yaw of a navy vessel in a beam sea sailing condition, and also particular realizations of motion simulated using the multisine. For further details, see Perez (2005).

32.5 Models for Maneuvering in a Seaway

There are applications in which it is necessary to consider models for maneuvering in waves. The hydrodynamic interactions in these situations are rather complex, and to date there is no theory that unifies maneuvering and seakeeping. Models for control design and testing can be built in different ways depending on the data available. One can consider the following options:

1. Linear seakeeping model augmented with nonlinear damping terms.
2. Maneuvering model with wave force excitation (force superposition, input disturbance).
3. Maneuvering model with wave motion excitation (motion superposition, output disturbance).

The first option consists of using (Equations 32.36 through 32.35) and augmenting it with nonlinear damping terms. This option is used when there is no data from the vessel other than the frequency

responses computed by the hydrodynamic codes—the viscous damping has to be added empirically. This approach can give the control designer an initial model to start a design. Though the seakeeping model is not for maneuvering, the Coriolis and centripetal terms will be incorrectly represented. Hence, one should be aware of the validity of the model—only mild maneuver should be attempted with this model.

The second and third option can be used when a maneuvering model is available (Equations 32.14 and 32.15), which has been obtained from either model scale or full scale trials, and we also have access to frequency responses computed by hydrodynamic codes. The frequency responses can be used in conjunction with the adopted wave spectrum to compute either the force or the motion response spectra (Equations 32.24 and 32.25), from which we can simulate realizations of either the wave-induced forces or motion. Some of the literature argues that using force superposition is a more natural approach than using motion superposition. However, the motion superposition model is more accurate in terms of describing the wave-induced motion.

32.6 Design Aspects of Vehicle Motion Control Systems

In this section, we discuss aspects that are fundamental to marine system control design. We first look at the observer design and the control allocation, and then describe the main characteristics of the most common motion control problems.

32.6.1 Observers and Wave Filtering

As discussed in Section 32.1, the frequency of oscillations of the linear wave forces do not normally affect the operational performance of vessels. Hence, controlling only low-frequency motion avoids correcting the motion for every single wave, which can result in unacceptable operational conditions for the propulsion system due to power consumption and potential wear of the actuators.

The control of only low-frequency motion is achieved by appropriate filtering of the wave-frequency components from the position and heading measurements and estimated velocities before the signals are passed on to the controller. This filtering process is known as *wave filtering*.

Early course-keeping autopilots used a proportional (P) controller with a deadband nonlinearity. The deadband provided an effect similar to wave filtering since it delivered a null control action until the control signals were large enough to be outside the deadband. The amount of deadband in the autopilot could be changed, and this setting was called *weather* since the size of the deadband was selected by the operator based on weather conditions. Other systems used lowpass and notch filters, which introduced significant phase lag, and thus performance degradation when a high-gain control is required. An alternative to traditional filtering consists of using a wave-induced motion model and an observer to separate the wave motion from the low-frequency motion. This is depicted in the block diagram shown in Figure 32.1.

The design of observers for positioning and course keeping is normally approached within the linear framework by augmenting the model with an output disturbance model that represents the wave-induced motion. Let us consider here the problem for positioning of surface vessels, that is, we consider maneuvering at low speed in the degrees of freedom of surge, sway, and yaw,

$$\boldsymbol{\eta} = [N,\ E,\ \psi]^T,$$

$$\boldsymbol{\nu} = [u,\ v,\ r]^T.$$

Because of the slow maneuvering assumption, the nonlinear damping and Coriolis and centripetal terms in Equations 32.14 and 32.15 can be neglected; thus,

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\,\boldsymbol{\nu}, \tag{32.38}$$

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu}_{rc} = \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{env}, \tag{32.39}$$

where the kinematic transformation reduces to a rotation matrix

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{R}^{-1}(\psi) = \mathbf{R}^T(\psi).$$

The model (Equations 32.38 through 32.39) is still nonlinear due to the kinematic transformation. As discussed in Fossen (2002), this model can be linearized dynamically by introducing the *vessel parallel coordinates*, which are defined in a reference frame fixed to the vessel with axes parallel to the Earth-fixed reference frame. The vessel parallel position coordinates $\boldsymbol{\eta}_p$ are defined by using the transformation

$$\boldsymbol{\eta}_p = \mathbf{R}^T(\psi)\boldsymbol{\eta}, \tag{32.40}$$

where $\boldsymbol{\eta}_p$ is the position-attitude vector expressed in body coordinates. For positioning control applications, rotation about the z-axis is often slow. Therefore, $\dot{r} \approx 0$ and $\dot{\boldsymbol{R}}(\psi) \approx \mathbf{0}$ are good approximations. Consequently, the time-derivative of Equation 32.40 leads to

$$\begin{aligned} \dot{\boldsymbol{\eta}}_p &= \dot{\mathbf{R}}^T(\psi)\boldsymbol{\eta} + \mathbf{R}^T(\psi)\dot{\boldsymbol{\eta}}, \\ &= \dot{\mathbf{R}}^T(\psi)\boldsymbol{\eta} + \mathbf{R}^T(\psi)\mathbf{R}(\psi)\boldsymbol{\nu}, \\ &\approx \boldsymbol{\nu}. \end{aligned} \tag{32.41}$$

Using the vessel parallel coordinates the kinematics are linearized,

$$\begin{aligned} \dot{\boldsymbol{\eta}}_p &= \boldsymbol{\nu}, \\ (\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu}_{rc} &= \boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{env}. \end{aligned}$$

The wave-frequency forces induce motion, and due to the linearity of the model, we can consider the wave-induced motion as an output disturbance. This disturbance can be modeled as filtered white noise,

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \mathbf{A}_w\boldsymbol{\xi} + \mathbf{E}_w\boldsymbol{w}, \\ \boldsymbol{\eta}_w &= \mathbf{C}_w\boldsymbol{\xi}, \end{aligned}$$

which in transfer-function form results

$$\boldsymbol{\eta}_w(s) = \begin{bmatrix} G_{xw}(s) & 0 & 0 \\ 0 & G_{yw}(s) & 0 \\ 0 & 0 & G_{\psi w}(s) \end{bmatrix} \boldsymbol{w}(s),$$

with

$$G_{iw}(s) = \frac{\omega_i^2\, s}{s^2 + 2\,\zeta_i\,\omega_i s + \omega_i^2}.$$

The low-frequency wave forces, wind, and the viscous component of the current forces, can all be modeled as constant input disturbances. This leads to the following model,

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_w\boldsymbol{\xi} + \mathbf{E}_w\boldsymbol{w}_1, \tag{32.42}$$

$$\dot{\boldsymbol{\eta}}_p = \boldsymbol{\nu}, \tag{32.43}$$

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} = \boldsymbol{\tau}_{ctrl} + \mathbf{b} + \boldsymbol{w}_2. \tag{32.44}$$

$$\dot{\mathbf{b}} = \boldsymbol{w}_3, \tag{32.45}$$

with measurement

$$\boldsymbol{\eta}_{tot} = \boldsymbol{\eta}_p + \mathbf{C}_w\boldsymbol{\xi} + \boldsymbol{n}. \tag{32.46}$$

The state-noises $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_3$ represent model uncertainty, and the noise $\boldsymbol{n}$ is due to measurement instrumentation.

By combining Equations 32.42 through 32.46 we obtain a state-space model of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\,\boldsymbol{\tau}_{ctrl} + \mathbf{E}_{obs}\boldsymbol{w},$$
$$\boldsymbol{\eta}_{tot} = \mathbf{C}\,\mathbf{x} + \boldsymbol{n},$$

where

$$\mathbf{x} = [\boldsymbol{\xi}^T, \boldsymbol{\eta}_p^T, \boldsymbol{\nu}^T, \mathbf{b}^T]^T.$$

With this model one can design an observer, and estimate the state. Then pass to the controller only low-frequency state variables, namely, $\boldsymbol{\eta}_p$, $\boldsymbol{\nu}$, and $\mathbf{b}$.

Figure 32.12 shows simulation data of a Kalman wave filter design for fishing vessel under positioning control (Fossen and Perez, 2009). In this simulation, the wave filter is switched on 120 s after the vessel is under positioning control. Then, at 200 s the vessel position is changed 10 m forward. Figure 32.12a shows the measured and wave-filtered surge position. Figure 32.12b shows the measured and wave-filtered surge velocity. Figure 32.12c shows the force generated by the controller. During the first 120 s, while the wave filter is switched off, the wave-induced motion produces significant control action. Once the wave filter is switched on, the control action at wave frequencies is reduced, which is the effect sought.

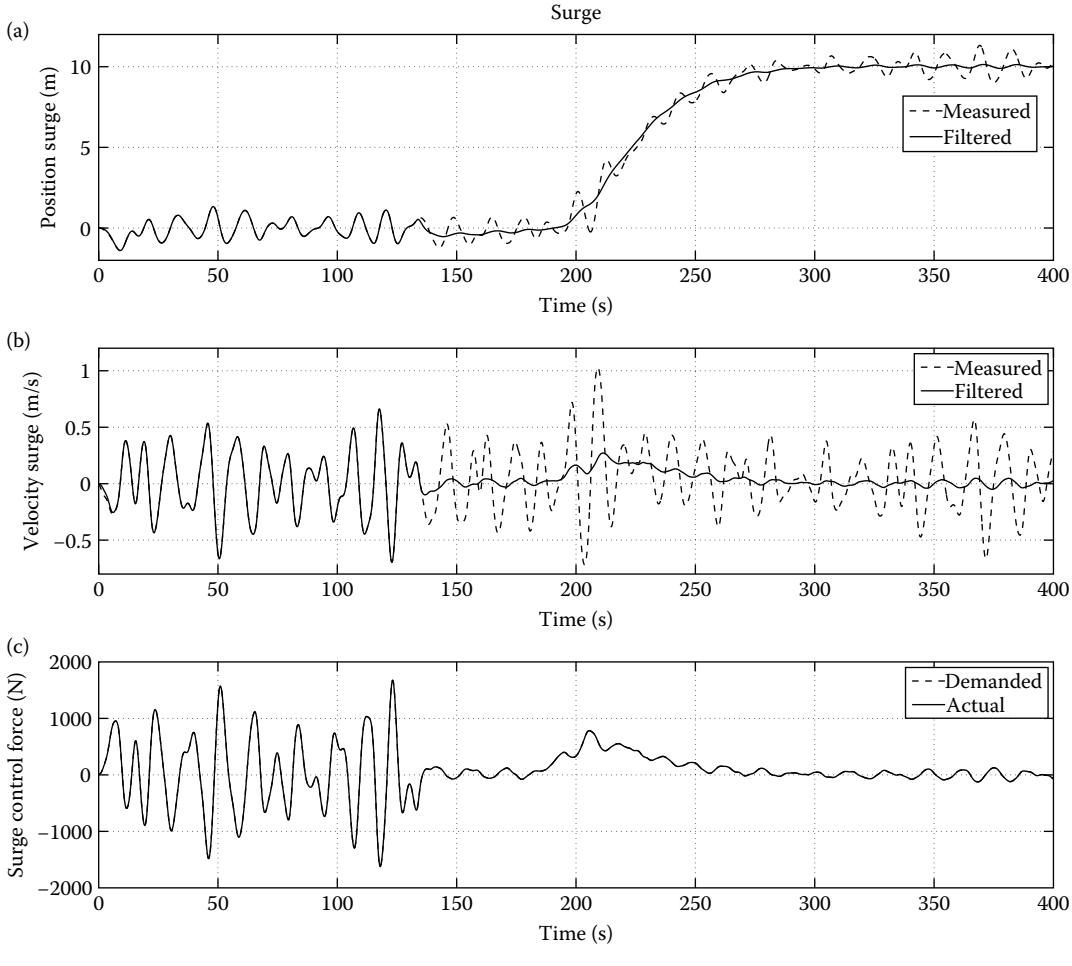

FIGURE 32.12 Wave filter performance for a 15-m fishing vessel under positioning control. The wave filter is switched on at 120 s.

32.6.2 Control Allocation

In order to increase reliability, some marine vehicles are usually equipped with more actuators than the minimum required to control the motion in the desired degrees of freedom. Then, the forces used to control the motion can be produced by different combinations of the forces produced by the actuators. This practice enables the vehicle to continue operation or to start a safe shutdown in case of an actuator failure by reconfiguring the forces produced by the remaining actuators. Within this framework, the craft motion control system is separated into two main components:

- Motion controller
- Control allocation mapping

This is depicted in the block diagram shown in Figure 32.1. The motion controller generates demands for generalized forces in the degrees of freedom in which the vehicle is controlled. Since the motion controller operates in terms of generalized forces, the control design and tuning can be done independently of the actuator configuration to a certain extent. The control allocation mapping then transforms the controller demands into individual actuator commands, such that the demanded generalized forces are implemented.

Each actuator produces a bounded force vector; that is, a vector with a prescribed line of action and point of application:

$$T_i \in \mathcal{S}_i, \qquad i = 1, 2, \ldots, N.$$

Here N identifies the number of actuators. Figure 32.13 shows a schematic diagram of a vehicle with four actuators.

The force vectors take values in the sets $\mathcal{S}_i$, which represent the constraints due to limited force magnitude and direction that each actuator can produce at a particular time.

Due to the location of the actuators on the craft, the forces are mapped into the craft generalized forces via a *thrust configuration matrix*,

$$\boldsymbol{\tau} = \mathbf{B}(\boldsymbol{\alpha})\,\mathbf{T},$$

where

$$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T,$$

and

$$\mathbf{T} = [|T_1|, \ldots, |T_N|]^T.$$

The *control allocation problem* can then be posed as a constrained optimization problem:

$$(\mathbf{T}^\star, \boldsymbol{\alpha}^\star, \mathbf{s}^\star) = \arg\min_{\mathbf{T},\boldsymbol{\alpha},\mathbf{s}} V(\mathbf{T}, \boldsymbol{\alpha}, \boldsymbol{\alpha}_0, \mathbf{s}) \tag{32.47}$$

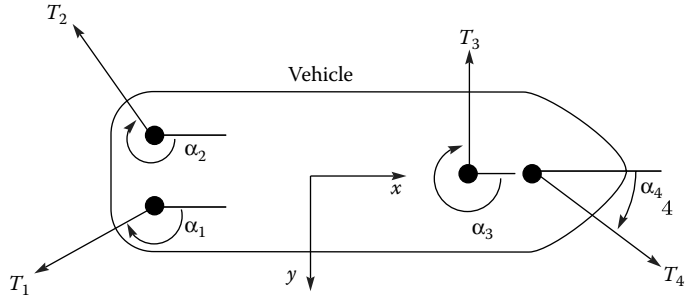

FIGURE 32.13 Vehicle actuator forces.

subject to

$$\mathbf{B}(\boldsymbol{\alpha})\,\mathbf{T} = \boldsymbol{\tau}_d - \mathbf{s} \tag{32.48}$$

$$\mathbf{T}_{min} \leq \mathbf{T} \leq \mathbf{T}_{max}, \tag{32.49}$$

$$\Delta\mathbf{T}_{min} \leq \mathbf{T} - \mathbf{T}_0 \leq \Delta\mathbf{T}_{max}, \tag{32.50}$$

$$\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{max}, \tag{32.51}$$

$$\Delta\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha}_{max}, \tag{32.52}$$

$$\mathbf{s} \geq \mathbf{0}. \tag{32.53}$$

The scalar-valued objective function $V(\cdot,\cdot,\cdot,\cdot)$ in Equation 32.47 relates to the control effort and may contain a barrier term that avoids singular actuator configurations, that is, the configurations in which the capacity to generate forces in some degrees of freedom are lost. For example, if in the vessel depicted in Figure 32.13 all the actuators direct their forces forward, then it is not possible to generate either side force or a turning moment; hence this is a singular configuration (in this case, the thrust configuration matrix loses rank).

The constraint (Equation 32.48) ensures that the desired generalized forces demanded by the controller, $\boldsymbol{\tau}_d$, are implemented. The remaining constraints are related to magnitude and angle of the forces and also their rates, which are set in terms of differences from given values $\mathbf{T}_0$ and $\boldsymbol{\alpha}_0$.

Due to the limited authority of the actuators, that is, the constraints (Equations 32.49 through 32.52), it may happen that the generalized force vector demanded by the controller is not feasible. To avoid infeasibility of the optimization problem, the slack variable $\mathbf{s}$ is included in Equation 32.48 with the constraint (Equation 32.53). If the demanded control vector is feasible, then $\mathbf{s}^\star = 0$.

The above optimization problem is solved online at each sampling instant; hence, the values $\mathbf{T}_0$ and $\boldsymbol{\alpha}_0$ correspond to the solution $\mathbf{T}^\star$ and $\boldsymbol{\alpha}^\star$ obtained in the preceding sampling instant. For further details on control allocation for marine vehicles, see Fossen et al. (2009) and references therein.

32.6.3 Overview of Vehicle Motion Control Problems

In the following we describe the main marine vehicle motion control problems and their characteristics.

32.6.3.1 Dynamic Positioning and Thruster-Assisted Position Mooring

Dynamic positioning refers to the use of the propulsion system to hold the vehicle's position despite environmental disturbances. For surface vessels, this problem involves horizontal position and heading regulation. that is surge, sway, and yaw. This type of control problem is common in offshore vessels and oil rigs. The objective is to control only the low-frequency motion. Therefore, the control system requires wave filtering. Since offshore vessels perform critical operations, they are over-actuated. Therefore, the controller generates generalized force commands and there is a control allocation mapping. The structure of the controller often consists of a velocity and a position loop. The control design can be approached as an optimal control problem for set point regulation. The implementation of the controller is done using proportional-integral (PI) and proportional–integral–derivative (PID) controllers. In Section 32.7, we provide an example. For underwater vehicles, the positioning problem extends to auto-depth. Since the vertical plane motion is often decoupled from the horizontal plane motion and there are specific actuators to control each type of motion, the two positioning problems can often be designed independently. In a thruster-assisted position mooring, the propulsion system is used to compensate the mean loading on mooring lines. This control problem is similar to that of dynamic positioning.

32.6.3.2 Autopilots

Autopilots used in surface and underwater vehicles are alike. Their degree of sophistication can vary from simple course keeping to maneuvering control and can sense and avoid functionality. For surface

vessels, the problem is considered in one degree of freedom (yaw) for a course autopilot controller, but the guidance system takes into account the position as well as the heading angle. Typical controller implementations consist of PID controllers with feedforward functionality. A turn rate loop is incorporated. The reason for it is that the designs should avoid direct PID control from heading angle since the resulting loop transfer function would have a double integrator, and therefore a step response will always overshoot. Autopilots for surface vessels do not normally require control allocation. Wave filtering is a feature of autopilots for large ocean-crossing vessels, hence there is no rudder correction for every single wave. Autopilots for maneuvering control require nonlinear control designs.

32.6.3.3 Ride Control

Ride control refers to the use of motion control to reduce the wave-induced motion in roll and pitch. These control systems are characteristics of vessels that carry passengers and goods. Local accelerations due to roll and pitch are the main causes of motion sickness, which can produce cargo damage, and prevent the use of equipment on board. The control objectives are to reject the wave-frequency motion without affecting the steering capability of the vessel. Since the spectrum or roll and pitch can vary significantly with the sea state and sailing condition, the control systems often have to be adaptive to maximize performance over a wide envelope of conditions. When combined roll and pitch control is required, control allocation may be used as some of the specific actuators used can generate both pitch and roll, such as T-foils, interceptors and trim flaps.

As a final general remark on control design, we should mention that the models of marine vehicles present a significant degree of uncertainty, and the dynamic response changes with sea state, velocity, wave direction, water depth, and other vessels in close proximity. Since vehicle motion follows physical laws of energy, passivity-based control designs have been very successful. A controller designed such that the stability depends only on dissipativity properties can result in closed-loop stability even under large parametric uncertainty—even changes in model order may be tolerated provided that dissipativity properties remain unchanged. For further details, see Fossen (2002). We next present two control design examples.

32.7 Example Positioning Control of a Surface Vessel

We consider the positioning problem in the horizontal plane of a surface vessel, that is, the generalized position, velocity, and force vectors are

$$\boldsymbol{\eta} \triangleq \begin{bmatrix} N \\ E \\ \psi \end{bmatrix}, \quad \boldsymbol{\nu} \triangleq \begin{bmatrix} u \\ v \\ r \end{bmatrix}, \quad \boldsymbol{\tau} \triangleq \begin{bmatrix} X \\ Y \\ N \end{bmatrix},$$

and the low-frequency control design model is given by

$$\begin{aligned} \dot{\boldsymbol{\eta}}_p &\approx \boldsymbol{\nu}, \\ \dot{\boldsymbol{\nu}} &= \mathbf{M}^{-1}\mathbf{D}\boldsymbol{\nu} + \mathbf{M}^{-1}(\boldsymbol{\tau}_{ctrl} + \boldsymbol{\tau}_{env}), \end{aligned}$$

where $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$.

The control system block diagram is shown in Figure 32.14. The measured generalized position vector, thus, has a wave- and low-frequency component, which is transformed to vessel parallel coordinates,

$$\boldsymbol{\eta}_{pm} = \boldsymbol{\eta}_{p,WF} + \boldsymbol{\eta}_p.$$

A wave-filter observer is used to estimate the low-frequency position and the velocity vector used to implement the controller. The motion controller consists of a velocity loop implemented with a PI

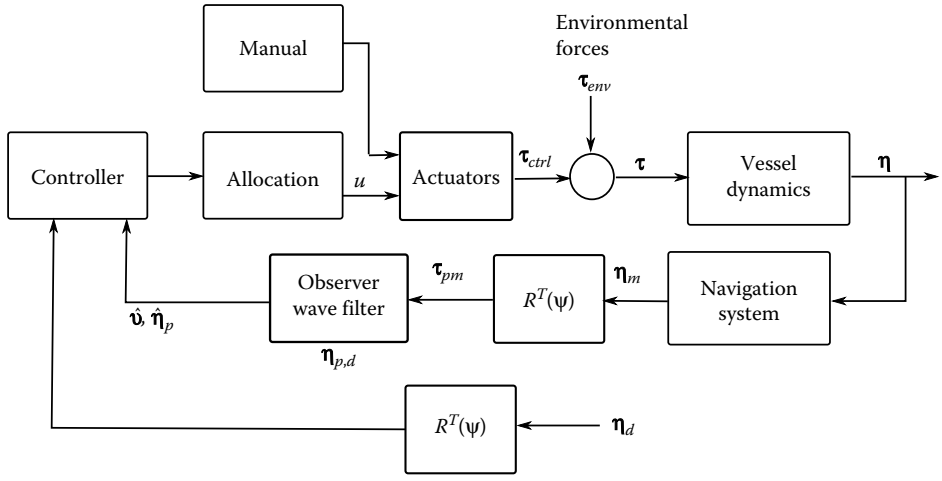

FIGURE 32.14 Block diagram of a dynamic positioning control system.

controller and a position loop implemented with a P controller. The structure of the controller is illustrated in the block diagram shown in Figure 32.15. This controller implements integral action for both position and velocity, which rejects low-frequency force disturbances (Perez and Donaire, 2009).

32.7.1 Unconstrained Control Allocation

The force vectors produced by the actuators can be decomposed into rectangular coordinates (along the longitudinal and transverse direction of the vessel) and combined into a single vector **T**,

$$\mathbf{T} = \begin{bmatrix} T_{X1} & T_{Y1} & T_{X2} & T_{Y2} \ldots T_{XN} & T_{YN} \end{bmatrix}^T,$$

where N is the number of actuators. This vector is mapped into the generalized forces via the actuator configuration matrix **B** (which for the sake of simplicity we had fixed):

$$\boldsymbol{\tau} = \mathbf{B}\,\mathbf{T}. \tag{32.54}$$

Since **T** has more components than $\boldsymbol{\tau}$, there are different vectors **T** satisfying (Equation 32.54) a given value of $\boldsymbol{\tau}$. In order to limit the number of solutions, one can pose the problem as an optimization problem, for example,

$$\begin{aligned} &\mathbf{T}^\star = \arg\min_{\mathbf{u}}(\mathbf{T}^T\mathbf{W}\mathbf{T}) \\ &\text{subject to } \boldsymbol{\tau}_d = \mathbf{B}\mathbf{T}. \end{aligned} \tag{32.55}$$

The objective function $\mathbf{T}^T\mathbf{W}\mathbf{T}$ is representative of the total energy or control effort, where **W** is a positive definite matrix weighting the relative cost of using different actuators. Thus, the control allocation seeks

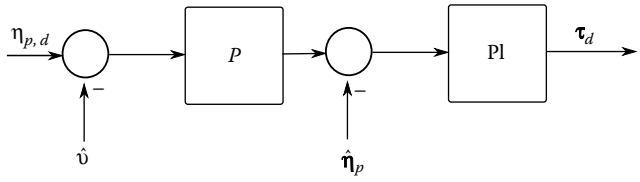

FIGURE 32.15 Proposed positioning controller.

the solution that implements the desired generalized force $\boldsymbol{\tau}_d$ whilst minimizing the control effort. As shown in Fossen (2002), the solution of the above problem is given by

$$\mathbf{T}^\star = \mathbf{B}^\dagger \boldsymbol{\tau}_d, \quad \mathbf{B}^\dagger = \mathbf{W}^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^T)^{-1}. \tag{32.56}$$

Note that since $\mathbf{B}$ depends only on the location of the actuators on the vehicle, the right inverse $\mathbf{B}^\dagger$ can be precomputed.

