# *Feedback Control of Computing Systems*

**Joseph L. Hellerstein**
**Yixin Diao**
**Sujay Parekh**
**Dawn M. Tilbury**

IEEE
IEEE PRESS

WILEY-
INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

# *Feedback Control of Computing Systems*

# *Feedback Control of Computing Systems*

**Joseph L. Hellerstein**
**Yixin Diao**
**Sujay Parekh**
**Dawn M. Tilbury**

IEEE
IEEE PRESS

WILEY-
INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2004 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400, fax 978-646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

***Library of Congress Cataloging-in-Publication Data:***

Feedback control of computing systems / Joseph L. Hellerstein … [et al.].
p. cm.
"A Wiley-Interscience publication."
Includes bibliographical references and index.
ISBN 0-471-26637-X (cloth)
1. Feedback control systems. 2. Control theory. 3. Electronic data processing. I. Hellerstein, Joseph, 1952-

TJ216.F44 2004
629.8′3–dc22
2004040490

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

*To Our Families*

# Contents

## 9 PID Controllers 293

## 10 State-Space Feedback Control 337

# *Preface*

This book is intended primarily for practitioners engaged in the analysis and design of computing systems. Analysts and designers are extremely interested in the performance characteristics of computing systems, especially response times, throughputs, queue lengths, and utilizations. Although steady-state characteristics can be well understood using queueing theory (e.g., as is done with capacity planning), practitioners lack the conceptual tools to address the *dynamics of resource management*, especially changes in