The optimization problem Equation 32.55 does not take into account the fact that the vector $\mathbf{T}$ must belong to a constraint set due to the maximum force that the various actuators can produce. Adding this constraint to Equation 32.55 requires on-line numerical optimization as discussed in Section 32.6.2, and the solution (Equation 32.56) is no longer the optimal solution. An alternative to this approach is to constrain the desired generalized force $\boldsymbol{\tau}_d$, such that the constraints on $\mathbf{T}$ are always satisfied. By doing so, the force controller is also informed about reaching constraints, which prevents performance degradation due to the combination of actuator saturation and integral action. This control method can be implemented using multivariable anti-wind up techniques as discussed in the next section.

32.7.2 Constrained Control via Input Scaling

One of the key issues in control design for systems that require integral action and present a potential for actuator saturation is that of an integrator windup. That is, if the actuators saturate and the integral controller is not informed about the saturation, the integrators continue integrating the error signals but the control action is not seen by the system. This often produces a degradation of performance in terms of undesirable oscillations and even instability. Control schemes that deal with this effect are called antiwindup schemes.

If a linear controller $\boldsymbol{C}(s)$ is in minimum phase and bi-proper (as in the case of a PI and PID controller), then antiwindup can be achieved simply by implementation. Goodwin et al. (2001) proposed the implementation shown in Figure 32.16. In this figure, *Lim* represents a saturation (magnitude, rate, or a combination of both), and the gain

$$\boldsymbol{c}_\infty = \lim_{s\to\infty} \boldsymbol{C}(s).$$

Note that if the limitation is not active, the loop of Figure 32.16, reduces to the controller:

$$\boldsymbol{C}(s) = \left[\mathbf{I} + \boldsymbol{c}_\infty \left(\boldsymbol{C}(s)^{-1} - \boldsymbol{c}_\infty^{-1}\right)\right]^{-1} \boldsymbol{c}_\infty.$$

When the limitation becomes active, it prevents the control signal $\boldsymbol{\tau}_d$ from exceeding its limits, and the constrained signal drives the states of the controller, which are all on the feedback path.

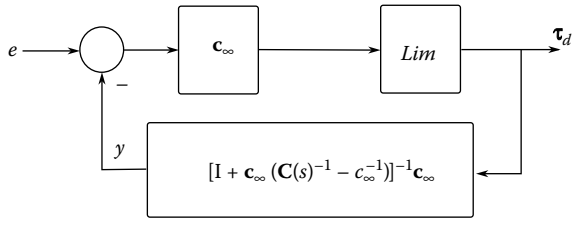

FIGURE 32.16 Block diagram of an anti-wind-up implementation of a strictly proper controller.

The antiwindup scheme described above can be applied to the velocity PI controllers of the velocity loops of the vehicle positioning controller:

$$\boldsymbol{C}_{vel}(s) = \begin{bmatrix} C_u(s) & 0 & 0 \\ 0 & C_v(s) & 0 \\ 0 & 0 & C_r(s) \end{bmatrix},$$

where, for $i = u, v, r$,

$$C_i(s) = K_p^i \frac{T_I^i s + 1}{T_I^i s}.$$

In order to constrain $\boldsymbol{\tau}_d$ we can construct a set $\mathcal{S}_\tau$, such that

$$\boldsymbol{\tau}_d \in \mathcal{S}_\tau \;\Leftrightarrow\; \mathbf{T} \in \mathcal{S}_T.$$

One way of enforcing the constraint is by computing an unconstrained control $\boldsymbol{\tau}_{uc}$ and then scaling it down if it is outside the constraint set, that is,

$$\boldsymbol{\tau}_d = \begin{cases} \boldsymbol{\tau}_{uc} & \text{if } \boldsymbol{\tau}_{uc} \in \mathcal{S}_\tau \\ \gamma\, \boldsymbol{\tau}_{uc} & \text{if } \boldsymbol{\tau}_{uc} \notin \mathcal{S}_\tau, \gamma < 1 \;:\; \gamma\, \boldsymbol{\tau}_{uc} \in \partial\mathcal{S}_\tau, \end{cases} \tag{32.57}$$

where $\partial\mathcal{S}_\tau$ denotes the boundary of the set $\mathcal{S}_\tau$.

By scaling the vector $\boldsymbol{\tau}_{uc}$, we preserve its direction. By this way of limiting the control command, the block *Lim* in Figure 32.16 can be replaced by

$$Lim \equiv \alpha(t)\, \mathbf{I}.$$

In order to implement Equation 32.57, we need to compute $\boldsymbol{\tau}_{uc}$ and then obtain γ. An algorithm to evaluate Equation 32.57 is given in Perez and Donaire (2009).

In the above proposed control system, the constraint set of actuator forces due to the limited authority of the actuators are mapped into a constraint set for the generalized forces. This approach results in a constrained control design with a simple, unconstrained, control allocation problem. The stability of the closed-loop system can be analytically proven by rearranging the closed-loop system with control allocation into a Lure system and application of Lyapunov and passivity theories (Perez and Donaire, 2009).

32.7.3 Simulation Case Study

To illustrate the performance of the controller, we consider a model of an offshore vessel from Fossen (2002), and we consider the actuator configuration depicted in Figure 32.13, that is,

- Two stern azimuth thrusters, which are set at angles of $\alpha_1 = 135°$, and $\alpha_2 = 225°$.
- One bow tunnel thruster, $\alpha_3 = 90°$.
- One bow deployable azimuth thruster, which is fixed to provide force only along the longitudinal axis of the vessel, $\alpha_4 = 0°$.

Figure 32.17 shows the results of a simulation experiment, in which the vessel is in hold position (regulation), then after 20 s we set a reference set-point change in surge, followed by a set point in sway, and finally one in yaw. This figure shows the demanded and actual surge, sway, and yaw positions, velocities and generalized forces. As we can see from this demanded and actual generalized forces, the antiwindup scheme works such that the demands are feasible. Due to the saturation of the actuators, the velocity demands cannot be followed. Figure 32.18 shows the corresponding forces of the four actuators. As we can see from the latter figure, the forces remain within the constraints on the maximum force magnitude. The control system presents a good performance despite the actuators reaching the saturation levels due to the antiwindup implementation.

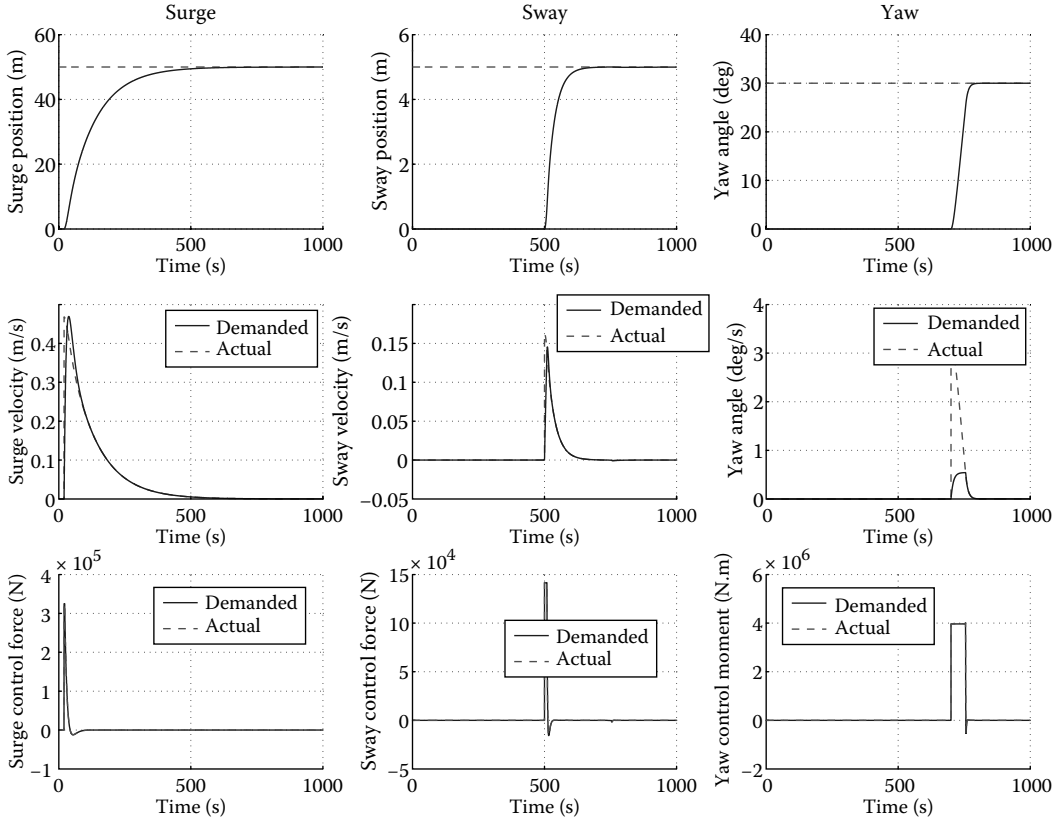

FIGURE 32.17 Performance of vehicle position regulation controller with a position set point change.

32.8 Example: Course Keeping Autopilot for a Surface Vessel

The response in yaw rate due to a small deviation in the angle of a control actuator, such as a rudder or the steering nozzle of a water jet, can be derived from Equation 32.13 by isolating the yaw motion, which is given by

$$(I_{zz}^b - N_{\dot{r}})\dot{r} - N_r r = N_\delta \delta,$$

where I_{zz}^b is the moment of inertia in yaw, $N_{\dot{r}}$, N_r, and N_δ are hydrodynamic coefficients, r is the yaw rate, and δ is the actuator angle (rudder or the steering nozzle of a water jet). This model, which is known as the first-order Nomoto model, can be written as the transfer function

$$\frac{r(s)}{\delta(s)} = \frac{K}{1 + Ts}.$$

The time constant and low-frequency gain are given by

$$T = \frac{I_{zz}^b - N_{\dot{r}}}{-N_r},$$

$$K = -\frac{N_\delta}{N_r},$$

which can be estimated from trials in calm water.

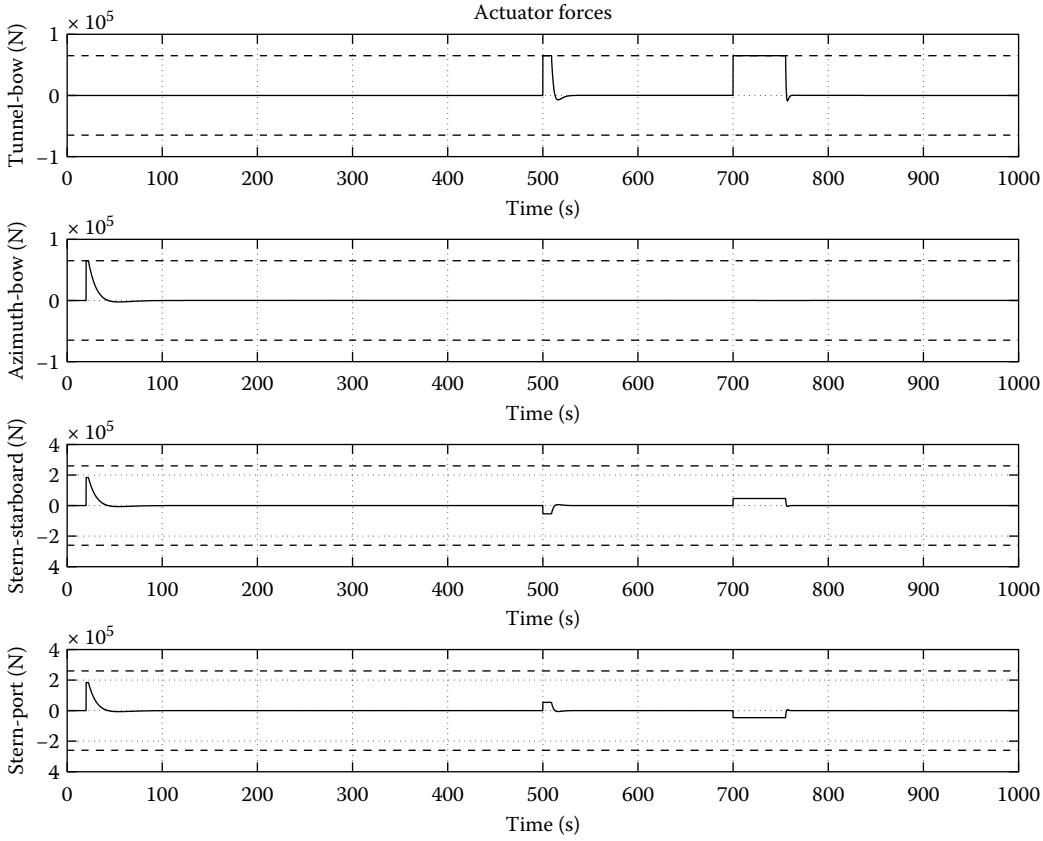

FIGURE 32.18 Actuator forces for a vehicle change in position.

Using the motion superposition assumption, as in the case of positioning control design, we model low-frequency environmental disturbances with a bias moment term in the equation of motion. Then, the state-space model can be written as

$$\begin{aligned}\dot{\psi} &= r,\\ \dot{r} &= -\frac{1}{T}r + \frac{1}{m}\tau_{\mathrm{N}} + b,\\ \dot{b} &= 0,\end{aligned}$$

where $m = I_{zz} - N_{\dot{r}}$ and

$$\tau_{\mathrm{N}} = m\frac{K}{T}\delta = N_{\delta}\,\delta$$

denotes the control yaw moment.

For autopilot control, it is common to design a PID controller with feedforward from wind and a smooth time-varying reference signal $\psi_d(t)$ according to

$$\tau_{\mathrm{N}}(s) = -\hat{\tau}_{wind} + \underbrace{m\left(\dot{r}_d - \frac{1}{T}r_d\right)}_{\tau_{\mathrm{FF}}} - \underbrace{m\left(K_p\tilde{\psi} + K_d\tilde{r} + K_i\int_0^t \tilde{\psi}(\tau)d\tau\right)}_{\tau_{\mathrm{PID}}}, \tag{32.58}$$

where τ_{N} is the controller yaw moment and τ_{FF} is a feedforward term using the reference signal $r_d = \dot{\psi}_d$. The heading and yaw rate errors are denoted by $\tilde{\psi} = \psi - \psi_d$ and $\tilde{r} = r - r_d$, respectively. The control

gains K_p, K_d, and K_i must be chosen such that the third-order linear error dynamics

$$\dddot{\tilde{r}} + \left(\frac{1}{T} + K_d\right)\ddot{\tilde{r}} + K_p\dot{\tilde{\psi}} + K_i\tilde{\psi} = 0$$

are asymtotically stable. The control law (Equation 32.58) depends on the wind-yaw moment estimate $\hat{\tau}_{wind}$, which is used as a feedforward term. This accelerates the response of the autopilot to changes in wind direction and intensity. The wind-yaw moment is modeled as

$$\hat{\tau}_{wind} = \frac{1}{2}\rho_a U_{rw}^2 A_{L_w} L_{oa} C_{N_w}(\gamma_{rw}), \tag{32.59}$$

where ρ_a is the air density, U_{rw} and γ_{rw} are the speed and direction of the wind relative to the vessel, L_{oa} is the overall length of the vessel, and A_{Lw} is a characteristic area exposed to the wind, and $C_{N_w}(\gamma_w)$ is a yaw moment wind coefficient. The uncertainty in Equation 32.59 has a low-frequency content, which is compensated by the integral action of the controller (Fossen, 2002). The rudder command is computed from the control input τ_N as

$$\delta = \frac{1}{N_\delta}\tau_N.$$

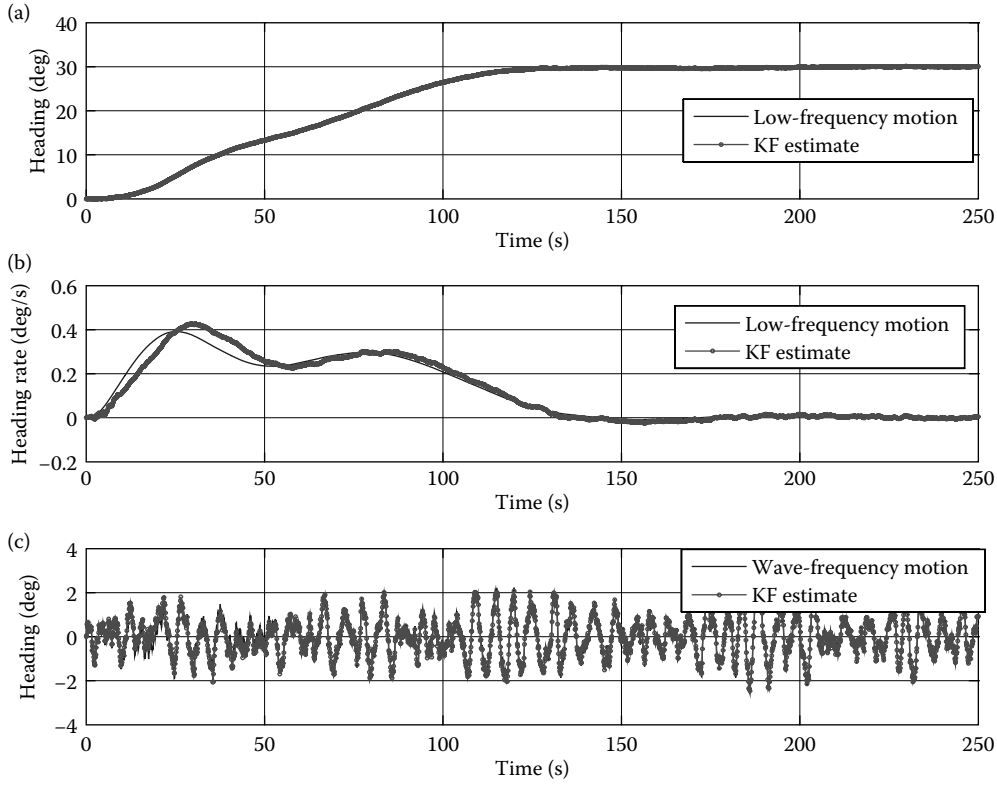

FIGURE 32.19 Performance of a wave filter for heading autopilot for a mariner cargo ship; (a) the true low-frequency heading ψ and Kalman filter estimate $\hat{\psi}$; (b) the true low-frequency heading rate r and its Kalman filter estimate $\hat{r}$; (c) the wave-frequency component of the heading ψ_w and its Kalman filter estimate $\hat{\psi}_w$. The Kalman filter uses measurements of the sum of low- and wave-frequency heading and estimates these states and the rate using a model of the vessel and a model of the wave-induced motion.

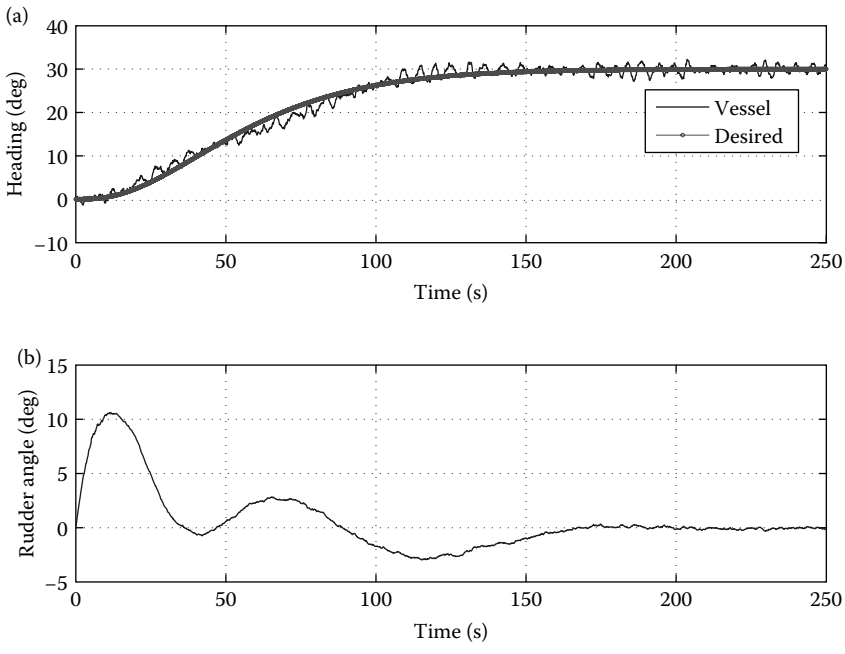

FIGURE 32.20 Performance of a heading autopilot for a mariner cargo ship with a wave filter; (a) the time series of the desired heading ψ_d and the actual vessel heading ψ; (b) the rudder angle δ. As the low-frequency estimates of the heading angle and rate produced by the Kalman filter alone are passed to the controller, the motion of the rudder does not respond to the wave motion, and thus the wave filtering is achieved.

In order to implement the control law, both ψ and r are needed. Most of the marine crafts have only compass measurements ψ, and thus the turning rate r must be estimated. In addition, it is necessary to perform wave filtering such that the oscillatory wave-induced motions are avoided in the feedback loop.

To illustrate the performance of a course keeping autopilot with wave filtering, we consider the case of an autopilot application taken from the MSS (MSS, 2010). This simulation package implemented in MATLAB® and Simulink® provides models of vessels and a library of simulink blocks for heading autopilot control system design and blocks for a Kalmanfilter-based wave filter from heading-only measurements. The vessel considered is a 160-m mariner class vessel. From the step tests performed on the nonlinear model, a first-order Nomoto model (Equation 32.8) is identified. Figures 32.19 and 32.20 show the performance of the wave filter. Figure 32.19a and b show the true low-frequency heading angle and rate together with the Kalman filter estimates. Figure 32.19c shows the first-order wave-induced heading angle component and its estimate. Figure 32.20 shows the performance of the control loop. Figure 32.20a shows the desired and the actual heading angle of the controlled vessel. Figure 32.20b depicts the rudder angle. In this figure, we can appreciate the effect of the wave filtering since the rudder angle has no motion at the wave frequency.

32.9 Conclusion

Marine craft perform operations that require tight motion control. During the past three decades, there has been an increasing demand for higher accuracy and reliability of marine craft motion control systems. Today, these control systems are an enabling factor for single and multicraft marine operations. This chapter provides an overview of the main aspects of marine craft motion control systems and their

designs. In particular, we discuss the architecture of the control system, the functionality of its main components, the characteristics of environmental disturbances and their influence on control objectives, and the essential aspects of modeling and motion control design. For further details on marine craft motion control, see Fossen (2002) and Perez (2005).

References

Abkowitz, M., 1964. Lecture notes on ship hydrodynamics—Steering and manoeuvrability. Technical report Hy-5, Hydro- and Aerodynamics Laboratory, Lyngby, Denmark.

Blanke, M., 1981. Ship propulsion losses related to automatic steering and prime mover control. Ph.D. thesis, Servolaboratory, Technical University of Denmark.

Cummins, W., 1962. The impulse response function and ship motion. Technical Report 1661, David Taylor Model Basin—DTNSRDC.

Egeland, O. and J. Gravdahl, 2002. *Modeling and Simulation for Automatic Control*. Marine Cybernetics, Trondheim.

Faltinsen, O., 1990. *Sea Loads on Ships and Offshore Structures*. Cambridge University Press, Cambridge.

Fedyaevsky, K. and G. Sobolev, 1964. *Control and Stability in Ship Design*. State Union Shipbuilding, Leningrad.

Fedyaevsky, K. and G. Sobolev, 1964. *Control and Stability in Ship Design*. State Union Shipbuilding, Leningrad.

Fossen, T., 2002. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, Trondheim.

Fossen, T. I., T. A. Johansen, and T. Perez, 2009. *A Survey of Control Allocation Methods for Underwater Vehicles*, chap. 7, pp. 109–128. In-Tech, Vienna, Austria.

Fossen, T. and T. Perez, 2009. Kalman filtering for positioning and heading control of ships and offshore rigs. *IEEE Control Systems Magazine* **29**(6):32–46.

Goldstein, H., 1980. *Classical Mechanics*. Addison-Wesley, Reading, MA.

Goodwin, G. C., D. E. Quevedo, and E. L. Silva, 2001. *Control System Design*. Prentice-Hall, Inc., New Jersey.

Ikeda, Y., 2004. Prediction methods of roll damping of ships and their application to determine optimum stabilisation devices. *Marine Technology* **41**(2):89–93.

Kirchhoff, G., 1869. Uber die Bewegung eines Rotationskorpers in einer Flüssigkeit. *Crelle* **71**:237–273.

Lamb, H., 1932. *Hydrodynamics*, 6th ed, Cambridge University Press, Cambridge.

MSS, 2010. *Marine Systems Simulator*. Viewed 29/1/2010, http://www.marinecontrol.org.

Newman, J. N., 1977. *Marine Hydrodynamics*. MIT Press.

Ochi, M., 1998. *Ocean Waves: The Stochastic Approach*. Ocean Technology Series. Cambridge University Press, Cambridge.

Perez, T., 2005. *Ship Motion Control. Advances in Industrial Control*. Springer-Verlag, London.

Perez, T. and A. Donaire, 2009. Constrained control design for dynamic positioning of marine vehicles with control allocation. *Modelling Identification and Control, The Norwegian Society of Automatic Control*, **30**(2):57–70, doi:10.4173/mic.2009.2.2.

Perez, T. and T. Fossen, 2007. Kinematic models for seakeeping and manoeuvring of marine vessels at zero and forward speed. *Modelling Identification and Control, The Norwegian Society of Automatic Control*, **28**(1):19–30, doi: 10.4173/mic.2007.1.3.

Perez, T. and T. Fossen, 2008a. Joint identification of infinite-frequency added mass and fluid-memory models of marine structures. *Modelling Identification and Control, The Norwegian Society of Automatic Control*, **29**(3):93–102, doi:10.4173/mic.2008.1.1.

Perez, T. and T. I. Fossen, 2008b. Time-domain vs. frequency-domain identification of parametric radiation force models for marine structures at zero speed. *Modelling Identification and Control, The Norwegian Society of Automatic Control*, **29**(1):1–19, doi:10.4173/mic.2008.3.2.

Perez, T. and T. I. Fossen, 2009. A MATLAB toolbox for parametric identification of radiation-force models of ships and offshore structures. *Modelling Identification and Control, The Norwegian Society of Automatic Control,* **30**(1):1–15, doi:10.4173/mic.2009.1.1.

Perez, T., T. Mak, T. Armstrong, A. Ross, and T. I. Fossen, 2007. Validation of a 4-DOF manoeuvring model of a high-speed vehicle-passenger trimaran. In *9th International Conference on Fast Sea Transportation (FAST)*, Shanghai, China, September.

Price, W. and R. Bishop, 1974. *Probabilistic Theory of Ship Dynamics*. Chapman & Hall, London.

Ross, A., 2008. Nonlinear maneuvering model based on low-aspect ratio lift theory and lagrangian mechanics. PhD thesis, Department of Engineering Cybernetics.

Ross, A., T. Perez, and T. Fossen, 2007. A novel manoeuvring model based on low-aspect-ratio lift theory and Lagrangian mechanics. In *Proceedings of the IFAC Conference on Control Applications in Marine Systems (CAMS)*. Bol, Croatia, September.

St Denis, M. and W. Pierson, 1953. On the motion of ships in confused seas. *SNAME Transactions* **61**:280–332.

33

Control of Unstable Oscillations in Flows

Anuradha M. Annaswamy
Massachusetts Institute of Technology

Seunghyuck Hong
Massachusetts Institute of Technology

33.1 Introduction

The most common use of feedback control is regulation. By using sensors that monitor the plant, and suitably altering its inputs, the controller regulates the plant outputs around their desired values. A more dramatic application of feedback control is *stabilization*. A plant that is prone to instability can be stabilized using a feedback controller by utilizing online information from the plant outputs and altering key inputs into the plant. Nowhere is this more apparent than in continuous combustion processes and impinging flows.

Continuous combustion processes are typically found in gas turbine engines and high-speed propulsion devices, where fuel and air are mixed and burned to produce thrust. In the process of enhancing the system performance, several turbine manufacturers noticed that the pressure levels reached alarmingly large levels, accompanied by humming or buzzing sounds and large vibrations [1]. The fact that several such incidents were independently reported illustrated the fact that this was not an isolated incident but something that is endemic to the system itself.

This instability phenomenon exhibited by the combustion processes has a long history, starting in the nineteenth century, when a number of independent studies revealed that sound of considerable amplitude can be generated when a gas flame is placed inside a large tube, leading to what has become colloquially known as "dancing" or "singing" flames and to Rayleigh's famous criterion [2]. Rayleigh hypothesized that this instability is due to an unstable dynamic coupling between the heat release emanating from the combustion process and pressure in the combustion chamber. Rayleigh's criterion has, over the years, served as an important analytical tool to predict potentially damaging interactions in combustor designs. As the Wall Street article [1] shows, modern combustion systems exhibit the same phenomenon, indicating that this is very much a "current" problem, exhibited in a wide range of combustion processes including lean premixed combustors [3], ramjet engines [4], and

pulsed combustors [5]. These are used for low emissions, high-speed propulsion, and flexible operations, respectively.

Active control of combustion processes has been used in reducing pressure oscillations and pollutant formation, increasing combustion intensity and heat transfer rates, and operating combustors beyond their natural flammability limits (see [6] and references therein). Active control is promising due to a number of reasons: (1) it can be used to overcome some of the tradeoffs inherent in current combustion technology, such as reduced pressure oscillations accompanied by increased pollutant formation; (2) passive control has been shown to be inadequate as the operating conditions change; (3) actuators such as acoustic drivers and dynamic fuel injectors, which provide means of modulating key variables in the combustion process, as well as fast and accurate sensors such as pressure transducers, radiometers, and photodiodes are becoming available; and (4) active controllers consume a small fraction of the power generated in the system and hence are viable for commercial use.

The second class of flows that exhibit unstable behavior pertains to impinging jets. Due to the interactions of the acoustics with shear flow instability, resonances occur and manifest once again as large and sustained pressure oscillations. This instability phenomenon has a long history as well, and is viewed as an example of an edge tone that occurs any time a flow encounters a flat edge, thereby causing acoustic reflections which lead to a feedback loop and consequently, resonant behavior [2,7].

This class of impinging jet flows is similar to the first in that both flows produce resonances in terms of amplified pressure oscillations, and in that, in both cases, resonances are produced due to feedback interactions. There is, however, a distinct difference between the two, which is in terms of their control. While in the first case, active closed-loop control effectively removes the resonance, in the second case, the same is not possible due to the very large magnitude of the resonance frequencies.

Despite these distinctions, similar to the combustion processes, active control has been shown to be highly effective in stabilizing the impinging jets as well. Faced by difficulties in modeling using physics-based principles, system-identification based models have been derived and shown to lead to effective control designs. Due to bandwidth constraints, open-loop rather than closed-loop controllers have been proposed, with the controllers introducing frequencies that are much lower than the dominant natural frequencies of the system. The results show a dramatic reduction in the acoustic resonances using a very small fraction of the system energy.

In what follows, both classes of flows, their unstable oscillations, and their active control will be discussed. Section 33.2 addresses combustion dynamics and their control, while Section 33.3 addresses impinging jets and their control.

33.2 Combustion Oscillations

The unstable oscillations that will be discussed in this article are in pressure, and as such, are acoustic in nature. An oscillation of pressure transmitted through a medium is essentially a travelling acoustic wave from one point to another, composed of frequencies within the range of hearing (∼20–20,000 Hz, for human), and hence produces sound. A related unit of pressure measurement is decibels, and is defined with respect to a reference value as

$$SPL(\text{dB}) = 20\log_{10}\left(p_{rms}/p_{ref}\right)$$

where SPL denotes sound pressure level, p_{rms} is the root-mean-square of the measured pressure p, and p_{ref} is a reference pressure. A common choice for $p_{ref} = 20\,\mu\text{Pa}$, where μPa denotes micro-Pascals, when measured in air, and is considered to be the threshold of human hearing. Throughout this chapter, we refer to a system as being *unstable* if the sound produced can be heard above the ambient noise, and a system as being *stable* if the sound level is at or below the ambient noise level.

33.2.1 Feedback Mechanism: A Pendulum Analogy

We begin with a model of pressure wave oscillations in a duct, which can be represented as a simple harmonic motion. Denoting the pressure component that deviates from a nominal value as p, the pressure oscillations can be described as

$$\ddot{p} + \omega^2 p = 0 \tag{33.1}$$

where ω is the oscillation frequency and is normally determined by the geometry of the duct and its boundary conditions. Equation 33.1 is often referred to as the wave equation. In a combustion system, there are additional inputs into this wave equation. The simplest example of such an input is an unsteady heat-release rate, q. The presence of such a q modifies (Equation 33.1) as [8]

$$\ddot{p} + \omega^2 p = \dot{q} \tag{33.2}$$

The additional twist in this problem is that this heat release rate q is not an independent quantity but rather depends on p itself. In some cases, the flame in the combustor is such that the corresponding q is of the form

$$q = kp \tag{33.3}$$

where k is a constant that depends on the nature of the mechanism that anchors the flame in the combustion chamber. Equations 33.2 and 33.3 result in a feedback system of the form shown in Figure 33.1. It follows that the closed-loop transfer function is given by

$$G_{cl}(s) = \frac{1}{s^2 - ks + \omega^2}$$

It is therefore clear that the resulting system has growing acoustic oscillations if $k > 0$. That is, if the heat-release rate is in phase with the unsteady pressure, then the system exhibits instability; conversely, if the heat-release rate is out of phase with the unsteady pressure, then the system is stable with the acoustic oscillations decaying down to zero. This observation was made as early as 1785 by several researchers including Lord Rayleigh, who says in his treatise on the Theory of Sound [2], that "If heat be given to the air at the moment of greatest condensation, or be taken from it at the moment of greatest rarefaction, the vibration is encouraged. On the other hand, if heat be given at the moment of greatest rarefaction, or abstracted at the moment of greatest condensation, the vibration is discouraged." The actual relationship between pressure and heat release is however not a simple gain as in Equation 33.3 but significantly more complex, and is spatio-temporal in nature. The underlying mechanisms are, in addition, numerous, difficult to determine, and stem from the dynamics of a reactive flow. The complexity is a function of the combustor geometry, the inlet conditions, the flow velocity, the fuel mixture, and the mechanism used to anchor the flame in the combustor. Systematic progress however is being made in this area and surprisingly simple models are shown to be sufficient for the purpose of designing active control strategies in some cases. These are discussed in further detail in the following section.

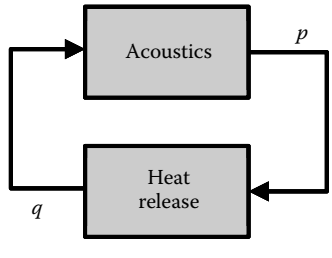

FIGURE 33.1 Schematic representation of the combustion dynamics.

33.2.2 A Dynamic Model of the Combustion Oscillations

One can derive a more detailed combustion model of a one-dimensional duct of length L which is assumed to have a concentrated heat release at a single location x_f produced by a flame which corresponds to a well-defined surface with unburnt reactants on one side and fully burnt product on the other side. This is of the form [9,10]

$$\ddot{\eta}_i + 2\zeta\omega_i\dot{\eta}_i + \omega_i^2\eta_i = b_i\dot{q} \tag{33.4}$$

$$\dot{q} = d_0 u + d_1(u_{\tau_f}(t)) + d_2(\phi_{\tau_f}(t)) + d_3\phi(t) + d_4\dot{\phi}(t) \tag{33.5}$$

$$\phi = \sum_{i=1}^{n} g_i\eta_i(t-\tau_c) \tag{33.6}$$

$$u = \sum_{i=1}^{n} c_i\dot{\eta}_i + \theta a_0 q \tag{33.7}$$

$$p = \overline{p}\sum_{i=1}^{n} c_{c_i}\eta_i, \tag{33.8}$$

where p is the main output variable of interest and denotes the perturbed pressure at a location x_s which lies between 0 and L, q is the heat release rate per unit area, η_i and $\dot{\eta}_i$, $i = 1, \ldots, n$ are the state variables and correspond to the n dominant modal amplitudes, u and ϕ correspond to the perturbed velocity and equivalence ratio at the combustion zone and denote two dominant coupling variables between acoustics and the heat release rate, and

$$x_\tau(t) \triangleq \int_{t-\tau}^{t} x(\zeta)\, d\zeta.$$

The parameters in Equations 33.4 through 33.8 are constants and denote the following: ζ represents the passive damping ratio in the combustor, $\omega_i = k_i\bar{c}$, $b_i = \gamma a_o\psi_i(x_f)/E$, $c_i = \frac{d\psi}{dx}(x_f)\frac{1}{\gamma k_i^2}$, $g_i = \frac{d\psi}{dx}(x_s)\frac{\overline{\phi}}{\rho\overline{u}}$, $E = \int_0^L \psi_i^2(x)dx$, $\psi_i(x) = \sin(k_i x + \phi_{i0})$, $i = 1, \ldots, n$. $a_0 = \gamma - 1/(\gamma\overline{p})$, θ represents the combined effects of the flow velocity both behind and ahead of the flame, L denotes the length of the combustor duct, k_i and ϕ_{i0} are determined from the boundary conditions, x_f and x_s denote the location of the flame and sensor, respectively, d_i, $i = 1, \ldots, 4$, depend on various flame parameters, physical constants such as density, burning velocity, and nominal flow velocity, γ denotes the specific heat, τ_f denotes the flame propagation delay, $\tau_c = L/\overline{u}$, $\overline{u}$ denotes the nominal velocity, $c_{c_i} = \psi_i(x_s)$, and $\overline{p}$ denotes the nominal pressure.

Equations 33.4 through 33.7 then determine the underlying uncontrolled combustion system, with state variables η_i, $\dot{\eta}_i$, $i = 1, \ldots, n$, and an output p, which has been shown to exhibit instability for certain parameter values and certain time delays τ_f and τ_c at all locations $x \in (0, L)$.

33.2.2.1 Models of Actuated Combustors

Active control of combustion instability is typically achieved through flow-modulating devices such as fuel injectors and loudspeakers, where the former has the dominant effect of additional mass flow, which results in additional heat release, and the latter introduces additional velocity, which impacts on both acoustics and heat release. Below we present both the models of actuated combustors and the models of the actuators.

The impact of an acoustic actuator whose diaphragm velocity, v_c, denotes the control input, is given by [11]:

$$\ddot{\eta}_i + 2\zeta_0\omega\eta_i + \omega_i^2\eta_i = b_i\,\dot{q} + b_{c_i}\dot{v}_c \tag{33.9}$$

$$y = \sum_{i=1}^{n} c_{c_i} \eta_i, \tag{33.10}$$

$$\dot{q} = \sum_{i=1}^{n} \left(g_f c_i \, \dot{\eta}_i + k_{ao} \alpha_r \nu_c \right), \tag{33.11}$$

where $y = p/\overline{p}$, the normalized unsteady pressure component, x_a is the location of the actuator, $k_{ao} = 0$ if $x_a > x_f$ and unity otherwise; $b_{c_i} = \frac{\gamma \alpha_r}{E} \psi_i(x_a)$; g_f is determined by the heat release rate per unit mass of the mixture and the flame stabilization mechanism. In Equation 33.11, no equivalence ratio perturbations are assumed to be present, and τ_f is negligible.

If the quantity added is fuel, in addition to the mass flow, heat input is introduced as well, since it changes the equivalence ratio. Defining the corresponding input as

$$\phi_c = \frac{\dot{m}'_c}{\overline{\dot{m}}_a \phi_0},$$

where $\overline{\dot{m}}_a$ is the mean air mass flow rate and ϕ_0 is the fuel-to-air ratio at stoichiometry, it can be shown that when only ϕ'-perturbations are present, the heat release dynamics in Equation 33.5 is altered as

$$\dot{q} = d_2(\phi_{\tau_f}(t) + \phi_{c_{\tau_f}}(t - \tau_{co})) + d_3(\phi + \phi_c(t - \tau_{co})) + d_4(\dot{\phi} + \dot{\phi}_c(t - \tau_{co})),$$

where $\tau_{co} = L_c/\overline{u}$ and L_c is the distance between the burning plane and the location of the fuel injector. If the heat release dynamics are only due to ϕ'-perturbations, for a single-mode model, the overall combustion model is of the form

$$\ddot{\eta} + 2\zeta\omega\dot{\eta} + \omega^2\eta - \beta\eta(t - \tau_c) = k_c \dot{\phi}_c(t - \tau_{co}). \tag{33.12}$$

Equations 33.9 through 33.12 represent two different examples of combustion systems together with a control input in their simplest form.

An alternative approach to modeling the pressure perturbations is to characterize one-dimensional pressure perturbations as waves that are reflected by boundary conditions at the upstream and downstream ends with a certain reflection coefficient. This leads to a different description than those mentioned above, whose structure consists of a number of propagation delays of the reflected and outgoing waves. The readers are referred to [12,13] for further details. The same has been used for several successful active control implementations as well [7].

33.2.2.2 Nonlinear Mechanisms

Nonlinear features are abundant in a combustion process. The most dominant of these is a limit-cycle behavior that is exhibited by almost all the variables in the process, including pressure, velocity, and heat release. The typical dynamic response of any of these variables consists of a divergent set of oscillations that transition to a sustained periodic signal, which is almost sinusoidal in nature. Several speculations have been made regarding mechanisms responsible for such a behavior. Nonlinearities in the heat-release dynamics have been noted in [3,12,14–16], whereas nonlinearities in acoustics are claimed to be responsible for these limit cycles in [17–19]. For a more detailed discussion of nonlinear models, see [20].

The presence of limit cycles suggests the obvious presence of bifurcations. A key parameter that appears to induce these bifurcations is the mean equivalence ratio, ϕ. Two distinct ranges of ϕ appear to be of interest, depending on the application. In ramjet engines and afterburners, instability appears to result close to stoichiometry, which is then followed by a Hopf bifurcation. In engines with strict emission requirements, as one attempts to burn lean, a "blow-out" limit is reached that once again is accompanied

by these bifurcations. In many of these cases, more than one limit cycle is encountered [16], suggesting the presence of both sub- and supercritical bifurcations.

Finally, in [16,21–23], hysteresis mechanisms have been observed and discussed. The parameters in question are the mean equivalence ratio and the mean inlet velocity. In [21], keeping other parameters constant as ϕ is increased steadily and then decreased, the behavior at the same value changes from instability to stability, and a drastic change in the flame structure is observed at some of these instances. In [21] and [23], it is shown that once such a mechanism is present, appropriate use of it can be made in designing active control strategies to reduce the amplitude of oscillations.

33.2.3 Control of Combustion Oscillations

33.2.3.1 Control Methods

We now address the control of the first class of flows that exhibit resonances, whose dynamics was discussed in the preceding section. Early approaches used in this field for control was an empirical one and deployed the use of a phase-shift. The basic idea of this controller is to measure the pressure, add an appropriate phase, and generate a signal that is equal and opposite to the actual pressure, and therefore attempt to cancel out the oscillations. Since the undesirable pressure oscillations often occur at a few, if not a single, frequency, the phase shift controller includes a filter in addition to the actual phase-shift action and some amplification. The phase-shift action is implemented in many cases as a pure time delay whose value is adjusted manually, by trial and error, until the oscillations are reduced. While this strategy is quite successful in some cases, often secondary peaks are generated due to the control action, thereby compromising the maximum damping that is achievable. If more than one frequency is present, the control design seems to prove quite challenging. In what follows, we describe model-based control approaches that have been successfully implemented in a range of rigs.

The first point to note when it comes to active control is that since the system is required to add damping, P or PI control is not adequate. Some phase-compensation is needed in order to obtain stability. Also, the system is high-dimensional, and very few system measurements are possible. It should also be kept in mind that the available control authority is limited. All of this makes an optimal control based on a linear quadratic Gaussian (LQG) very suitable for its control. Given the myriad sources of delay in the system, a control approach based on time delay is attractive as well. Finally, the varied sources of uncertainties behove an adaptive solution as well. These control methods are discussed below.

33.2.3.1.1 Linear Optimal Control

Since the goal is to reduce the pressure oscillations as quickly as possible for a given actuator with a certain control authority, a linear control strategy that seeks to minimize a cost function that is of the form

$$J = \int_0^\infty \left(p^2 + \rho u_c^2\right) dt \tag{33.13}$$

where u_c is the control input from either a speaker or a fuel-injector and ρ is chosen so as to represent the available control effort, is found to be quite suitable for this problem. The model in Equations 33.9 through 33.11 can then be used to determine the control input u_c as a function of the pressure measurements and the model parameters. To minimize the effect of modeling uncertainty, an LQG- loop-transfer recovery (LTR) control procedure [24] can be used so that the estimator minimizes the effect of the modeling error by representing the latter as a fictitious Gaussian noise. This controller has been implemented in several rigs, and some of the results are briefly described in the section on active combustion control practice.

An alternative control procedure is based on the $\mathscr{H}_\infty$ approach, which ensures that desired measures of stability robustness and performance specified in the frequency domain are achieved. These specifications are given as desired shapes for closed-loop transfer functions between selected groups of exogenous inputs and controlled outputs. For example, robust stability usually requires the appropriate closed-loop transfer function to be small at high frequencies, as the size of uncertainty is large there. Since $\mathscr{H}_\infty$-optimal control

leads to all-pass closed-loop transfer functions, the $\mathscr{H}_\infty$ control problem is formulated using frequency-weighted transfer functions. This method has also been applied for combustion instability using models that are based on a wave-approach in [25] and models of the form Equations 33.9 through 33.11 in [26] and led to satisfactory pressure reduction.

Note that both of the above methods either neglect the effects of time delay or use Pade approximants to represent time-delay effects using a finite-dimensional model, which restricts the domain of applicability of these controllers to systems where the delay is small.

33.2.3.1.2 Time-Delay Control

When time delays are large, it is efficacious to use control methods that explicitly include time delays in their design. Simple phase-lead control strategies that make optimal use of actuator locations can be used [27] to either cancel out or minimize the delay effects in some cases. A more general strategy is the Posicast controller based on the Smith predictor [28–30]. The idea behind this control strategy is to forecast the future output using the system model and use this in turn to stabilize the system. The controller structure is given by

$$\begin{aligned}
\dot{\omega}_1 &= \Lambda_0\omega_1 + \ell u_c(t-\tau)\\
\dot{\omega}_2 &= \Lambda_0\omega_2 + \ell p(t)\\
u_c &= \theta_1^T\omega_1 + \theta_2^T\omega_2 + u_1(t)\\
u_1(t) &= \int_{-\tau}^{0}\sum_{i=1}^{n}\alpha_i e^{-\beta_i\sigma}u(t+\sigma)\,d\sigma,
\end{aligned} \tag{33.14}$$

where u_c is the control input, p is the pressure measurement, ω_1 and ω_2 are the state estimates of the complete combustion system, n is the order of the system, u_1 corresponds to the output prediction, Λ_0 is an $n \times n$ stable matrix, (Λ_0, ℓ) is controllable, and θ_1, θ_2, α_i, and β_i are the controller parameters. The reader is referred to [27,31,32], for further details regarding the stability and robustness properties and experimental and numerical results of the closed-loop performance.

33.2.3.1.3 Adaptive Time-Delay Control

As shown in the section on time-delay control, the presence of delay can be accommodated by adding a signal to the control input that attempts to anticipate the effects of the delay. The same approach can be adopted in an adaptive controller as well. The structure of the controller is of the same form as in Equation 33.14, but the parameters θ_1 and θ_2 are adjusted online and u_1 is chosen as

$$\begin{aligned}
u_1 &= \overline{\lambda}^T(t)\overline{u}(t)\\
\dot{\theta}(t) &= -y(t)\omega(t-\tau),
\end{aligned}$$

where $\theta = [\theta_1^T, \theta_2^T, \overline{\lambda}^T]^T$; $\omega = [\omega_1^T, \omega_2^T, \overline{u}^T]^T$; $\overline{u}_i$, the ith element of the vector $\overline{u}(t)$, is the ith sample of $u(t)$ in the interval $[t-\tau, t)$, $i = 1, \ldots, p$; and p is chosen to be small enough so that the sampling error in the realization of u_1 is small. The stability of the above controller is discussed in [32]. A controller whose order depended on the relative degree of the plant rather than its own order was developed in [13] and successfully implemented on a benchtop combustor in [31].

33.2.3.2 Control Demonstrations

The first successful demonstration of active combustion control occurred in 1983 when, using a Rijke tube (an organ pipe driven into resonance using a heat source) and a loudspeaker and a microphone as an actuator-sensor pair, Dines [33] demonstrated that a 40 dB reduction can be achieved in the heat-induced noise. Since then, this technology has grown considerably and has been studied in the context of a number

of laboratory-scale (1–100 kW), medium-scale (100–500 kW), and large-scale rigs (1 MW and above). In this section, examples of these studies are presented. The examples are chosen to illustrate the wide variety of combustors studied, such as rigs with varied configurations, different kinds of feed delivery, various boundary conditions, and operating conditions that range from lean burning to burning near stoichiometry. Although most of these examples have used model-based control strategies that were presented earlier, a few experimental results that used empirical strategies are also included for comparison.

33.2.3.2.1 Laminar Tube Combustors

Pressure oscillations in several tube-combustors have been controlled successfully since 1984 using empirical control strategies, whereas in [26,31,34], model-based controllers were used. A system-identification-based model was used in [34], while physically based models as in Equations 33.9 through 33.11 and those based on a wave approach were used in [26] and [31], respectively. A comparative study between empirical and model-based controllers was carried out in [26] and is briefly described below. The combustion chamber is a 5.3 cm diameter, 47 cm long tube closed at the upstream end, and open at the downstream end. The flame was anchored on a perforated disc with 80 holes fixed 26 cm from upstream end, with several ports included for mounting actuators and sensors. A condensor microphone was used as the sensor, and a 0.2 W loudspeaker was used as an actuator, whose parameters were determined using system-identification methods. The inlet temperatures were at ambient values. Measurements on the test rig were recorded using a Keithley MetraByte DAS-1801AO data acquisition and control board with a maximum sampling frequency of 300 KHz. Most experiments were conducted with an equivalence ratio between 0.69 and 0.74 and an airflow rate of 333 mL/s (0.38 g/s), which corresponded to an unstable operating condition without control (equivalence ratios of less than 0.69 corresponded to a stable operating point). The flow rate was varied between 267 and 400 mL/s, and the power of the combustor was 0.831 kW. The unstable frequency of the combustion process was found to be 470 Hz. Using the models in Equations 33.9 through 33.11, LQG-LTR and $\mathscr{H}_\infty$ strategies were designed and implemented on a Pentium PC with a sampling frequency of 10 kHZ. A 50 dB pressure reduction with a fast settling time was achieved with the former using a peak electrical power of 3 mW (see Figure 33.2). In contrast, it was observed that the phase-shift controller resulted in a 20 to 30 dB pressure reduction and often resulted in secondary peaks.

In [31], model-based adaptive controllers listed in the section "Adaptive Control" were implemented on a tube combustor, where it was observed that the adaptive time-delay controller outperformed the others when the time delay was about four times the acoustic time constant and a 40% uncertainty was introduced in the unstable frequency by increasing the tube length.

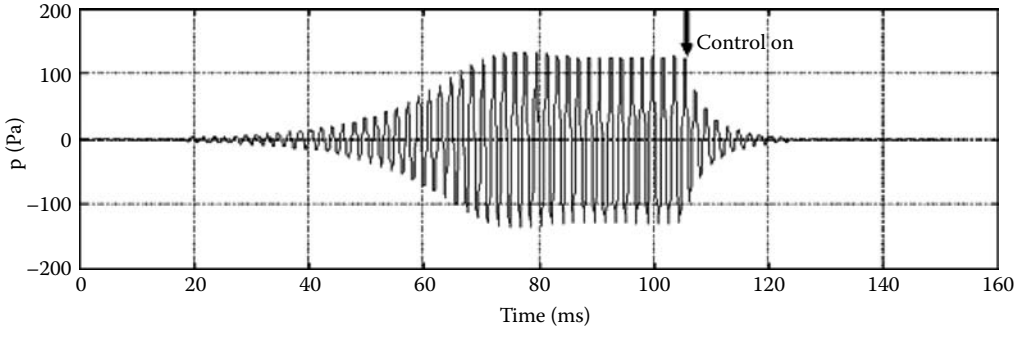

FIGURE 33.2 Pressure response using the laminar 1 kW combustor. The figure shows the linear instability over the first 80 ms, followed by the nonlinear limit-cycle behavior. The controller is turned on at 300 ms, which results in a settling time of about 40 ms. (Adapted from A. Annaswamy et al., *IEEE Transactions on Control Systems Technology*, vol. 8, no. 6, pp. 905–918, November 2000.)

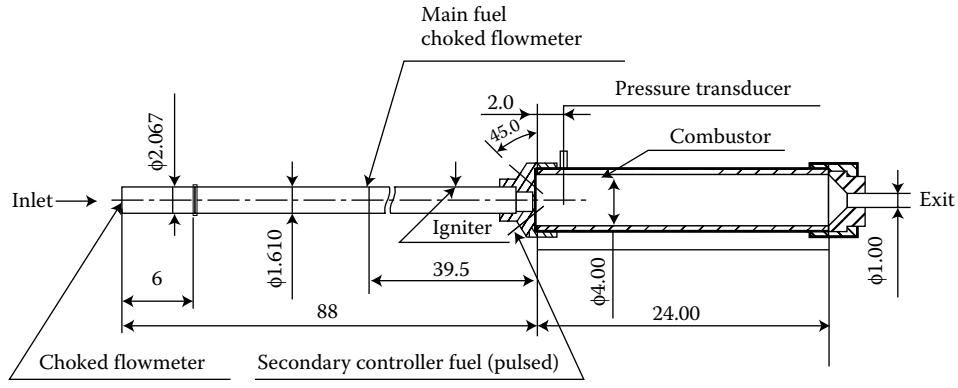

FIGURE 33.3 Schematic representation of a dump combustor.

33.2.3.2.2 Dump Combustor

A dump combustor that mimics realistic ramjet operating conditions was constructed in [35] to evaluate the effect of model-based control on combustion dynamics at a high Mach number. The combustor consisted of a 1.6 in. circular pipe expanding into a square dump whose height and length are 4 and 24 in., respectively (see Figure 33.3 for a schematic representation of the dump combustor). Ethylene was premixed with the air 1 m upstream of the dump plane. Actuation was accomplished by four secondary fuel injectors located at the dump plane, oriented 45° from the combustor axis pointing downstream. Liquid ethanol is used in the secondary fuel stream at flow rate of 0.6 g/s which is less that 10% of the main fuel. The Mach number of the inlet flow was set at 0.3 so as to mimic realistic ramjet operating conditions. A pressure sensor was located 2 in. downstream from the dump plane actuator.

The combustor showed strong instability for an equivalence ratio of $\phi > 0.72$. The level of instability is 3 psi in rms pressure with more than 170 dB at 250 Hz which corresponds to the quarter wave mode of the combustor, as shown in Figure 33.4. For $\phi < 0.72$, the combustor was stable, and the transition from stable to unstable combustion was observed to be sudden.

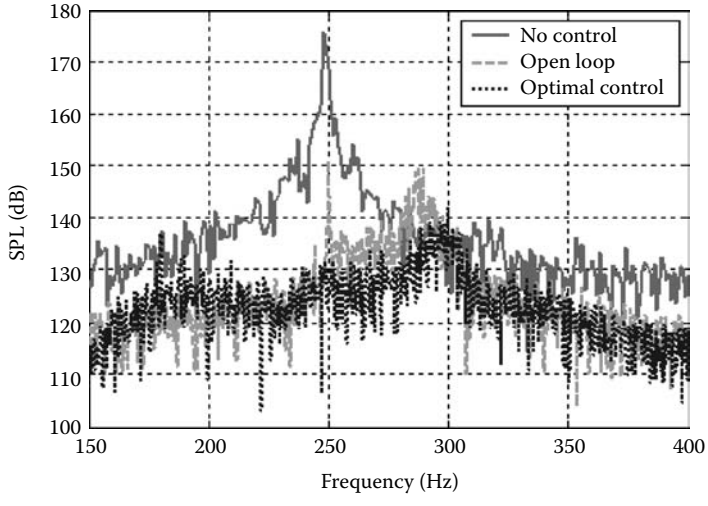

FIGURE 33.4 Pressure spectrum without control, with forcing at 250 Hz, phase-shift control and optimal control at $\phi = 0.8$ and $Ma = 0.3$.

A system-identification based model was developed using injector and pressure signals as an input and output, respectively. A sine sweep signal from 200 to 360 Hz was supplied to the injector and corresponding pressure signal was recorded using a data acquisition board.

It was observed that an LQG-LTR controller was most suitable for reducing the oscillations. An operating condition corresponding to an equivalence ratio of 0.8 and $Ma = 0.3$ was chosen to implement the active controller. The LQG-LTR controller resulted in pressure reduction of 30 dB shown in Figure 33.4. To test whether the control can be used in a wide range, the overall equivalence ratio was varied from 0.8 to 0.92. It was observed that the optimal controller could suppress pressure oscillations up to at least $\phi = 0.9$. A noteworthy fact is that without active control, the maximum achievable equivalence ratio was $\phi = 0.72$.

33.2.3.2.3 Large-Scale Rigs

The best example of demonstrating the active control technology in a large-scale industrial rig is that of a Siemens AG 94.3A 260 MW ring combustor. This study indicates the feasibility, the gains that can be obtained using active control, the hardware and software needed to implement the technology, and the state of maturation of the technology in the context of combustion control.

A 260 MW heavy-duty gas turbine developed by Siemens AG Power Generation was shown to use active combustion control successfully in suppressing pressure oscillations. Judiciously combined with passive measures, pressure reductions on the order of 15–20 dB were obtained at a range of loading conditions. The first set of results related to this rig was reported in 1997 [36], followed by [37], which documents further improvements to the control design (see Figures 33.5 and 33.6 for a schematic representation and the response due to active control, respectively).

Currently, this design has been installed in 13 turbines. The field-leading installation was implemented in January 1999, has been operating for more than 6000 h, and continues to demonstrate the long-term reliability of the active control system.

The active control system in the Siemens combustor consists of modulation of the pilot gas supply using a Moog direct-drive valve, which has a bandwidth of about 420 Hz and can withstand ambient temperatures of about 120°C. A piezoelectric pressure transducer was used as a sensor. Similar burners,

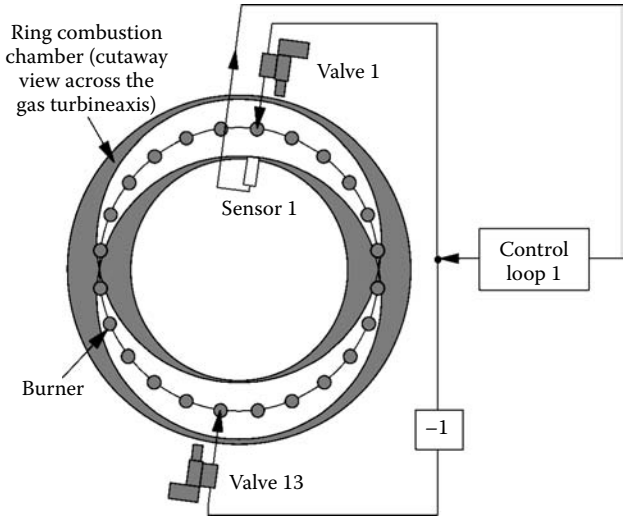

FIGURE 33.5 A schematic representation of the 260 MW heavy-duty gas turbine. The original version of this figure has been published by the Research and Technology Organization, North Atlantic Treaty Organization (RTO/NATO) in MP-51, "Active Control Technology for Enhanced Performance Operation Capabilities of Military Aircraft, Land Vehicles, and Sea Vehicles" in March 2001. (Adapted from J. Hermann et al., *NATO RTO/AVT Symposium on Active Control Technology for Enhanced Performance in Land, Air, and Sea Vehicles,* Braunschweig, Germany, May 2000.)

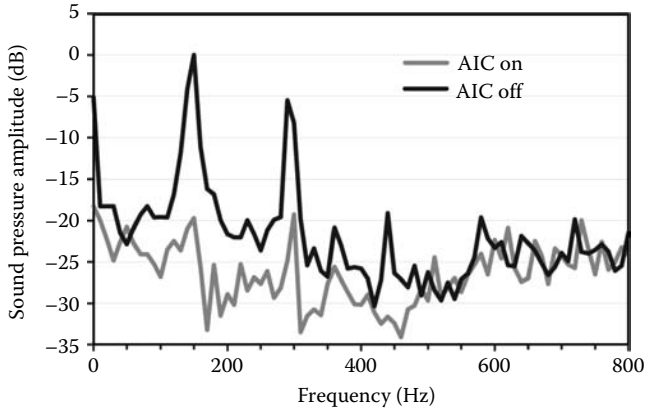

FIGURE 33.6 The impact of active control in the 260 MW heavy-duty gas turbine. The original version of this figure has been published by the Research and Technology Organization, North Atlantic Treaty Organization (RTO/NATO) in MP-51, "Active Control Technology for Enhanced Performance Operation Capabilities of Military Aircraft, Land Vehicles, and Sea Vehicles" in March 2001. (Adapted from J. Hermann et al., *NATO RTO/AVT Symposium on Active Control Technology for Enhanced Performance in Land, Air, and Sea Vehicles*, Braunschweig, Germany, May 2000.)

each equipped with a pilot system, were actively controlled using different control loops. Each control loop modulated two valves, which in turn controlled two burners located diametrically opposite of each other. The feedline system was designed so as to amplify the control authority at the unstable frequencies where active control was most needed. The turbines actively controlled ranges from 233 to 267 MW. Additional passive measures were combined with the active control strategy to increase the pressure reduction.

Although the details of the control strategy are proprietary, the descriptions of the active control strategy reported show that it was based on a phase-shift idea; the control signal was generated using the pressure measurement, to which a time delay was added and appropriately tuned. Initial versions of the controller reported in [36] were capable of adding this delay only at a single frequency; currently, however, their algorithm allows the simultaneous phase addition at any two frequencies of oscillation [37].

The above study illustrates that under realistic conditions of temperature and pressure in a large-scale rig and a high-power output, active control can be implemented successfully. It shows that significant improvement can be obtained using this technology, leading to considerable savings. The fact that the rig has been operational for about 6000 h attests to the reliability and longevity of the proposed technology.

33.3 Impinging Jets

We now consider a second class of flows that exhibits resonances, which is an impinging jet. Similar to the previous class of combustion systems, here too, the underlying phenomenon is one of growing amplitudes of the unsteady pressure, and is caused through feedback interactions. In what follows, we present the details of these interactions and a dynamic model that encapsulates them. Unlike combustion systems, finding the requisite actuator that is capable of affecting the underlying flow processes in a dynamic manner is shown to be a significantly more challenging task. In addition, unlike the combustion systems, the control of resonances in impinging jets is enabled not through closed-loop control methods but through open-loop. In the subsequent sections, the actuator, the experimental results that illustrate the control of impinging jets, as well as a dynamic model that explains the underlying principle behind how the control is achieved using the actuator are described.

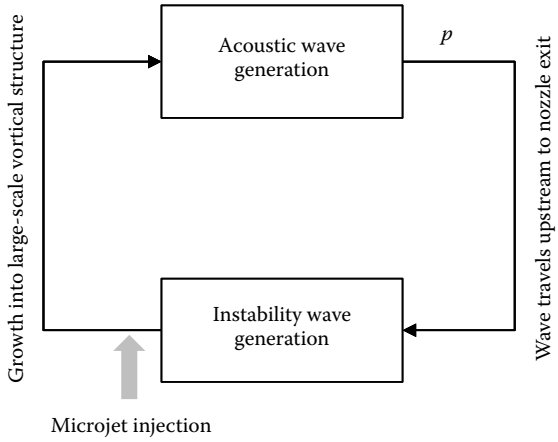

FIGURE 33.7 A block diagram of the physics-based feedback model of the supersonic impinging jet. p denotes the pressure perturbation close to the ground.

33.3.1 Feedback Mechanism

In addition to resonances in combustors which have thermally excited flow, acoustic resonances have their origin in the instability of several other fluid motions. One of these motions is in the context of impinging high-speed jets which correspond to high-speed jets that impinge on a hard surface. Experienced by Short Take Off and Vertical Landing (STOVL) aircraft while hovering in close proximity to the ground, impinging tones, which are discrete, high-amplitude acoustic tones, are produced due to interactions between high speed jets emanating from the STOVL aircraft nozzle and the ground [38]. These feedback interactions occur thus: Instability waves are generated by the acoustic excitation of the shear layer near the nozzle exit, which then convect down and evolve into spatially coherent structures. Upon impinging on the ground, the instability waves excite the waves of neutral acoustic modes of the jet, which in turn excite the shear layer at the nozzle exit, thereby closing the feedback loop. The high-amplitude impinging tones are undesirable not only due to the associated high ambient noise, but also because of the accompanied highly unsteady pressure loads on the ground plane and on nearby surfaces. While the high noise levels can lead to structural fatigue of the aircraft surfaces in the vicinity of the nozzles, the high dynamic loads on the impingement surface can lead to an increased erosion of the landing surface as well as a dramatic lift-loss during hover.

The underlying mechanisms and their feedback interactions are shown in Figure 33.7, which consist of a feedforward mechanism pertaining to noise generation due to the impingement of the large-scale vortical structure on the ground that is reflected back to the nozzle, and a feedback mechanism that corresponds to the generation of shear layer instability produced at the nozzle exit due to the acoustic excitation. The feedforward process can be modeled as the impingement of the large scale vortical structure on the wall by viewing it as a head-on collision of two identical vortices [39]. Unlike the feedforward process, the generation of shear layer instability at the nozzle exit and its response to acoustic excitation is significantly more difficult to model. Therefore instead of using the underlying physics, a system identification based model of the above feedback loop was derived.

33.3.2 A Dynamic Model

In this section, a low-order lumped parameter model structure is used to represent the dynamics of acoustic resonances in an impinging jet. This model structure has two blocks that represent the two dominant components of shear-layer dynamics and acoustics, which interact, as mentioned before, through

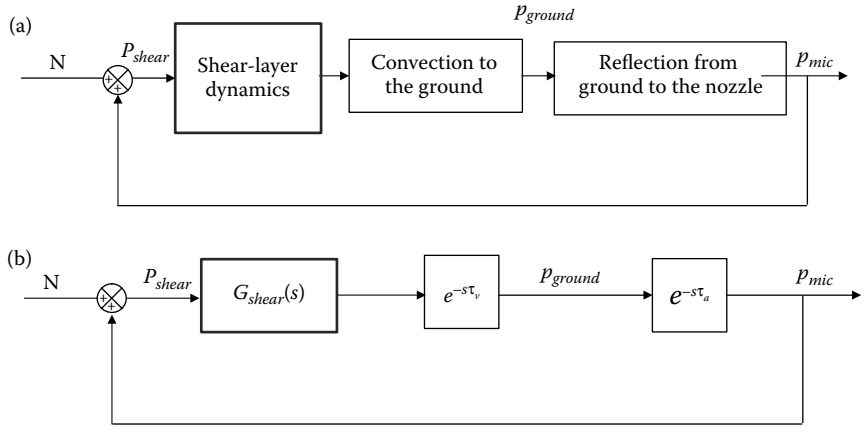

FIGURE 33.8 A block diagram of a system-identification based feedback model of the supersonic impinging jet in the uncontrolled case. Feedback loop with (a) the dominant mechanisms, and (b) the corresponding transfer functions.

feedback (Figure 33.8). In this figure, P_{shear} denotes the perturbations in the pressure near the nozzle exit and acts as an input to the shear-layer dynamics. The shear layer in turn responds to these acoustic perturbations whose response in turn is convected down, leading to P_{ground}, which denotes the pressure perturbations on the ground plane. The resulting acoustics is reflected by the ground, travels upwards toward the nozzle, and is measured as P_{mic}. Since the objective is to capture the most dominant dynamics exhibited by the impinging jets, the smallest set of mechanisms that correspond to both the acoustics and the shear-layer dynamics is retained in this model. Since the dominant role of the acoustics appears to be one of convection, the transfer function between P_{ground} and P_{mic} is modeled as that due to a pure time delay, $e^{-\tau_a s}$ as, where $\tau_a = h/C_a$ is the time taken for the reflected acoustic wave to travel from the ground to the nozzle exit, h is the distance between ground and lift plate, and C_a is the velocity of the acoustic wave. The shear-layer dynamics is decomposed into two parts, where the first represents the response of the shear layer to the pressure perturbations at the nozzle, with a transfer function $G_{shear}(s)$, and the second represents the convection of the shear-layer wave from the nozzle to the ground, with a transfer function $G_2(s)$. Again, since the dominant role of the latter is convective lag, we model $G_2(s)$ as a time-delay $e^{-\tau_v s}$, where $\tau_v = h/C_v$ represents the convective time-delay between the nozzle and the ground, and C_v is convective velocity of the large-scale vortical structure. From the operating conditions, C_v is given by

$$C_v = \frac{1}{h}\int_0^h C(\xi)\, d\xi$$

where $C(\xi)$ denotes the convective velocity of the large scale vortical structure at a height ξ. Particle image velocimetry studies indicate that this is about 52% of the speed of the main jet [38]. Therefore, for a given height h, τ_a and τ_v can be easily calculated. The resulting closed-loop system with the corresponding transfer functions is illustrated in Figure 33.8b.

The remaining and most complex component that produces acoustic resonances in impinging jets is $G_{shear}(s)$ since it encapsulates the primary cause of the amplification. This amplification occurs at the nozzle, and is due to factors such as the shear-layer thickness, acoustic intensity of the propagating wave, and primary jet velocity, and occurs at certain preferred frequencies. Hence, we represent $G_{shear}(s)$ to be

$$G_{shear}(s) = \sum_{i=1}^{N_{mode}} \frac{K_i}{s^2 + 2\zeta_i\omega_i s + \omega_i^2} \tag{33.15}$$

where ζ_i is the damping ratio, ω_i is the natural frequency, and K_i is the amplitude of the ith mode, respectively. The number N_{mode} depends on the frequencies that are dominant, and respond to input excitation. As will be justified later, we essentially choose $N_{mode} = 2$, where the first frequency corresponds to the largest peak at a staging mode [38,40], and the second being a low frequency peak. While the former is clearly a dominant peak and therefore needs to be retained, the latter is not obvious in the uncontrolled response. However, it appears that this mode is excited in the presence of a control input, which is discussed in detail in the next section, and hence, we include it in our model. A noise input N is added at the input, which represents all other input excitations that exist at other frequencies including broadband noise effects. This results in an overall loop transfer function from N to P_{mic}, which corresponds to the pressure measurement at the sensor location, of the form

$$G_{closed}(s) = \frac{G_{shear}(s)e^{-sT}}{1 - G_{shear}(s)e^{-sT}} \tag{33.16}$$

where $T = \tau_a + \tau_v$ and represents the transfer function of the impinging jet resonance in the uncontrolled case. It should be noted that a feedback model similar to that suggested in Figure 33.8 was introduced in Rowley et al. [41] in the context of cavity tones.

The different parameters of the model described above are determined in the following manner. Based on Figure 33.9, ω_1 is set to the dominant frequency of 4.6 kHz. As will be shown in the next section, the flow responds at a very low frequency of 10 Hz in the presence of control, and hence ω_2 is set to be 10 Hz. The noise input N is determined from PIV measurements of the velocity distribution at the nozzle exit. We denote h as the height of the lift plate from the ground and d as the diameter of the jet nozzle at the exit. The remaining parameters K_i and ζ_i in Equation 33.15 are chosen by minimizing the peak response in pressure from the closed-loop model in equation and experimentally observed values at a given height h. At a height of $h/d = 3.5$, it was observed that $K_1 = 2.45\text{E} + 06$, $K_2 = 475$, $\zeta_1 = 0.001$, and $\zeta_2 = 0.1$. Because parameters K_i, ζ_i are determined from corresponding actual physical variables,

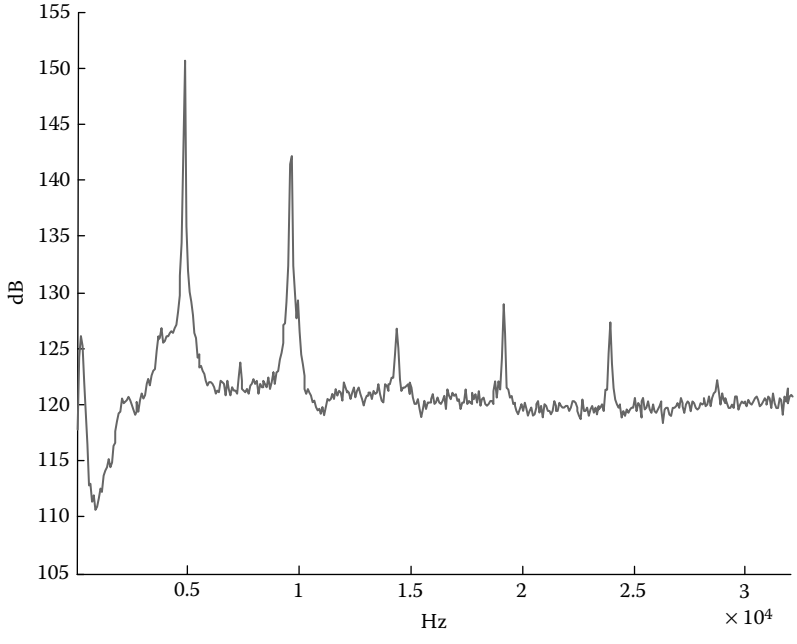

FIGURE 33.9 Spectral plot of the pressure response from the uncontrolled flow of the supersonic impinging jet, at operating conditions $h/d = 3.5$, NPR = 3.7.

phenomena such as shear-layer modification and spreading of the acoustic wave front, which is derived from shear-layer interaction mechanism, are interpreted by the lumped parameter transfer function.

33.3.3 Control of Impinging Tones

33.3.3.1 Actuator: A Supersonic Microjet

As mentioned in the introduction, impinging jets exhibit pronounced acoustic resonances due to the combined action of intrinsic instabilities in the flow and feedback action resulting from ground reflections. Using the STOVL facility, the acoustic resonances were recreated, an instance of which is illustrated in Figure 33.9, where the impinging tones are depicted. The response shown in Figure 33.9 corresponds to an operating condition where $h/d = 3.5$, where h is the height of the lift plate from the ground plate. The figure clearly shows the resonances, and a dominant natural frequency at 4.6 kHz and its harmonics. Similar resonances are exhibited in the impinging flow field for a range of heights. In order to represent the total noise produced by a scalar measure, a metric denoted as overall sound pressure level (OASPL) is calculated, which is defined as [42]

$$\text{OASPL}(p) = 20 \log_{10} \left(P_{rms}/P_{ref} \right) \tag{33.17}$$

where $P_{rms} = \lim_{T\to\infty} \sqrt{\frac{1}{T} \int_0^T p(t)^2 \, dt}$, and $P_{ref} = 20\,\mu Pa$. The OASPL that corresponds to $h/d = 3.5$ to 6 is shown in Figure 33.11. Figures 33.9 through 33.11 show that acoustic resonances in impinging jets are significant, appear to have nonlinear components, and persist at all operating conditions.

As mentioned earlier, the main cause of the acoustic resonances is a feedback loop, which gets closed by the acoustic wave travelling up to the nozzle. In order to interrupt the shear layer in a most efficient manner, a circular array of microjets was used as actuators. Due to their small size which is of the order of a few hundred microns in diameter, these microjets can be optimally distributed along the circumference and can also be introduced on-demand. In [43], it was shown that when microjets are introduced at a constant flow rate at the nozzle exit, the impinging tones were either significantly reduced or completely

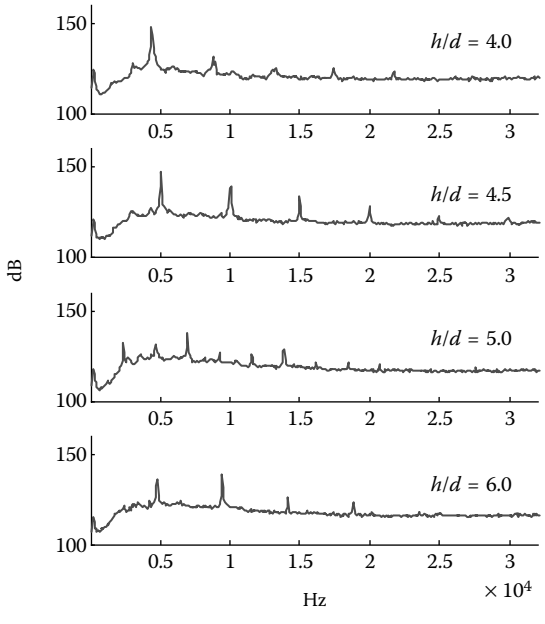

FIGURE 33.10 Spectral plot of the pressure response from the uncontrolled flow, for $h/d = 4.0$, 4.5, 5.0, and 6.0 NPR = 3.7.

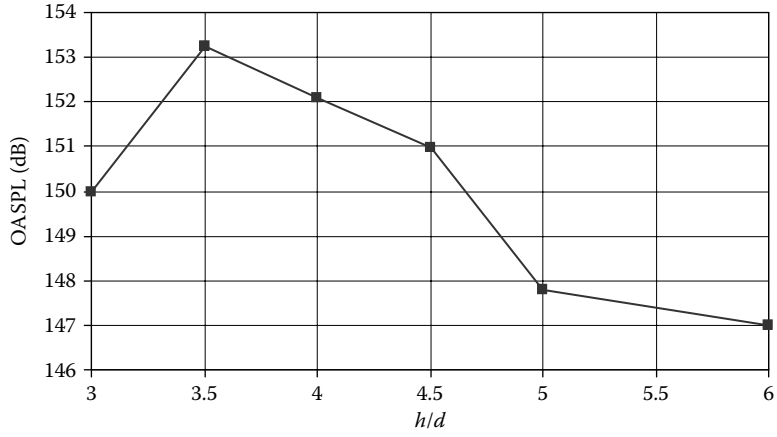

FIGURE 33.11 OASPL of the pressure response from the uncontrolled flow as a function of h/d, at NPR = 3.7.

eliminated. It was also observed that the amount of suppression is dependent on the operating conditions to a large extent.

33.3.3.2 Pulsed Microjets

In order to maintain a uniform reduction across all operating points in an efficient manner, steady microjet injection was replaced by microjets that were pulsed. The pulsing was accomplished by placing a rotating cap at the exit of the microjet (see Figure 33.12). Several alternate candidates were explored, and it was found that the maximum impact on the flow resulted using the rotating cap since it altered the flow at the nozzle exit [44,45]. This cap consists of several teeth which block and unblock the microjet holes as the cap rotates, simulating an on–off microjet action when spun with a motor. The bandwidth of the resulting actuation is limited to the speed of the motor.

The effect of the pulsed microjets was varied by changing the parameters of the pulsed flow such as amplitude, frequency, duty-cycle, and phase. The pulsing amplitude is directly proportional to the supply pressure delivered to the microjet chamber, while pulsing frequency is solely controlled by the rotation speed of the cap. Therefore, these two parameters can be easily and electronically varied by changing the microjet pressure and the motor speed. The duty cycle and the phase, on the other hand, depends on the design of the rotating cap and requires a mechanical design procedure. For example, if d_c is the duty cycle

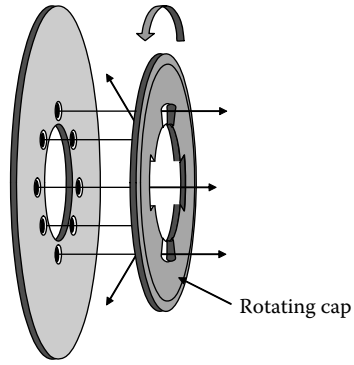

FIGURE 33.12 A schematic representation of the pulsing actuator. In the indicated position, the microjets are unblocked. With a clockwise turn, the cap geometry is such that the microjets are blocked, which simulates a pulsing action.

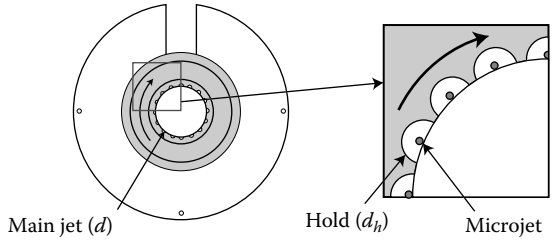

FIGURE 33.13 Pulsed microjet configuration, indicating the lift-plate and a magnified view, to illustrate the diameters of the microjets and teeth of the rotating cap. A total of 16 microjets were used in all experiments reported in this chapter.

of pulsing, which is the ratio of the valve opening time to pulsing period, then

$$d_c = 100\left(\frac{N_h d_h}{\pi d}\right)(\%) \tag{33.18}$$

where d is the main jet diameter, d_h is the diameter of hole in the rotating cap (see Figure 33.13), and N_h is the number of holes in the rotating cap. This implies that the duty cycle is changed by varying the number and diameter of holes of the rotating cap. If the number of holes in the rotating cap is the same as that of microjets, all microjets pulse synchronously.

33.3.3.3 Results Using Pulsed Microjets

Of all the parameters, it was found that the noise reduction was affected mainly by varying the frequency of pulsing. Spinning the rotating cap over a range of frequencies from 60 to 150 Hz was found to produce a noise reduction over the entire range. While the amount of noise reduction was independent of this frequency range, when we decreased the pulsing frequency below 60 Hz, it was observed that an additional noise reduction was possible at low frequencies around 20 Hz. In repeated trials, an additional 1–2 dB reduction was always achieved by low frequency pulsing injection. Figure 33.14 indicates, for instance, the reductions obtained in two independent trials, denoted as Tests 1 and 2. This is in significant contrast to high-frequency excitation-based actuation used in other flow control applications [46]. Such a dominant

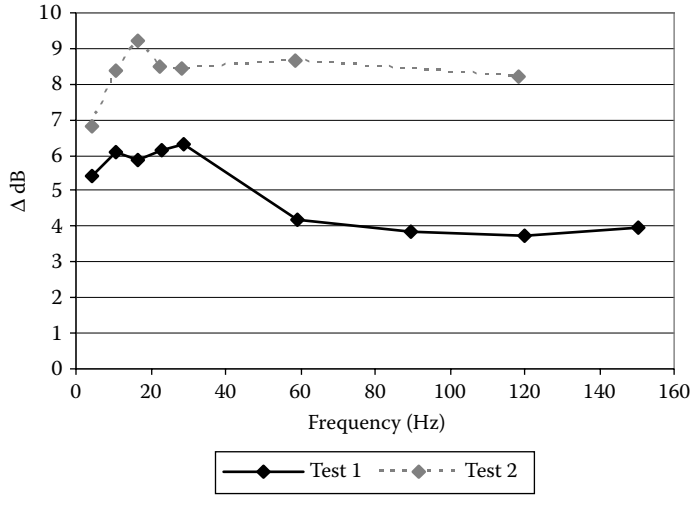

FIGURE 33.14 The OASPL reduction obtained using the pulsed microjet as a function of different pulsing frequencies in two independent tests, Test 1 and Test 2. The corresponding $d_c = 56\%$, and $h/d = 3.5$.

response suggests that the system response at this frequency is due to a linear effect rather than a nonlinear effect due to harmonics. Hence, we chose a model as described in Equations 33.15 and 33.16 to describe the response of the uncontrolled jet with ω_2 corresponding to this low frequency. The question that arises then is if the uncontrolled system exhibits a peak at a low frequency in the range of 10–50 Hz. This is examined below.

In almost all investigations of the impinging jets, the spectral plot of the uncontrolled impinging jet noise is distinguished by a dominant peak ~4.6 kHz which corresponds to the impinging tone (shown in Figure 33.15b). However, Figure 33.15a, which is the spectral plot of low-frequency band, shows a moderate peak also at a low-frequency of 1–10 Hz. The corresponding resolution of this plot was 2 Hz which ensured that no aliasing effects were present. Peaks in the low-frequency region were also observed at other locations such as near the lift and ground plate as well as in the far-field region. This proves that the low frequency peak is not due to experimental errors but a meaningful mode that should be considered for model construction, and hence justifies the model developed earlier.

33.3.3.3.1 A Systems Model for the Control of Impinging Jets

We note from Figures 33.14 and 33.16 that the pulsed microjet injection not only excites the low frequency but also leads to a reduction of the amplitude at the high frequency and its harmonics. Such a reduction is possible only through one of two effects, one of which is damping, while the other is a reduction of the input excitation at this frequency. The former case requires the introduction of a control input at the same frequency but with a different phase. This was clearly not the case in the pulsed microjet actuator, since the frequency of its operation was not only small but about one-hundredth of the natural frequency of the system. This implies that the mechanism of pulsing microjet-injection leading to noise reduction could be due to a modification of the input amplitude at different frequencies. That is, we model the effects of the pulsed microjet-control, via a transfer function $G_f(s)$. Given that the control input is at a frequency which is significantly smaller than the natural frequency, and since it can be viewed as a constant, compared to the timescales of the dominant variables of the impinging jet-flow, we model the actuation input into $G_f(s)$ as a parameter, rather than a time-varying input. That is, the impact of control is represented as

$$P_{int} = G_f(s, \theta) P_{shear}, \quad \theta = f(\phi_u) \tag{33.19}$$

where ϕ_u corresponds to the parameters of the pulsing microjet whose velocity is u, and θ represents the parameters of $G_f(s)$ (see Figure 33.17). This model becomes nonlinear if f is nonlinear with respect to ϕ_u.

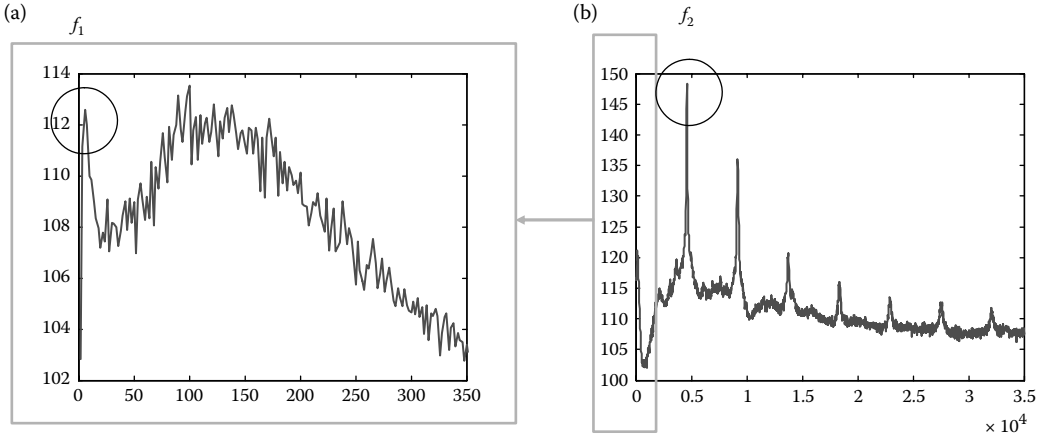

FIGURE 33.15 Spectral plot of the pressure responses of the uncontrolled flow in (a) the low-frequency region, (b) the high-frequency region, at $h/d = 3.5$, NPR = 3.7; f_1 and f_2 denote the dominant natural frequencies of the uncontrolled system.

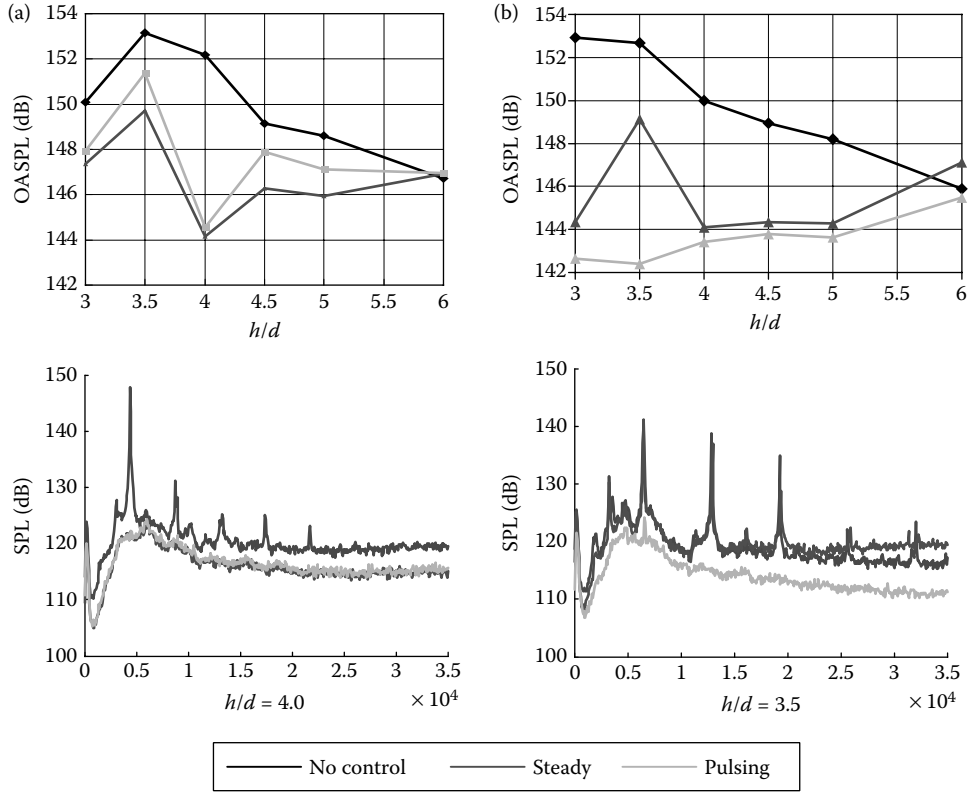

FIGURE 33.16 Experimental pressure responses obtained using the pulsing actuator at different duty cycles; (a) $d_c = 42\%$, (b) $d_c = 74\%$. The pulsing frequency was chosen to be $f = 121$ Hz. Both the OASPL and SPL are indicated in the figures.

In order to motivate the control model in Equation 33.19, we examine its input, P_{shear}, and output, P_{int} in more detail. The input P_{shear}, represents the acoustic input at the nozzle from the reflected acoustics after impingement. The output P_{int} represents the effect of pulsed injection which modifies P_{shear}. This output in turn excites the shear layer that is formed and convects down. The results shown in Figure 33.16 clearly indicate that P_{mic} varies significantly due to the action of the pulsed microjets, and since they vary the flowfield primarily around the nozzle, their impact can be modeled as a variation of the incoming

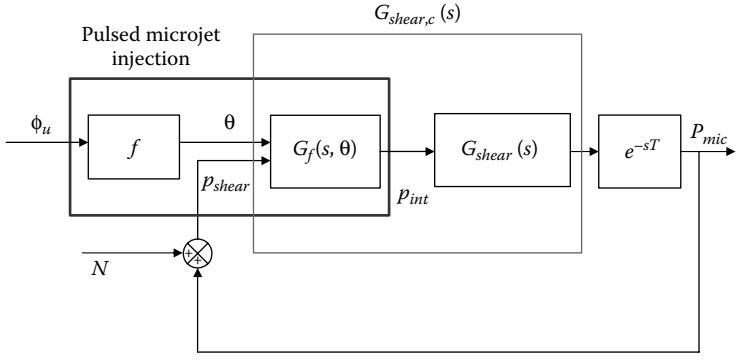

FIGURE 33.17 A nonlinear control model of the pulsed-microjet control of the impinging jets.

pressure field P_{shear} as in Equation 33.19. In particular, their primary impact is a variation in the amount of excitation that P_{shear} contains at the two dominant frequencies ω_1 and ω_2, which can be expressed compactly via a change in θ. In the absence of any microjet injection, it therefore follows that the transfer function $G_f(s, \theta) = 1$. In what follows, for ease of notation, we omit the argument θ.

The problem that remains is the determination of how $G_f(s)$ varies as a function of the different pulsing parameters such as frequency, amplitude, and phase. This is an extremely challenging task due to many reasons. First, $G_f(s)$ is a subcomponent of the overall system, and $G_{shear}(s)$, by and large, is unknown. Second, we note that at any time, P_{mic} is the only sensor that can be measured. This is due to the fact that P_{shear} cannot be measured since the rotating cap that generates the pulsed microjet-injection precludes the location of any sensors near the nozzle exit. We therefore determine $G_f(s)$ in the following manner.

We note from Figures 33.8 and 33.17 that the predominant effect of the pulsed microjet injection is to lead to a reduced excitation of the dominant frequency. This implies that if we denote $G_{shear,c}(s) = G_f(s)G_{shear}(s)$ that $G_{shear,c}$ is essentially of the form

$$G_{shear,c}(s) = \sum_{i=1}^{N_{mode}} \frac{K_{i,c}}{s^2 + 2\zeta_{i,c}\omega_i s + \omega_i^2} \tag{33.20}$$

where $K_{i,c}$ and $\zeta_{i,c}$ are different from K_i and ζ_i, for $i = 1, 2$. The corresponding closed-loop system, denoted as $G_{closed,c}(s)$ is given by

$$G_{closed,c}(s) = \frac{G_{shear,c}(s)e^{-sT}}{1 - G_{shear,c}(s)e^{-sT}}. \tag{33.21}$$

The parameters $K_{i,c}$ and $\zeta_{i,c}$ are then determined exactly in the same manner as in the uncontrolled case, by matching the experimental response of P_{mic} and the response of the closed-loop system $G_{closed,c}(s)$ for the noise input N as in the uncontrolled case. From $G_{shear,c}$ and $G_{shear}(s)$, $G_f(s)$ is determined as

$$G_f(s) = G_{shear,c}(s)G_{shear}^{-1}(s) \triangleq \frac{Z_1(s)}{Z_2(s)} \tag{33.22}$$

The parameter θ in Equation 33.19 therefore corresponds to the coefficients of $Z_1(s)$ and $Z_2(s)$.

We validate the above model as well as determine the structure of the mapping f in Equation 33.19 below. At a height of $h/d = 3.5$ and NPR = 3.7, the values $K_{i,c}$ and $\zeta_{i,c}$ obtained for steady and pulsed-injection with pulsing parameters $d_c = 54\%$ and $P_{supply} = 100$ psig, for pulsing frequencies 10.8 Hz, 16.4 Hz, 21.6 Hz, and 118 Hz are indicated in Table 33.1. The resulting responses P_{mic} for all these cases were obtained in a single experimental run, and hence can be compared to the same baseline case. We note that in the uncontrolled case, $K_{1,c} = K_1$ and $K_{2,c} = K_2$. It was found that the control gains $K_{i,c}$ and $\zeta_{i,c}$ varied very little with the pulsing frequency above 60 Hz. The mapping f in Equation 33.19 is determined therefore by the relation between the second column, an element of ϕ_u, and seventh column, which represent elements of θ, in Table 33.1, and can be seen to be nonlinear. It is interesting to note that

TABLE 33.1 $K_{i,c}$ and $\zeta_{i,c}$ for different pulsing parameters. The high-frequency mode (ω_1) = 4.6 kHz, the low frequency mode (ω_2) = 10 Hz

	f_p (Hz)	$K_{1,c}$	$K_{2,c}$	$\zeta_{1,c}$	$\zeta_{2,c}$	Zeros of $G_f(s)$	Poles of $G_f(s)$
No control		2.45E + 06	475	0.001	0.1		
Steady injection		2.33E + 06	480	0.001	0.1	−6.29 ± 418.14j	−6.29 ± 405.93j
Pulsed injection	118.0	2.30E + 06	481	0.001	0.1	−6.29 ± 450.60j	−6.29 ± 405.93j
Pulsed injection	21.6	1.8E + 06	460	0.001	0.1	−6.29 ± 464.70j	−6.29 ± 405.93j
Pulsed injection	16.4	1.8E + 06	465	0.001	0.1	−6.29 ± 646.17j	−6.29 ± 405.93j
Pulsed injection	10.8	1.85E + 06	470	0.001	0.1	−6.29 ± 463.36j	−6.29 ± 405.93j

variations in the pulsing frequency have a negligible effect on the poles of $G_f(s)$. The parameters $K_{i,c}$ and $\zeta_{i,c}$ corresponding to the duty-cycle $d_c = 76\%$ are not included in Table 33.1 since it needed the installation of a different rotating cap thereby necessitating a different experimental run with a different baseline response. However, a similar table could be constructed at that duty-cycle as well, with the parameters determined in the same manner as the pulsing frequency varied, using data in a single experimental run.

We now examine P_{int} obtained with pulsed and steady injection, and compare it with the uncontrolled case. Figure 33.18 illustrates that the uncontrolled system produces an input excitation P_{shear} (equal to P_{int} in this case), which has a large amplitude at the impinging frequency of 4.6 kHz. Relatively speaking, with a pulsed microjet, the input P_{shear}, which is now modified due to the presence of the pulsed microjet action as P_{int}, has a much smaller amplitude at the same frequency. It should also be noted from Figure 33.18d and f that the pulsed microjet increases the input amplitude at the lower frequency. However, due to the fact that the overall flow field is such that the lower frequency has a much higher damping, the pressure response at this frequency is not increased despite the increase at the input. This could be the reason for the effective noise reduction due to the pulsed microjet.

For comparison, the effect of a steady microjet is also modeled using the same transfer functions, the results of which are illustrated in Figures 33.18b and e. A comparison of Figures 33.18d–f shows two facts; first, the same input-shaping effect as in the pulsed microjet is exhibited in the steady microjet in that the input excitation at a higher frequency of 4.6 kHz is lowered (by 5 dB) and is increased (by 5 dB) at a lower frequency of 50 Hz. In contrast, with pulsed injection, the input excitation at 4.6 kHz is lowered by 20 dB, and increased by 18 dB at 50 Hz. This could be the reason for the noise reduction in the pulsed injection case to be greater than that in the steady case. These discussions also indicate that the parameter θ in Equation 33.19 needs to be chosen such that the output P_{int} of $G_f(s, \theta)$ has a redistributed frequency response, with reduced amplitudes at the dominant impinging tones.

The above implies that an optimal pulsed microjet injection is one that perhaps reduces the excitation at around 4.6 kHz even further and increases the amplitude correspondingly at the lower frequency of

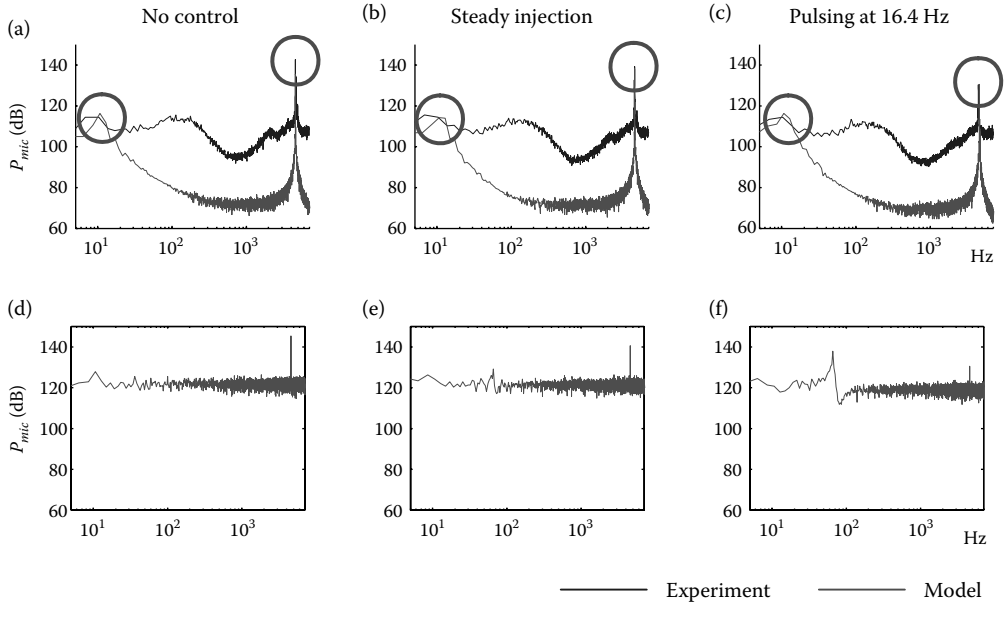

FIGURE 33.18 Spectral plots of P_{mic} (a)–(c) and the corresponding P_{int} (d)–(f) in the uncontrolled and controlled cases. The steady-injection case is included for the sake of comparison. In (a)–(c), the model predictions are compared with the corresponding experimental response. In (c), the pulsing frequency was 16.4 Hz.

50 Hz. Denoting the transfer function of the corresponding controller as $G_{f,optimal}(s, \theta^*)$, where θ^* was computed as follows. Defining

$$G_{shear,optimal}(s) = \sum_{i=1}^{N_{mode}} \frac{K_{i,optimal}}{s^2 + 2\zeta_{i,optimal}\omega_i s + \omega_i^2} \tag{33.23}$$

the gains $K_{i,optimal}$ and $\zeta_{i,optimal}$ were adjusted until the corresponding P_{int} displayed a frequency response similar to Figure 33.18f with the amplitude at 4.6 kHz decreased by about 30 dB, and at 50 Hz increased by 25 dB. It should be noted that the 30 dB increase and the 25 dB decrease were somewhat arbitrary choices and represented a desirable, yet feasible, performance that can be expected from the controlled impinging jet. The corresponding $G_{f,optimal}(s, \theta^*)$ was computed as

$$G_{f,optimal}(s, \theta^*) = G_{shear,optimal}(s) G_{shear}^{-1}(s).$$

The corresponding parameters were given by $K_{1,optimal} = 0.7\text{E}+06$, $K_{2,optimal} = 700$, $\zeta_{1,optimal} = 0.001$, and $\zeta_{2,optimal} = 0.1$. The associated zeros and poles of this transfer function are shown in Figure 33.20. The corresponding P_{mic} that can be achieved with such a P_{int} is shown in Figure 33.19a which illustrates a better reduction of the resonances compared to that in Figures 33.18a–c. The remaining problem then is to determine f, the mapping between the pulsing velocity and θ^* so that the performance in Figure 33.19 can be experimentally achieved using $f^{-1}(\theta^*)$. This is currently under investigation.

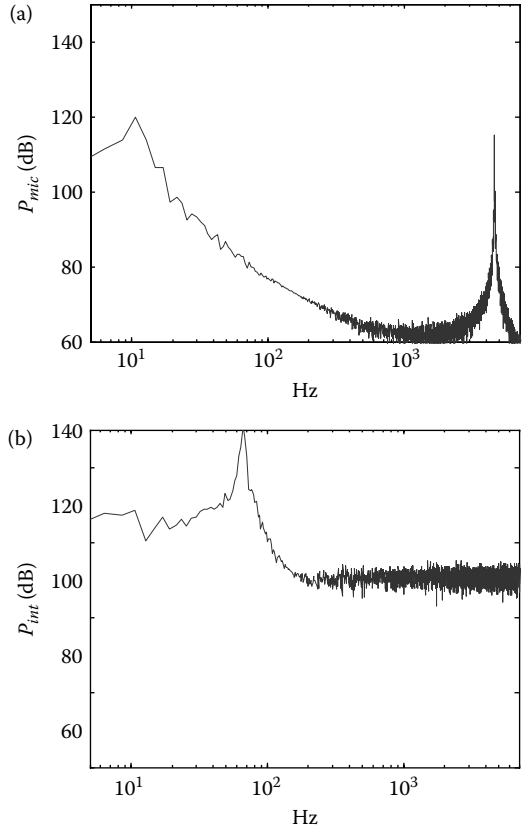

FIGURE 33.19 Spectral plots of (a) P_{mic} and (b) the corresponding P_{int} for the case of an optimal pulsed actuator.

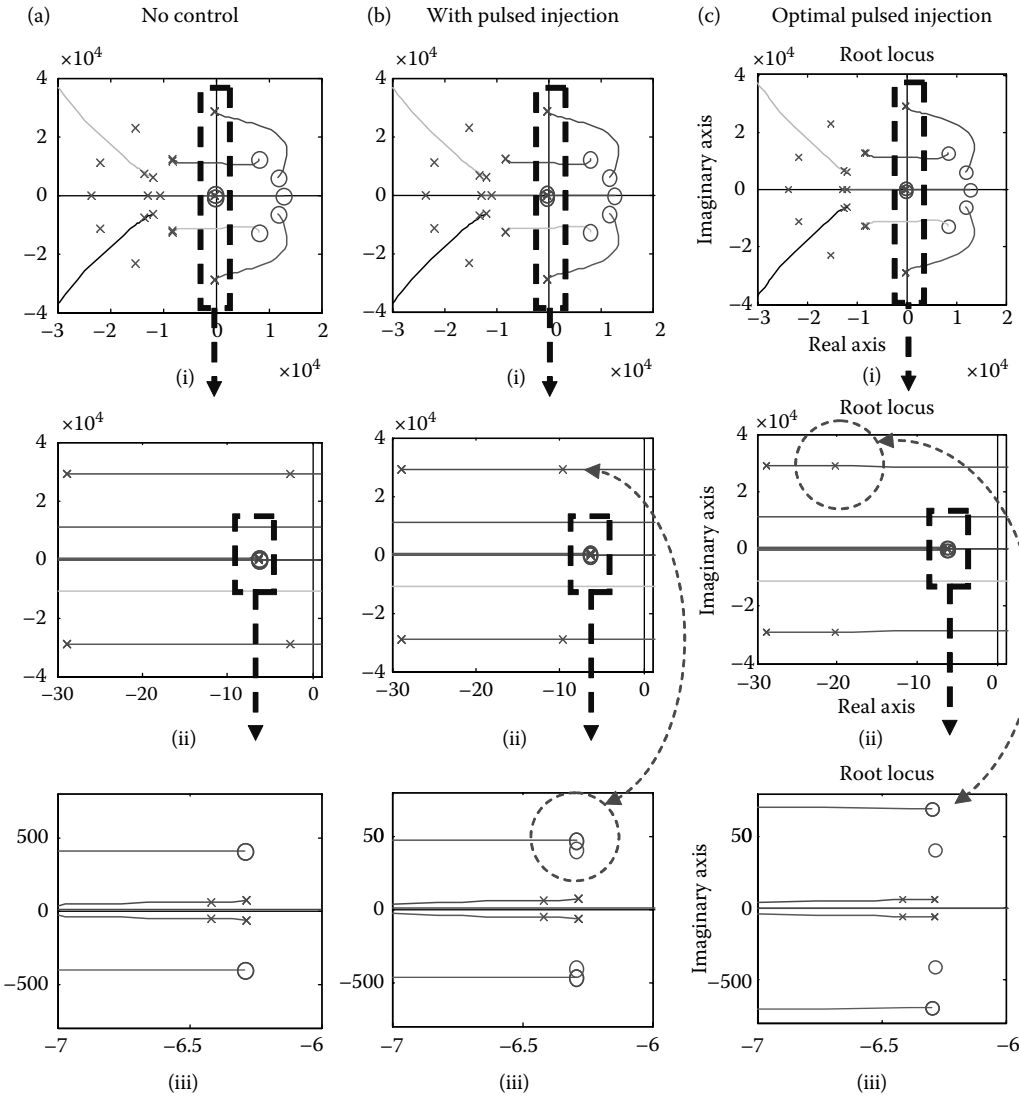

FIGURE 33.20 Pole-zero locations of the forward-loop, and the closed-loop poles for (a) the uncontrolled case (b) the pulsed injection case, at a frequency of 16.4 Hz, and (c) the optimal pulsed-injection case.

Acknowledgments

This work has been supported in part by Office of Naval Research, contract N00014-05-1-0252, and in part by the Air Force Office of Scientific Research with Drs. Schmisseur and Jeffries as program managers.

References

1. C. Fleming, Turbine makers battle innovation trap, *Wall Street Journal*, (Eastern edition). New York, NY, February 13, 1998.
2. J. Rayleigh, *The Theory of Sound*. New York: Dover, vol. 2, 1945.

3. A. A. Peracchio and W. Proscia, Nonlinear heat release/acoustic model for thermoacoustic instability in lean premixed combustors, in *ASME Gas Turbine and Aerospace Congress*, Sweden, 1998.
4. E. Gutmark, T. Parr, K. Wilson, D. Hanson-Parr, and K. Schadow, Closed-loop control in a flame and a dump combustor, *IEEE Control Systems*, vol. 13, pp. 73–78, April 1993.
5. B. Zinn, *Pulsating Combustion. Advanced Combustion Methods*. London: Academic Press Inc. (London) LTD., 1986.
6. K. McManus, T. Poinsot, and S. Candel, A review of active control of combustion instabilities, *Energy and Combustion Science*, vol. 19, no. 1, pp. 1–30, 1993.
7. A. Powell, On the edgetone, *Journal of the Acoustical Society of America*, vol. 33, no. 4, pp. 395–409, April 1961.
8. B. Chu, Stability of systems containing a heat source—The Rayleigh criterion, NASA Research Memorandum RN 56D27, Technical Report, 1956.
9. B. T. Zinn and M. E. Lores, Application of the Galerkin method in the solution of nonlinear axial combustion instability problems in liquid rockets, *Combustion Science and Technology*, vol. 4, pp. 269–278, 1972.
10. F. Culick, Nonlinear behavior of acoustic waves in combustion chambers, *Acta Astronautica*, vol. 3, pp. 715–756, 1976.
11. M. Fleifil, J. P. Hathout, A. M. Annaswamy, and A. F. Ghoniem, The origin of secondary peaks with active control of thermoacoustic instability, *Combustion Science and Technology*, vol. 133, pp. 227–265, 1998.
12. A. Dowling, Nonlinear self-excited oscillations of a ducted flame, *Journal of Fluid Mechanics*, vol. 346, pp. 271–290, 1999.
13. S. Evesque, A. Dowling, and A. Annaswamy, Adaptive algorithms for control of combustion, in *NATO RTO/AVT Symposium on Active Control Technology for Enhanced Performance in Land, Air, and Sea Vehicles*, Braunschweig, Germany, May 2000.
14. M. Fleifil, A. Annaswamy, and A. Ghoniem, A physically based nonlinear model of combustion instability and active control, *Proceedings of Conference on Control Applications*, Trieste, Italy, August 1998.
15. J. Rumsey, M. Fleifil, A. Annaswamy, J. Hathout, and A. Ghoniem, Low-order nonlinear models of thermoacoustic instabilities and linear model-based control, *Proceedings of the Conference on Control Applications*, Trieste, Italy, August 1998.
16. T. Lieuwen and B. Zinn, Experimental investigation of limit cycle oscillations in an unstable gas turbine combustor, *AIAA 2000-0707, 38th AIAA Aerospace Sciences Meeting*, Reno, NV, January 2000.
17. F. Culick, Combustion instabilities in liquid-fueled propulsion systems—An Overview, in *AGARD Conference Proceedings, Paper 1, 450, The 72nd(B) Propulsion and Energetics Panel Specialists Meeting*, Bath, England, 1988.
18. V. Yang and F. Culick, Nonlinear analysis of pressure oscillations in ramjet engines, *AIAA-86-0001*, 1986.
19. F. Culick, Some recent results for nonlinear acoustics in combustion chambers, *AIAA*, vol. 32, pp. 146–169, 1994.
20. A. Annaswamy, Nonlinear modeling and control of combustion dynamics, in *Fluid Flow Control*, P. Koumoutsakos, I. Mezic, and M. Morari, Eds. New York, NY: Springer Verlag, 2002.
21. G. Isella, C. Seywert, F. Culick, and E. E. Zukoski, A further note on active control of combustion instabilities based on hysteresis, *Short Communication, Combustion, Science, and Technology*, vol. 126, pp. 381–388, 1997.
22. G. A. Richards, M. C. Yip, and E. H. Rawlins, Control of flame oscillations with equivalence ratio modulation, *Journal of Propulsion and Power*, vol. 15, pp. 232–240, 1999.
23. R. Prasanth, A. Annaswamy, J. Hathout, and A. Ghoniem, When do open-loop strategies for combustion control work? *AIAA Journal of Propulsion and Power*, vol. 18, pp. 658–668, 2002.
24. G. Stein and M. Athans, The LQG/LTR procedure for multivariable feedback control design, *IEEE Transactions on Automatic Control*, vol. 32, pp. 105–114, 1987.
25. Y. Chu, A. Dowling, and K. Glover, Robust control of combustion oscillations, in *Proceedings of the Conference on Control Applications*, Trieste, Italy, August 1998.
26. A. Annaswamy, M. Fleifil, J. Rumsey, J. Hathout, R. Prasanth, and A. Ghoniem, Thermoacoustic instability: Model-based optimal control designs and experimental validation, *IEEE Transactions on Control Systems Technology*, vol. 8, no. 6, pp. 905–918, November 2000.
27. J. Hathout, M. Fleifil, A. Annaswamy, and A. Ghoniem, Combustion instability active control using periodic fuel injection, *AIAA Journal of Propulsion and Power*, vol. 18, pp. 390–399, 2002.
28. O. Smith, A controller to overcome dead time, *ISA Journal*, vol. 6, 1959.

29. A. Manitius and A. Olbrot, Finite spectrum assignement problem for systems with delays, *IEEE Transactions on Automatic Control*, vol. AC-24 no. 4, 1979.
30. K. Ichikawa, Frequency-domain pole assignment and exact model-matching for delay systems, *International Journal of Control*, vol. 41, pp. 1015–1024, 1985.
31. S. Evesque, Adaptive control of combustion oscillations, Ph.D. dissertation, University of Cambridge, Cambridge, UK, November 2000.
32. S. Niculescu and A. Annaswamy, A simple adaptive controller for positive-real systems with time-delay, in *The American Controller Conference*, Chicago, IL, 2000.
33. P. J. Dines, Active control of flame noise, Ph.D. dissertation, University of Cambridge, 1983.
34. J. E. Tierno and J. C. Doyle, Multimode active stabilization of a Rijke tube, in *DSC-Vol. 38*. ASME Winter Annual Meeting, 1992.
35. B. Pang, K. Yu, S. Park, A. Wachsman, A. Annaswamy, and A. Ghoniem, Characterization and control of vortex dynamics in an unstable dump combustor, *42nd AIAA Aerospace Sciences Meeting & Exhibit*, January 2004.
36. J. Seume, N. Vortmeyer, W. Krause, J. Hermann, C.-C. Hantschk, P. Zangl, S. Gleis, and D. Vortmeyer, Application of active combustion instability control to a heavy-duty gas turbine, in *Proceedings of the ASME-ASIA*, Singapore, 1997.
37. J. Hermann, A. Orthmann, S. Hoffmann, and P. Berenbrink, Combination of active instability control and passive measures to prevent combustion instabilities in a 260 MW heavy duty gas turbine, in *NATO RTO/AVT Symposium on Active Control Technology for Enhanced Performance in Land, Air, and Sea Vehicles*, Braunschweig, Germany, May 2000.
38. A. Krothapalli, E. Rajakuperan, F. S. Alvi, and L. Lourenco, Flow field and noise characteristics of a supersonic impinging jet, *Journal of Fluid Mechanics*, vol. 392, pp. 155–181, 1999.
39. A. M. Annaswamy, J. J. Choi, and F. S. Alvi, Pulsed microjet control of sueprsonic impinging jets via low-frequency excitation, *Journal of Systems and Control Engineering*, vol. 222, no. 5, pp. 279–296, 2008.
40. A. Powell, On edge tones and associated phenomena, *Acoustica*, vol. 3, pp. 233–243, 1953.
41. C. W. Rowley, D. R. Wilianms, T. Colonius, R. Murray, D. MacMartin, and D. Fabris, Model-based control of cavity oscillatios, part ii: System identification and analysis, *AIAA Paper*, no. 2002-0972, 2002.
42. N. Zhuang, Experimental investigation of supersonic cavity flows and their control, Ph.D. dissertation, Florida State University, 2007.
43. F. S. Alvi, C. Shih, R. Elavarasan, G. Garg, and A. Krothapalli, Control of supersonic impinging jet flows using supersonic microjet, *AIAA Journal*, vol. 41, no. 7, pp. 1347–1355, 2003.
44. H. Lou, F. S. Alvi, C. Shih, J. Choi, and A. M. Annaswamy, Flowfield properties of supersonic impinging jets with active control, *AIAA Paper*, no. 2002-2728, 2002.
45. J. Choi, A. M. Annaswamy, F. S. Alvi, and H. Lou, Active control of supersonic impingement tones using steady and pulsed microjets, *Experiments in Fluids*, vol. 41, no. 6, pp. 841–855, 2006.
46. W. W. Bower, V. Kibens, A. Cary, F. Alvi, G. Raman, A. Annaswamy, and N. Malmuth, High-frequency excitation active flow control for high speed weapon release (HIFEX), *AIAA Paper*, no. 2004.

34

Modeling and Control of Air Conditioning and Refrigeration Systems

Andrew Alleyne
University of Illinois at Urbana–Champaign

Vikas Chandan
University of Illinois at Urbana–Champaign

Neera Jain
University of Illinois at Urbana–Champaign

Bin Li
University of Illinois at Urbana–Champaign

Rich Otten
University of Illinois at Urbana–Champaign

34.1 Introduction

Air conditioning and refrigeration (AC&R) systems are ubiquitous in modern society since they perform the key engineering function of transporting thermal energy from one physical location to another. In doing so, they are able to change the condition of a defined spatial environment to prescribe a particular temperature and humidity. While this may seem to be a simple task, it has had tremendous impact on the way we live and work. It is safe to say that nearly all people in the developed countries have interacted with AC&R systems throughout their lives. Whether traveling in an air-conditioned car or using products produced in an air-conditioned factory, there are very few people who have not had their lives touched by this valuable technology. Two key examples, buildings and refrigerated transport, give more detail as to the impact level of these systems on our society.

Buildings are a dominant mode of energy usage in the world today. As illustrated in Figure 34.1, they comprise over 1/3 of the total energy used in the United States [1]. Additionally, their carbon footprint is similarly high, accounting for greater carbon emissions than the transportation sector [2]. The heating, ventilation, and air-conditioning (HVAC) systems within these buildings are one of the major sources of the energy consumed, as illustrated in Figure 34.1. In fact, HVAC systems use in the residential and commercial sectors alone consumes 20% of all energy in the United States [1], and AC&R systems are the largest contributors to peak electrical demand on the grid [3]. Proper control of these systems is essential to minimize energy usage while maintaining occupant comfort. Occupant comfort is particularly important

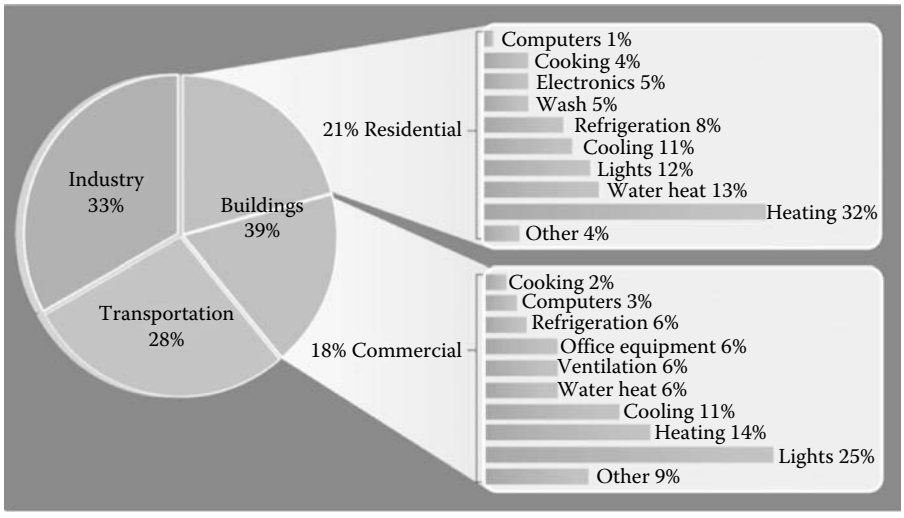

FIGURE 34.1 U.S. energy use within buildings. (Adapted from Buildings Energy Data Book 2008. Also, http://buildingsdatabook.eere.energy.gov.).

since people spend more than 90% of their lives inside the buildings whose interiors are conditioned by these systems. In addition, other valuable systems, such as computer servers, reside within buildings and must be thermally controlled.

In addition to their significant energy usage, AC&R systems have tremendous societal impact. As evidence, AC&R has been identified as one of the 20 great engineering achievements of the twentieth century [5]. AC&R enabled a decoupling of food production from population centers by allowing food to be produced geospatially disparately from where it would be consumed. AC&R provided the means for perishable food to be safely transported over long distances, something unavailable at the start of the twentieth century. Cities and suburbs could grow without needing to have agriculture nearby to sustain them, radically altering the national landscape. Simultaneously, AC&R enabled domestic food storage, which allowed a relatively large separation in time between when food was purchased and when it was consumed in residences. This very greatly influenced societal flexibility that by the end of the twentieth century, the market penetration for AC&R in residences was over 99.5%. No other technology could claim this level of ubiquity.

This chapter will illustrate the underlying phenomena governing the behavior of the basic thermodynamic cycle for AC&R systems as well as the role that control systems play in their performance. The remainder of this chapter is organized as follows. An overview of the fundamental operation of AC&R systems is given in Section 34.2. This covers the basic components as well as the control objectives. Section 34.3 illustrates the system dynamics of the components that comprise a typical AC&R system. This section also identifies some of the control challenges that are associated with system nonlinearities. Section 34.4 gives an overview of typical approaches to meet the control objectives outlined in Section 34.2. These would be the most common types of controllers available at present. Section 34.5 discusses some of the more advanced control strategies that have been demonstrated on AC&R systems. A brief set of concluding remarks is given in Section 34.6.

34.2 AC&R Fundamentals

A typical AC&R system in operation is shown in Figure 34.2. Here a refrigeration unit is attached to the cargo compartment of a trailer as it transports perishable food. Figure 34.2b gives a detailed view of the AC&R unit that conditions the space within the trailer.

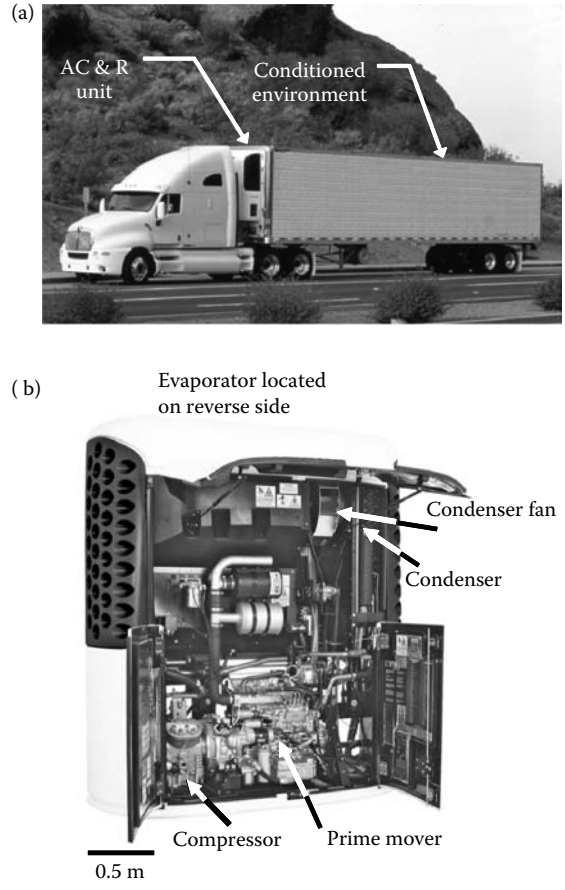

FIGURE 34.2 (a) AC&R unit on a trailer and (b) standalone AC&R unit. (Courtesy: Thermo King Corp.)

The schematic representation of the AC&R system shown in Figure 34.2 is given in Figure 34.3 which shows several components interconnected to form a complete system. The ambient side of the system rejects hot air to the external environment and the refrigerated side pulls heat from the conditioned environment. There are several components in the physical system of Figure 34.2b that have been abstracted away in Figure 34.3, including the prime mover operating the compressor.

The fundamental refrigeration process involves moving a fluid through different phases in a closed thermodynamic cycle. A further abstraction of the physical process is depicted in Figure 34.4a which indicates the most basic components used in creating this cycle along with the ideal thermodynamic cycle on a pressure–enthalpy (P–h) diagram [6,7] in Figure 34.4b. This particular cycle is termed a vapor compression cycle and a comparison among Figures 34.2b, 34.3, and 34.4a indicate the level of abstractions performed between the physical system and the idealized one. The P–h diagram shown in Figure 34.4b is for a subcritical cycle and the reader is referred to [7] for other thermodynamic cycles. The basic refrigeration cycle is composed of four primary components: an evaporator, a compressor, a condenser, and an expansion device. Beginning at the condenser inlet, the high-pressure superheated vapor flows through the condenser rejecting heat to the ambient air flowing over the condenser coil. As the fluid cools, it condenses and becomes a two-phase mixture of vapor and liquid, eventually becoming a liquid at the outlet of the condenser. From the condenser, the refrigerant flows through an expansion device and transitions from a liquid to a two-phase mixture at a lower pressure and temperature. Next, the cold refrigerant enters the evaporator where heat is absorbed from the conditioned environment's

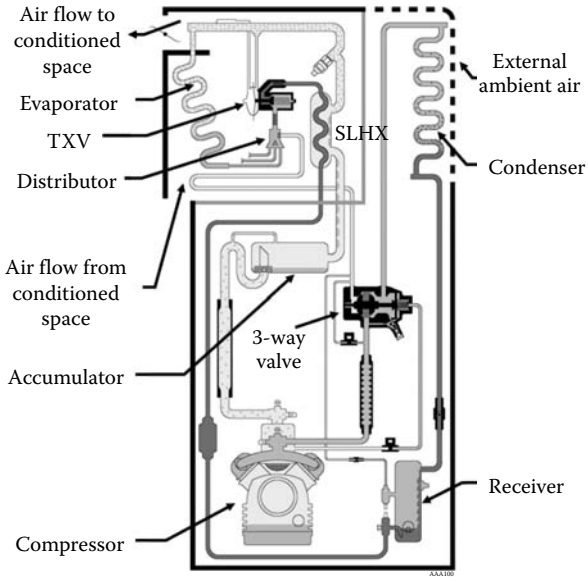

FIGURE 34.3 Schematic representation of Thermo King trailer unit. (Courtesy: Thermo King Corp.)

air flow over the evaporator coil and the refrigerant evaporates. Within the evaporator, the fluid transitions from a two-phase mixture to a vapor as heat is absorbed by the refrigerant. The vapor exiting from the evaporator enters the compressor, is compressed to a higher pressure, and then continues cycling through the system.

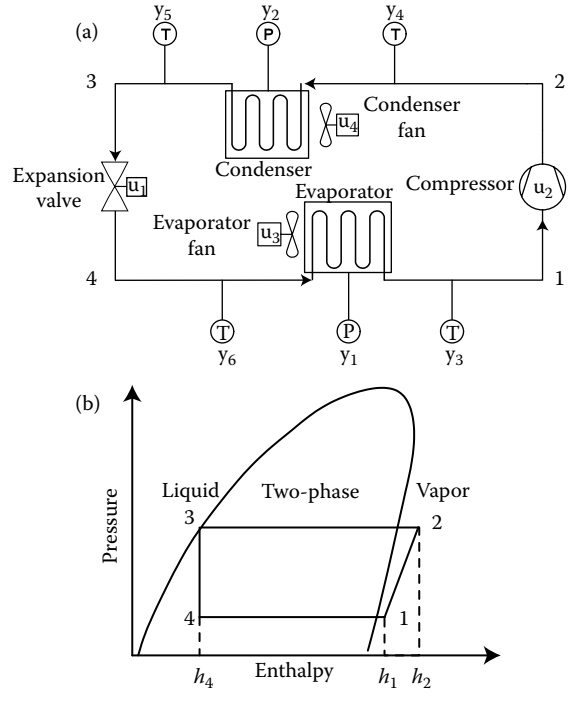

FIGURE 34.4 (a) Ideal subcritical vapor compression cycle and (b) pressure–enthalpy diagram.

The secondary fluid carrying thermal energy from the conditioned space in Figures 34.2 and 34.3 is assumed to be air. Many AC&R systems interface their evaporator to a liquid loop as a secondary fluid since it has the potential for high heat transfer and can transport thermal energy much more efficiently than air over long distances. These are termed *chiller systems* and the cooled secondary fluid exiting the evaporator is transported a significant distance to then condition a space. Such systems are common in large buildings or large manufacturing/process plants where economies of scale are gained by centralizing the AC&R plant and then distributing the conditioning capabilities through a secondary fluid [6].

There are a number of variations that can be made to the standard vapor compression cycle. As illustrated in Figure 34.3, a receiver is frequently placed at the condenser outlet and is used as storage for excess refrigerant in the system. The receiver forces the refrigerant condition at the receiver outlet to be that of a saturated liquid. This ensures a phase change upon refrigerant expansion through the expansion device. Accumulators are occasionally placed at the outlet of the evaporator to ensure that vapor is entering the compressor thereby preventing potentially damaging liquid from entering. Internal heat exchangers, such as the suction line heat exchanger (SLHX) in Figure 34.3, are also used in variations of the basic vapor compression cycle system to increase system efficiency. Although there is a great deal of variety in specific system configurations, the basic operation of the four major components forces these systems to behave somewhat similarly from a thermodynamic cycle point of view. These variations on the basic cycle are further explained in [7] and other textbooks in this subject.

34.2.1 Control Objectives

The primary objectives of AC&R systems are twofold. First, they must provide the demanded cooling to the conditioned environment defined as the cooling *capacity* of the system. The difference between h_1 and h_4 in Figure 34.4b represents the increase in enthalpy across the evaporator, that is, the amount of heat (Q) removed from the conditioned environment. This is a measure of evaporator capacity. Second, the AC&R system should be as efficient as possible. The difference between h_2 and h_1 represents the increase in enthalpy across the compressor, that is, the amount of work (W) done by the compressor to increase the pressure of the refrigerant vapor. The system coefficient of performance (COP), a measure of system efficiency, is defined as the ratio between the two changes in enthalpy. Maximizing this value is a key systems-level priority since it will minimize energy usage for a given amount of cooling.

$$COP = \frac{|Q|}{W} \approx \frac{h_4 - h_1}{h_1 - h_2}. \tag{34.1}$$

In addition to the two primary performance goals, there are additional control-related goals. It is important that key system components are protected at all times. To prevent the possibility of liquid entering the compressor inlet, it is important to maintain a prescribed level of superheated vapor at the exit of the evaporator. The control of superheat and capacity are regulation problems while the energy use is a minimization problem.

34.2.2 Input–Output Pairs

Figure 34.4a illustrates the inputs (u_i, $i \in [1, 4]$) and outputs (y_i, $i \in [1, 6]$) for the vapor compression cycle. The potentially controllable inputs available for the vapor compression cycle system given in Figure 34.4 are the fan speeds of the two heat exchangers, the compressor speed, and the expansion device opening. It should be noted that for most commercial systems, these are not individually controllable due to cost concerns; they are introduced here for generality. The fans control the mass flow of air across the heat exchangers and the compressor/valve pair controls the mass flow of refrigerant through the heat exchangers. The possible outputs of the system are functions of the pressures and temperatures of all the four corners in the diagram of Figure 34.4b. Simplifying assumptions of isobaric fluid conditions throughout the heat exchangers result in two pressures (condenser, evaporator) and four temperatures (compressor inlet/outlet, expansion valve inlet/outlet) as six total system outputs associated with the

refrigerant loop. Combinations of these can then be transformed to more industrially common output variables such as superheat: the temperature value of the refrigerant above saturated vapor at the exit of the evaporator.

$$T_{sh} = T_g - T_{sat}(P_e). \tag{34.2}$$

Additionally, from Figure 34.4, cooling capacity can be a calculated as function of the difference between the refrigerant inlet (T_4) and outlet (T_1) temperatures for a given pressure (P_e) in the evaporator. Air-side measurements, such as the evaporator air inlet temperature, or heat exchanger wall temperatures can also be used in feedback. These are often easier to obtain since they do not involve sensors immersed within the refrigerant. There are additional inputs and outputs that can be considered; for example, Jensen and Skogestad [8] utilized refrigerant charge as an input and condenser subcooling as an output for a particular optimal control approach. Additionally, for chiller applications [9], the secondary loop temperature characteristics can be used as outputs with flow control valves as inputs. A more extensive set of output descriptions can be found in [6,7].

34.3 Basic System Dynamics

Understanding the complex dynamics of AC&R systems is vital to proper control. In this section we give basic starting points for system modeling and separate the exposition by time scales. The mass flow devices (compressor and valve) in Figure 34.4 are described algebraically. The energy flow devices (evaporator and condenser heat exchangers) are described by dynamics of varying degrees of complexity depending on the approach taken.

34.3.1 Mass Flow Devices

Using the simplified system given in Figure 34.4, the basic behavior of the four components can be described. Assuming a positive displacement compressor, two simple algebraic relationships are usually sufficient to create a model of its mass flow rate. Mass flow rate is calculated in Equation 34.3, where $\rho_k = \rho(P_{k,in}, h_{k,in})$, and a volumetric efficiency, η_{vol}, is assumed. Additionally, compression can be assumed to be an adiabatic process with an isentropic efficiency, and therefore the relationship between the entrance and exit enthalpies is given in Equation 34.4, where $h_{out,isentropic} = h(P_{out}, s_k)$ and $s_k = s(P_{in}, h_{in})$. This can be rearranged to give Equation 34.5. Both the volumetric and isentropic efficiencies are assumed to change with operating condition and are given by semiempirical maps (Equations 34.6 and 34.7), where $P_{ratio} = \frac{P_{out}}{P_{in}}$.

$$\dot{m}_k = \omega_k V_k \rho_k \eta_{vol}, \tag{34.3}$$

$$\frac{h_{out,isentropic} - h_{in}}{h_{out} - h_{in}} = \eta_k, \tag{34.4}$$

$$h_{out} = \frac{1}{\eta_k}\left[h_{out,isentropic} + h_{in}(\eta_k - 1)\right], \tag{34.5}$$

$$\eta_{vol} = f_1(P_{ratio}, \omega_k), \tag{34.6}$$

$$\eta_k = f_2(P_{ratio}, \omega_k). \tag{34.7}$$

An example of a volumetric efficiency mapping is shown in Figure 34.5 along with the data used to create it. If necessary, the static nonlinear compressor model can be linearized around a particular system operating condition. This results in a linear compressor model of the form $\dot{m}_k = Gain_k \cdot \omega_k$ that can be used for control design and to study the parametric sensitivity of the system.

Similar to the compressor, two algebraic relationships can be used to model a fixed opening expansion device. Mass flow rate is calculated assuming standard orifice flow (Equation 34.8) and using a

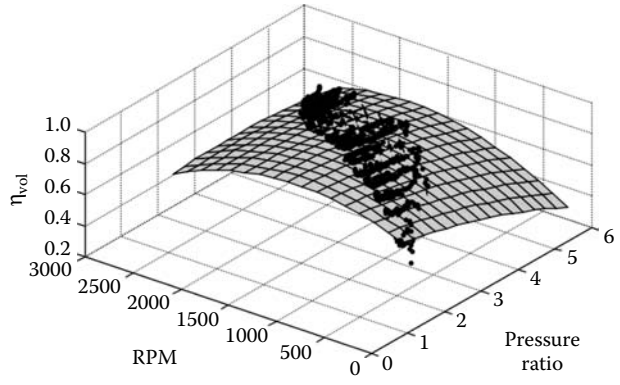

FIGURE 34.5 A two input performance map of volumetric efficiency with data superimposed.

semiempirical map for the discharge coefficient (Equation 34.9). The discharge coefficient is assumed to be a function of valve area opening input, u_v, and pressure differential, $\Delta P = (P_{in} - P_{out})$. Additionally, expansion is assumed to be an isenthalpic process (Equation 34.10).

$$\dot{m}_v = C_d \sqrt{\rho(P_{in} - P_{out})}, \tag{34.8}$$

$$C_d = f_3(u_v, \Delta P), \tag{34.9}$$

$$h_{v,in} = h_{v,out}. \tag{34.10}$$

This basic approach can represent static orifice tubes and controlled area electronic expansion valves (EEVs) quite well. The area opening model can be linearized around a particular system operating condition. This results in a linear model of the form $\dot{m}_v = Gain_v \cdot u_v$ that can be used for control design. Components such as thermostatic expansion valves (TXVs) have significant dynamics based on their construction. Equations 34.8 through 34.10 could then be augmented with an appropriate lag filter [10] forced by an evaporator superheat temperature.

$$u_v = \frac{K_{TXV}}{\tau_{TXV} s + 1} T_{sh}. \tag{34.11}$$

34.3.2 Heat Exchanger Models

In general, the three modeling paradigms that have been applied to modeling of heat exchangers for AC&R systems are lumped parameter, finite volume (or discretized), and moving boundary models. The most important task in modeling AC&R systems is effectively capturing the behavior of the heat exchangers [9,12], since they dominate the dynamic behavior of the system. Refs. [9,11,12] provide literature reviews of the relevant vapor compression system modeling efforts.

The first paradigm is termed here as the lumped parameter model. Lumped parameter heat exchanger models attempt to capture the behavior of a heat exchanger with a single lumped heat transfer parameter. These models are commonly presented in textbooks, as illustrated in [13]. Often, lumped parameter models are used to model vapor compression systems in conjunction with some other component (e.g., the cabin of a car or a room in a building). In this case, the focus of the modeling effort is not on the dynamics of the vapor compression system but on the cooling of the conditioned environment. The simplicity of lumped parameter models tends to be insufficient to capture the dynamic response of some important system outputs (e.g., superheat), and therefore their use in control design is limited.

Finite volume and discretized approaches to the dynamic modeling of vapor compression systems decompose the heat exchanger geometry to a finite set of small regions, allowing spatial effects to be

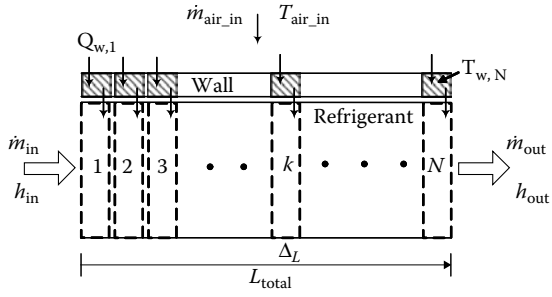

FIGURE 34.6 Schematic representation of a discretized finite volume modeling paradigm.

captured by the model. Figure 34.6 illustrates the concept of discretizing the heat exchanger into many (N) finite volume regions where the accuracy of the system model improves with an increased number of regions. The governing partial differential conservation equations for mass, momentum, and energy are applied to each region, resulting in high-order dynamic models. The complexity of the models is primarily used to capture the spatially varying fluid flow and heat transfer phenomena that occur in compact heat exchangers. Commercial software packages are available that use a finite volume approach to modeling multiphase flow within heat exchangers (e.g., E-Thermal [14], Modelica [15], or SINDA/FLUENT [16]).

The third modeling paradigm is termed the moving boundary approach. Moving boundary models attempt to capture the dynamics of multiple phase flows within a heat exchanger by allowing the effective position(s) of phase change to vary as a function of time. The parameters for each fluid phase region in the heat exchanger are lumped, resulting in a model of fairly low dynamic order. This approach was first presented by Wedekind et al., who proposed using a mean void fraction to develop a transient model of evaporating and condensing flows [17]. As detailed below, the dominant dynamics associated with the multiphase flow within the heat exchanger are captured by the varying interface between fluid phase regions represented in the moving boundary approach. As a result, the moving boundary framework provides models that can accurately predict the behavior of important system outputs that must be controlled to obtain efficient system operation (i.e., superheat, heat exchanger pressure). The lumped parameter nature of each fluid phase region ensures that the overall dynamic model complexity remains low enough to permit the application of known control design techniques. For controls related activity, the low-order dynamic model afforded by this approach usually makes it the model of choice [18].

The moving boundary approach is based on the assumption of 1-dimensional fluid flow with effective diameters, flow lengths, and surface areas. In essence, this treats any heat exchanger as a long thin tube. The approach also assumes uniform pressure throughout the heat exchanger. As shown schematically in Figure 34.7, the heat exchanger is divided into regions based on the fluid phase, and the effective parameters are lumped in each region. The interface between fluid phase regions is allowed to be a dynamic variable and vary throughout the length of the heat exchanger. The following discussion provides an overview of moving boundary models of an evaporator. A comprehensive derivation of the models, as well as other heat exchanger configurations such as condensers and internal heat exchangers, is given in [19,20].

The evaporator model assumes a two-phase flow condition at the heat exchanger inlet that transitions to a single-phase flow at a specific point within the heat exchanger. The location of the interface between these two phase regions is allowed to be a dynamic variable. The governing ordinary differential equations ODEs are obtained by integrating the governing partial differential equations PDEs along the length of the heat exchanger and assuming lumped parameters in each fluid region [18,19].

Several assumptions are made regarding the lumped parameters of the evaporator model. The air temperature used to determine the heat transfer between the walls of the heat exchanger and the air is assumed to be a weighted average of the inlet and outlet air temperatures across each lumped region, $T_a = T_{a,in}(\mu) + T_{a,out}(1-\mu)$, $\mu \in [0,1]$. In the two-phase region, the fluid properties are determined by

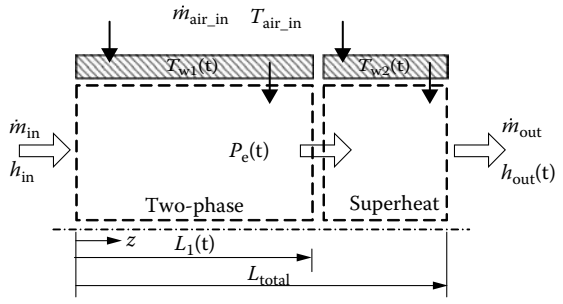

FIGURE 34.7 Diagram of the moving boundary evaporator with two fluid regions.

assuming a mean void fraction; for example, $\rho_1 = \rho_f(1-\bar{\gamma}) + \rho_g(\bar{\gamma})$, $\bar{\gamma} \in [0,1]$. In the superheat region, average properties between the inlet and outlet refrigerant state are used, that is, $h_2 = (h_g + h_{out})/2$, $T_{r2} = T(P_e, h_2)$, and $\rho_2 = \rho(P_e, h_2)$. For the evaporator model, the time derivative of the mean void fraction is neglected. This assumption is valid not only because the change in mean void fraction tends to be small during transients considered, but also because its time dependence is related to dynamic modes that are much faster than the dominant system dynamics. Thus mean void fraction dynamics can usually be replaced with their instantaneous, algebraic equivalents.

As a consequence of assuming uniform pressure throughout the heat exchanger, the momentum of the fluid is assumed to be the same upon exit as it is upon entrance. Therefore, full conservation of momentum equations are not applied; instead there is a simple pressure drop augmentation given at the exit of the heat exchanger. The remaining governing partial differential equations for the conservation of refrigerant mass, refrigerant energy, and heat exchanger wall energy in a fluid region are given by Equations 34.12 through 34.14.

$$\frac{\partial\left(\rho A_{cs}\right)}{\partial t} + \frac{\partial(\dot{m})}{\partial z} = 0, \tag{34.12}$$

$$\frac{\partial\left(\rho A_{cs} h - A_{cs} P\right)}{\partial t} + \frac{\partial(\dot{m}h)}{\partial z} = p_i \alpha_i (T_w - T_r), \tag{34.13}$$

$$(C_p \rho A)_w \frac{\partial(T_w)}{\partial t} = p_i \alpha_i (T_r - T_w) + p_o \alpha_o (T_a - T_w). \tag{34.14}$$

The integration of Equations 34.12 through 34.14 over the two-phase and superheat regions of the evaporator results in the relevant ODEs governing system behavior [18,19].

The resulting ODEs for conservation of refrigerant mass, refrigerant energy, and wall energy for the two-phase and superheat regions contain only five explicit time derivatives: $\dot{L}_1, \dot{P}_e, \dot{h}_{out}, \dot{T}_{w1}$, and $\dot{T}_{w2}$. The equations can be combined to result in the descriptor system as

$$Z(x,u) \cdot \dot{x} = f(x,u). \tag{34.15}$$

The states are shown in Figure 34.7 as $x = [L_1 \quad P_e \quad h_{out} \quad T_{w1} \quad T_{w2}]^{\mathrm{T}}$, and the elements of the $Z(x,u)$ matrix and $f(x,u)$ vector are given in [19]. Here, the inputs to the evaporator model in Equation 34.15 are $u = \left[\begin{array}{ccccc} \dot{m}_{in} & \dot{m}_{out} & h_{in} & \dot{m}_{air\_in} & T_{air\_in} \end{array}\right]^{\mathrm{T}}$.

The nonlinear model presented in Equation 34.15 can be linearized around a particular system operating condition. This results in a linear evaporator model that can be used for control design and to study the parametric sensitivity of the system. The full derivation of the linear model is presented in [19]. The procedure would then be repeated for the condenser heat exchanger with increased states due to the additional refrigerant zone present [19]. A similar approach can be done for other heat exchangers; for example, a counterflow liquid–liquid heat exchanger. The modeling approach can also be augmented in

various ways. For example, a hybrid or switched model can be used when the heat exchanger models given above encounter conditions that would remove one of the zones in the model. Ref. [21] demonstrates a switched system model for a condenser heat exchanger based on the notion of bumpless transfer techniques that allows for large transients, such as compressor on–off cycling, to be accommodated in the models.

34.3.3 Other Component Models

The previous two subsections gave modeling approaches for the four basic components in Figure 34.4. To achieve an accurate representation of practical vapor compression cycle systems, several other components may be needed. These include chiller loops, counterflow heat exchangers, cooling towers, oil separators, filter/driers, and expansion tanks. Additionally, piping elements that model flow splits and flow convergence may also be needed, for example, if there are multiple evaporators for a single condensing unit. Furthermore, it may be necessary to combine functionality of individual components as would be the case of treating heat transfer through pipes as a combination of both mass transport and the heat exchanger model. Refs. [22,23] detail the dynamic modeling of several other types of vapor compression cycle component models along with validation results.

Figure 34.8 summarizes the modeling approach for the moving boundary method. The two mass flow devices can be treated as static nonlinear maps. The two heat exchangers vary in their complexity depending on the amount of state information that is necessary for the task; the condenser has types of fluid regions necessitating additional states. Between them, they contain the dominant dynamics of the systems. The forced air flow across the heat exchangers, similar to the refrigerant mass flow devices, can be represented by static nonlinear maps based on fan charts [24]. This approach has been utilized to develop MATLAB®- and Simulink®-based simulation tools for control-oriented modeling of AC&R systems [19,21–23].

34.3.4 Simplified System Models

The more complex models given above are good for codesign of new controllers and plants in that they allow for parametric variations in system design parameters. Additionally, they are good for embedded applications involving diagnostics and residual generation. However, there is a challenge to direct parameterization of a system representation due to the descriptor form of Equation 34.15. Should the goal be to close a loop around an existing physical system, system identification (ID) techniques are often very suitable. Via ID and model reduction [25] demonstrated quantitatively that the dominant system dynamics correspond to the thermal capacitances of the heat exchanger wall sections. Therefore, the evaporator shown in Figure 34.7 could be represented by two states, the two wall temperatures, with the other characteristics represented by algebraic relations to these states. This order reduction indicates that low-order models can be sufficient for control-based approaches and could be identified using available ID approaches. For single-input single-output (SISO) system models, such as expansion device opening

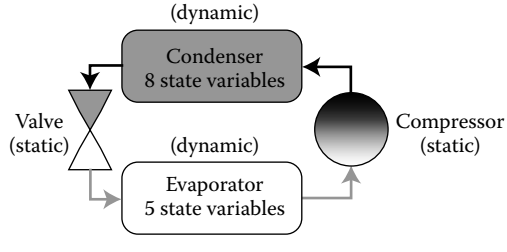

FIGURE 34.8 Summary of moving boundary dynamic modeling structure for AC&R system.

to superheat, a simple transfer function such as Equation 34.16 would be able to capture local system behavior.

$$\frac{T_{sh}}{u_v} = G(s) = \frac{K_e e^{-s \cdot t_{delay}}}{(s + p_e)}. \tag{34.16}$$

It should be noted that the system identification approach could be utilized in conjunction with the higher fidelity simulation models described above. Detailed simulation models would be created and parameterized; then ID techniques would be used on those models to create appropriate, and possibly multivariable, input–output models. These could then be used for control design and system evaluation without having to build physical hardware. Additionally, individual components of a system, such as shown in Figure 34.7, could be provided by system identification while the rest can be constructed from first principles. This "gray box" approach to system modeling can be successfully implemented for hardware-in-the-loop testing as well as system diagnostics.

34.3.5 System Nonlinearity

The system given in Equation 34.16 would be simple to design SISO controllers for if it were linear and time invariant (LTI). However, the parameters in Equation 34.16 change significantly as the system changes operating condition. This is due to the significant system nonlinearity associated with the multiphase fluid flow and heat transfer conditions. An example of this system behavior is given in Figure 34.9 that shows changes in superheat owing to 100 rpm step variations in compressor speed at different operating speeds. The system response varies significantly depending on the operating speed (i.e., cooling capacity) of the compressor. As evidenced, there is up to a factor of 7 change in the steady-state response as the operating condition changes by a factor of 2.5.

This level of nonlinearity makes it challenging to design robust control algorithms that meet performance requirements over the range of conditions these systems are likely to encounter. Additionally, the complex descriptor-based nature of the dynamics described above and in [19,23] makes it difficult to use techniques such as linear parameter varying (LPV) control. The ability to handle large dynamical plant variations in a robust fashion is one of the current and future challenges for AC&R control.

34.4 Basic Control Approaches

34.4.1 Hysteretic On–Off Control

By far, the most common approach to the control of individual AC&R systems is to use the compressor in a cyclic on–off fashion to modulate cooling capacity while engaging a mechanical expansion device

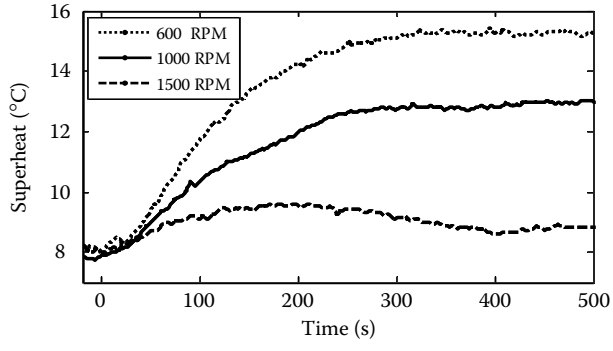

FIGURE 34.9 Step response data depicting nonlinearity in AC&R system dynamics.

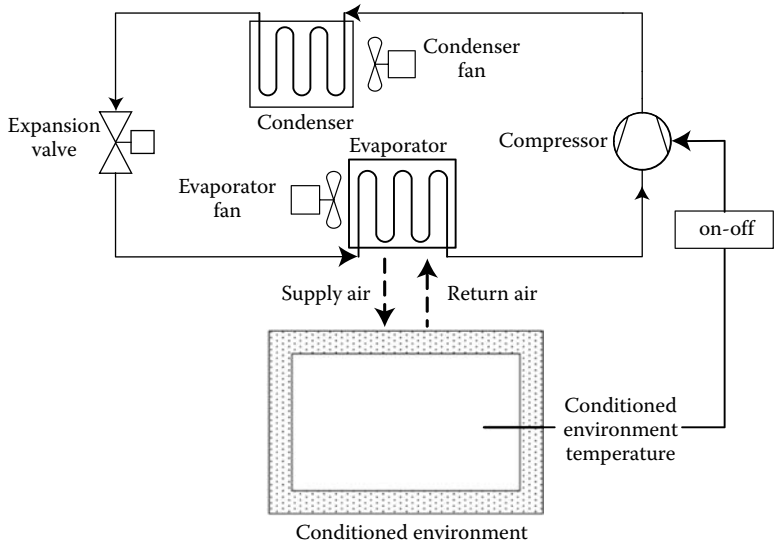

FIGURE 34.10 Schematic representation of an AC&R system interacting with a conditioned environment.

(e.g., orifice, capillary tube, TXV) to control the amount of superheated vapor exiting the evaporator. Some amount of evaporator superheat is necessary to protect the compressor but too much leads to inefficient evaporator operation since the heat transfer is much greater with two-phase fluid than with vapor. The cooling capacity is modulated either by turning on or off a motor driving the compressor or by engaging/disengaging a clutch mechanism being driven by some prime mover as shown in Figure 34.2. For different systems, the fans may cycle with the compressor or may operate on a separate schedule. A schematic representation of such a system is given in Figure 34.10 with a block diagram schematic given in Figure 34.11.

The typical performance for such an approach is shown in Figure 34.12 whereby a hot truck environment, such as the one shown in Figure 34.2, is cooled to a prescribed setpoint. As can be seen in Figure 34.12, the temperature oscillates about a given setpoint after convergence from its initial condition. The size of the oscillation and the symmetry about the setpoint are functions of the hysteresis parameters. The benefit of this approach to capacity control is its simplicity and low cost. The on–off compressor function is much cheaper than the power electronics needed to drive a variable speed compressor or the mechanical system needed to drive a variable speed transmission off of a prime mover. Additionally, the fan speeds can be cued to the compressor; for example, via a set of belt drives for compact systems. Therefore, a simple temperature sensor and rule-based logic is sufficient to close the loop. Moreover, since the thermal time constants are usually long, the cycling of the compressor is filtered, much like the current pulse width modulation of a DC motor driver. The drawback to this approach is the inability to adapt to changing conditions, both for the enclosed space and for the ambient external conditions. Should tighter temperature control or higher efficiency be desired, it may be necessary to consider other control approaches.

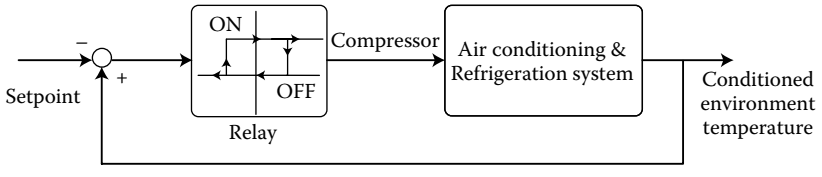

FIGURE 34.11 On–off hysteretic control system for AC&R compressor capacity control.

34.4.2 Variable Input Control: PID

The amount of energy consumed by an AC&R system varies considerably and depends on the desired temperature of the conditioned environment, the ambient conditions, and the level of internal heat generation within the conditioned environment. Capacity control methods, such as the on–off approach, allow these systems to meet varying cooling loads. A summary of various studies to determine the best capacity control method was conducted by [26] which found that variable speed compressor control provided the greatest flexibility to match heat loads, resulting in the best overall system efficiency. The variable speed compressor control strategies resulted in 20–40% reductions in seasonal power consumption, albeit with a significant increase in cost. In addition to variable speed compressors, it is possible to utilize variable orifice EEVs and even variable speed heat exchanger fans. All these increase the cost and complexity of both the physical system and the control system but offer the potential for improved performance in terms of temperature regulation and energy consumption. The simplest approach to control an AC&R system with variable inputs, and one that is often used for these system, is to utilize individual Proportional-Integral-Derivative (PID) loops for particular input–output pairs. Figure 34.13 illustrates valve control of superheat by estimating the refrigerant temperature and pressure and compressor control of capacity by using the evaporator air inlet temperature as a feedback variable. There are several alternative feedback variables that can be used in addition to those shown here. For example, Refs. [9,27] utilize additional valve control to capture refrigerant in the receiver and isolate it from the rest of the loop. Consequently, the results in [9,27] used refrigerant charge as an input variable to control subcooling in the condenser. Other approaches include control of frost buildup on the evaporator coil. It is possible to extend the operating range of SISO approaches by utilizing more advanced loop shaping or optimal H-infinity techniques. However, the dynamics of the SISO control loop are usually of sufficiently low order that PID is sufficient.

The drawback to closing individual loops is that the coupling among the different input–output pairs can lead to controller fighting [28,29]. For example, both the valve and the compressor are regulating mass flow throughout the refrigerant circuit and are therefore coupled. Figure 34.14 illustrates the coupled nature for a particular automotive system model; tight regulation of the valve-controlled evaporator superheat loop acts as a major disturbance to the compressor controlled evaporator pressure (i.e., capacity) loop. One alternative, to be described later, is to use multivariable control to compensate for the system coupling. However, this would limit the accessibility of the control approach to the majority of AC&R design and calibration engineers. The PID approach is one that is well understood and accepted by a broad range of users. A more favorable approach would be to utilize decoupling techniques for the different feedback loops and then apply PID control to the decoupled system. There are several different decoupling techniques ranging from static decoupling of a given input–output representation [30] to the

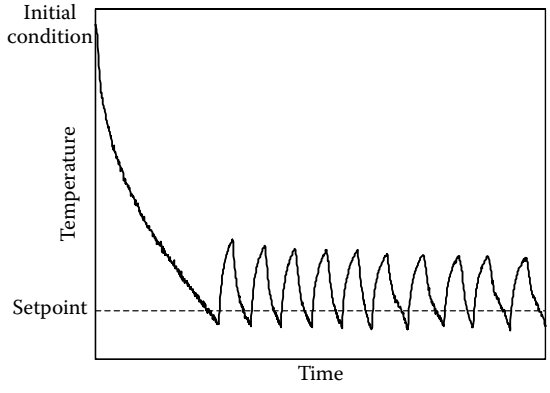

FIGURE 34.12 On–off hysteretic control system performance.

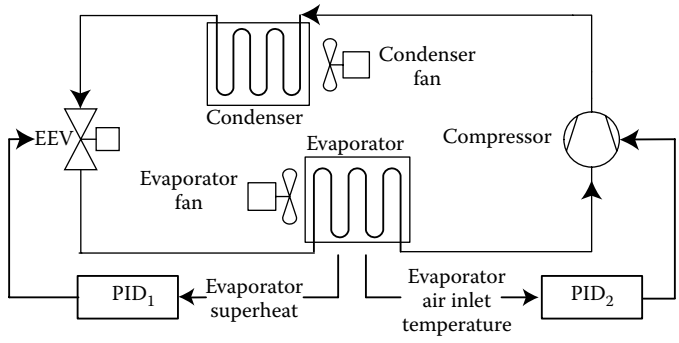

FIGURE 34.13 Individual PID control loops for an AC&R system.

reconfiguration of the system by redefining control objectives so as to realize a more naturally decoupled system [31] suitable for simple PID control.

34.4.3 Gain Scheduling

As illustrated in Figure 34.9, AC&R systems vary significantly under different operating conditions. Oftentimes, this plant variation is not explicitly compensated for and the controller is conservatively tuned to do the best it can. This is particularly true for the hysteretic on–off control approach given above. One approach to compensate for the plant variation is to schedule the controller as a function of operating conditions [32]. There are different methods of scheduling but, if one assumes the most common structure of a PID controller, a straightforward approach is to directly interpolate PID gains as a function of the scheduling variable. Figure 34.15 shows a candidate interpolation strategy based on Takagi–Sugeno models [33] for three different nominal controllers where the scheduling variable is the evaporator inlet air temperature. This variable would be a suitable indicator of the conditioned environment temperature for the system. Each control gain (K_p, K_i, K_d) is multiplied by the appropriate weighting or control

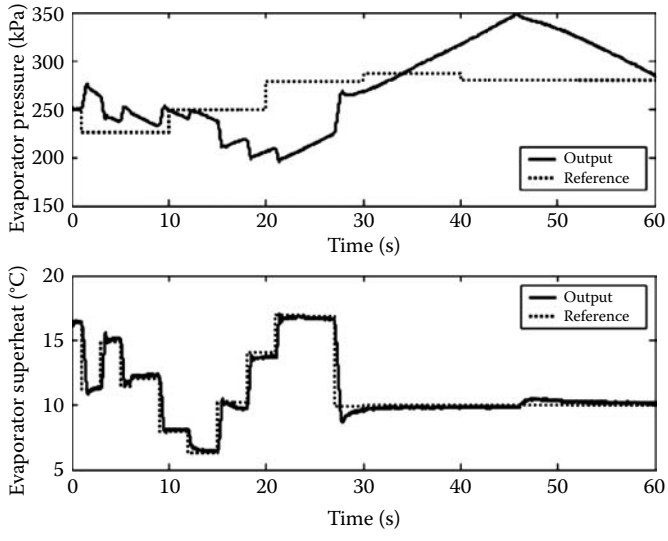

FIGURE 34.14 Dual SISO loop proportional-integral control of an automotive AC&R system [28].

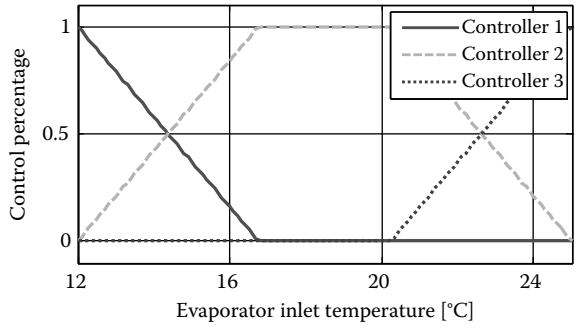

FIGURE 34.15 Scheduling algorithm for multiple local controllers.

percentage. Stability results for a class of this type of system used in AC&R applications can be found in [34].

Figures 34.16 and 34.17 show evaporator superheat regulation for a system with an EEV and constant compressor/fan speeds. The scheduling of PID gains with evaporator inlet air temperature provides a much more consistent system response to a change in the desired setpoint. The superheat setpoint value chosen was 9°C for this particular experimental system. Typical superheat setpoints for automotive AC&R systems are in the range of 3–5°C, while many home and commercial systems can be as high as 10°C. A similar scheduling approach could be used for the hysteretic capacity control approach above whereby the relay parameters can be interpolated as a function of operating condition. Comparing Figures 34.16 and 34.17, the scheduled PID controller provides a more uniform superheat control performance than a fixed PID controller over a range of operating conditions.

In closing discussions of basic control approaches, it is useful to mention that typical augmentations to standard SISO feedback control approaches include the use of feedforward controllers; for example, to minimize the effect of one control input on another. For electronic systems, it may be possible to feed the compressor speed reference commands to either the fan or the valve controllers that may utilize them in an anticipatory sense.

34.5 Advanced Control Design

The next level of control sophistication is the utilization of multi-input, multi-output (MIMO) approaches aimed at coordinating multiple outputs and actuators. This can be vital if there is very tightly coupled dynamic behavior between the different input–output variables. Additionally, it becomes more important

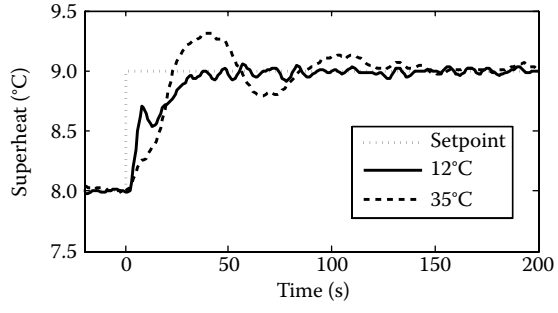

FIGURE 34.16 Fixed PID superheat regulation with varying evaporator air inlet temp.

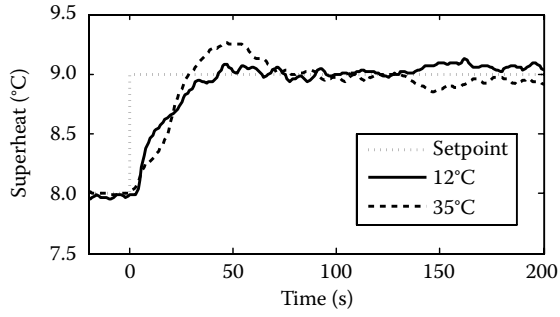

FIGURE 34.17 Gain-scheduled PID superheat regulation with varying evaporator air inlet temp.

for complicated systems where there may be multiple sets of heat exchangers, valves, and compressor racks as may be found in large distributed systems such as supermarket refrigeration display cases [35]. There have been previous MIMO control efforts and, with the move toward electrification of AC&R systems, it is promising that MIMO control approaches will become prevalent. Some of the earliest MIMO work was that of [36] which examined a linear quadratic regulator (LQR) and linear quadratic gaussian (LQG) approach to coordinating inputs affecting mass flow such as compressor speed and valve opening. The control performance was significantly superior to SISO control system performance in terms of response speed and disturbance minimization. Also, the utilization of model predictive control (MPC) has emerged [37,38] as a means to satisfy input and state constraints on the system while still maintaining good output tracking performance. The use of linear optimal methods such as LQR and LQG with Loop Transfer Recovery are very suitable for AC&R systems, provided that the system maintains its operation within a prescribed range of operating conditions where the assumption of linearity is valid. Excursions outside the range of validity would necessitate the type of scheduling arrangements mentioned in Section 34.4. For MPC schemes, it is possible to implicitly schedule by continuously adjusting the plant model as the control horizon recedes.

Within the context of AC&R research, several advances beyond PID control loops have focused on optimal setpoint generation for minimizing some cost function. Figure 34.18 illustrates an outer loop optimization routine that would feed setpoints to an inner loop feedback controller. The outer optimization loop would monitor the overall system, including the conditioned environment and the ambient environment, and then determine the appropriate cooling capacity needed. This would then be fed to the inner closed-loop controller that would regulate to these setpoints. In addition to capacity, some of the

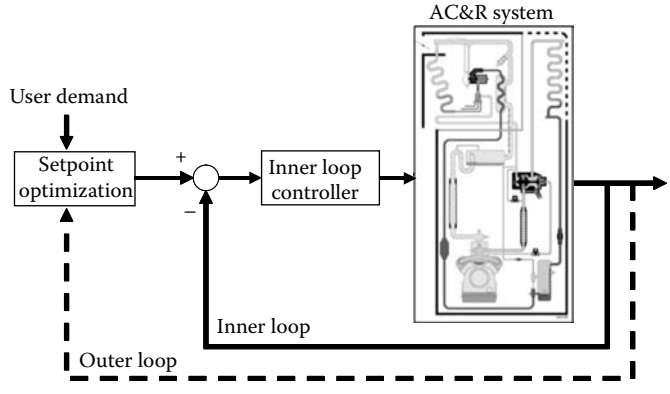

FIGURE 34.18 Inner and outer loop approach to AC&R control design.

elements of a cost function include the power consumed by the components, particularly compressors and fans. The cost functions can also contain temperature deviation errors and even direct estimates for the state of items being refrigerated [35]. The optimization can be performed offline and stored in tabular form and then retrieved during operation. Online optimization approaches, in a receding-horizon MPC sense, have shown some promise and are likely to gain acceptance with time.

34.6 Concluding Remarks

The field of AC&R is a fertile one for controls. There is a clear impact that can be made on the operation and efficiency of these devices through the use of modern control design tools. Throughout the 1990s and 2000s the industry evolved from a purely mechanical control approach using clutches and pressure-balanced TXV's to a more electronic approach with sensors and embedded systems. An analogy could be made between the internal combustion engine's conversion from mechanical governance (e.g., carburetors) in the 1970s to networked integrated powertrain management systems at present. If the analogy holds, we can expect future AC&R systems to provide much more accurate temperature control while significantly cutting energy use and also improving machine reliability.

The systems and descriptions given here should be considered a starting point for understanding the dynamics and control of AC&R systems. There are significant complicating factors and configurations for different types of systems. For example, the vapor compression cycle system can be run in reverse to act as a heat pump which can be desirable when trying to remove the frost that would build up on an evaporator coil. Moreover, as shown in Figure 34.3, there can be additional subsystems that are present on a real system that would complicate the implementation of controllers designed for an idealized system. However, the basic feedback concepts described in this chapter for idealized systems would hold for the more complex physical systems.

There remain some key barriers to achieving advanced system control. A primary one is the system nonlinearity coupled with the absence of a well-parameterized nonlinear system model in a convenient form such as LPV. Related to this is the need for further modeling work that is explicitly focused on modeling for control design. Finally, the cost of the overall electrification of the system should be sufficiently reduced such that it becomes economically feasible, given the cost of available energy, to implement advanced system hardware.

Nomenclature

Variable	Description
α	heat transfer coefficient
Δ	change
$\bar{\gamma}$	mean void fraction
η	efficiency
ρ	density
τ	time constant
ω	rotational speed
A	area
C_d	flow coefficient
C_p	specific heat
G	transfer function
h	specific enthalpy
K	control gain

L	length
$\dot{m}$	mass flow rate
N	finite volume regions
P	pressure
p	perimeter, pole location
Q	heat
RPM	revolutions per minute
s	entropy, Laplace variable
t	time
T	temperature
u	input
V	volume
W	work
x	dynamic state
z	spatial coordinate
Z	descriptor form matrix

Subscript	Description
1,2,3,4	1st, 2nd, 3rd, 4th transition point
1,2	1st, 2nd region
a	air
cs	cross-sectional
d	derivative
e	evaporator
f	liquid
g	vapor
i	inner, integral
in	inlet
k	compressor
o	outer
out	outlet
p	proportional
r	refrigerant
sat	saturation
sh	superheat
TXV	thermostatic expansion valve
v	valve
vol	volumetric
w	wall

References

1. Energy Information Administration, *Annual Energy Review* 2006, Washington, DC, June 2007. Also, http://www.eia.doe.gov.
2. DOE Report No. DOE/EIA-0573, 2007. Also, http://www.eia.doe.gov/oiaf/1605/ggrpt.
3. Koomey, J. and Brown, R.E., The role of building technologies in reducing and controlling peak electricity demand, LBNL Technical Report 49947, September 2002.
4. Buildings Energy Data Book 2008. Also, http://buildingsdatabook.eere.energy.gov.
5. Constable, G. and Somerville, R., *A Century of Innovation: Twenty Engineering Achievements that Transformed our Lives*, National Academies Press, Washington, DC, 2003.

6. Althouse, A.D., Turnquist, C.H., and Bracciano, A.F., *Modern Refrigeration and Air Conditioning*, Tinley Park, IL: The Goodheart-Willlcox Co., 1995.
7. Stoeker, W.F., *Industrial Refrigeration Handbook*, New York: McGraw-Hill, 1998.
8. Jensen, J. and Skogestad, S., Optimal operation of simple refrigeration cycles: Part I: Degrees of freedom and optimality of sub-cooling, *Computers & Chemical Engineering*, 31(5–6), 712–721, 2007.
9. Bendapudi, S., Braun, J.E., and Groll, E.A., A comparison of moving-boundary and finite-volume formulations for transients in centrifugal chillers, *International Journal of Refrigeration-Revue Internationale Du Froid*, 31(8), 1437–1452, 2008.
10. James, K.A. and James, R.W., Transient analysis of thermostatic expansion valves for refrigeration system evaporators using mathematical models, *Transactions of Institution of Measurement and Control*, 9(4), 198–205, 1987.
11. Lebrun, J. and Bourdouxhe, J.P., *Reference Guide for Dynamic Models of HVAC Equipment*, ASHRAE Project 738-TRP, Atlanta, GA, 1998.
12. Bendapudi, S. and Braun, J.E., A review of literature on dynamic models of vapor compression equipment, ASHRAE Report #4036–5, May 2002.
13. Incropera, F. and deWitt, D.P., *Introduction to Heat Transfer*, New York: John Wiley & Sons, 2002.
14. Anand, G., Mahajan, M., Jain, N., Maniam, B., and Tumas, T.M., e-thermal: Automobile air conditioning module, *Society of Automotive Engineers 2004 World Congress, SAE Paper 2004-01-1509*, Detroit, MI, 2004.
15. Eborn, J., Tummescheit, H., and Prolss, K., Air conditioning—a Modelica library for dynamic simulation of AC systems, *4th International Modelica Conference*, pp. 185–192, Hamburg-Harburg, Germany, March 7–8, 2005.
16. Cullimore, B.A. and Hendricks, T.J., Design and transient simulation of vehicle air conditioning systems, *Society of Automotive Engineers 5th Vehicle Thermal Management Systems Conference, Paper VTMS 5 2001-01-1692*, 2001.
17. Wedekind, G.L., Bhatt, B.L., and Beck, B.T., A system mean void fraction model for predicting various transient phenomena associated with two-phase evaporating and condensing flows, *International Journal of Multiphase Flow*, 4, 97–114, 1978.
18. He, X.D., Liu, S., and Asada, H., Modeling of vapor compression cycles for multivariable feedback control of HVAC systems, *ASME Journal of Dynamic Systems, Measurement and Control*, 119(2), 183–191, 1997.
19. Rasmussen, B.P. Dynamic modeling and advanced control of air conditioning and refrigeration systems, PhD. Thesis, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana-Champaign, IL, 2005.
20. Eldredge, B.D., Rasmussen, B.P., and Alleyne, A.G., Moving-boundary heat exchanger models with variable outlet phase, *ASME Journal of Dynamic Systems, Measurement, and Control*, 130(6), Article ID 061003, 2008.
21. Li, B. and Alleyne, A., A dynamic model of a vapor compression cycle with shut-down and start-up operations, *International Journal of Refrigeration*, 33(3), 538–552, May 2010.
22. Alleyne, A.G., Rasmussen, B.P., Keir, M.C., and Eldredge, B.D., Advances in energy systems modeling and control, *Proceedings of 2007 American Controls Conference*, pp. 4363–4373, New York, NY, July 2007.
23. Li, B. and Alleyne, A.G., A full dynamic model of a HVAC vapor compression cycle interacting with a dynamic environment, *Proceedings of 2009 American Controls Conference*, pp. 3662–3668, St. Louis, MO, June 2009.
24. Chen, H., Thomas, L., and Besant, R.W., Fan supplied heat exchanger fin performance under frosting conditions, *International Journal of Refrigeration*, 26(1), 140–149, 2003.
25. Rasmussen, B., Musser, A., and Alleyne, A. Model-driven system identification of transcritical vapor compression systems, *IEEE Transactions on Control Systems Technology*, 13(3), 444–451, 2005.
26. Qureshi, T.Q. and Tassou, S.A., Variable-speed capacity control in refrigeration systems, *Applied Thermal Engineering*, 16(2), 103–113, 1996.
27. Jensen, J. and Skogestad, S., Optimal operation of simple refrigeration cycles: Part II: Degrees of freedom and optimality of sub-cooling, *Computers & Chemical Engineering*, 31(12), 1590–1601, 2007.
28. Shah, R., Rasmussen, B.P., and Alleyne, A.G., Application of a multivariable adaptive control strategy to automotive air conditioning systems, *International Journal of Adaptive Control and Signal Processing*, 18(2), 199–221, 2004.
29. Keir, M.C., Dynamic modeling, control, and fault detection in vapor compression systems, MS Thesis, Department of Mechanical and Industrial Engineering, University of Illinois, Urbana-Champaign, IL, 2006.

30. Astrom, K.J., Johansson, K.H., and Wang, Q., Design of decoupled PID controllers for MIMO systems, *Proceedings of the 2001 American Control Conference*, pp. 2015–2020, Arlington, VA, June 2001.
31. Jain N., Li, B., Keir, M., Hencey, B., and Alleyne A., Decentralized feedback structures of a vapor compression cycle system, *IEEE Transactions on Control Systems Technology*, 18(1), 185–193, January 2010.
32. Shamma, J.S. and Athans, M., Gain scheduling: Potential hazards and possible remedies, *IEEE Control Systems Magazine*, 12(3), 101–107, 1992.
33. Murray-Smith, R. and Johansen, T.A. (Eds), *Multiple Model Approaches to Modelling and Control*, Bristol, PA, Taylor & Francis, 1997.
34. Rasmussen, B.P. and Alleyne, A.G., Gain scheduled control of an air conditioning systems using the Youla parameterization, *Proceedings of the 2006 American Control Conference*, pp. 5336–5341, Minneapolis, MN, June 2006.
35. Cai, J., Jensen, J.B., Skogestad, S., and Stoustrup, J., On the trade-off between energy consumption and food quality loss in supermarket refrigeration systems, *Proceedings of the 2008 American Control Conference*, pp. 2880–2885, Seattle, WA, June 2008.
36. He, X.D., Asada, H.H., Liu, S., and Itoh, H., Multivariable control of vapor compression systems, *HVAC & R Research*, 4(3), 205–230, 1998.
37. Elliott, M.S. and Rasmussen, B.P., Model-based predictive control of a multi-evaporator vapor compression cooling cycle, *Proceedings of the 2008 American Control Conference*, pp. 1463–1468, Seattle, WA, 2008.
38. Larsen, L.F., Model based control of refrigeration systems, PhD Thesis, TU Aalborg, Denmark, 2006.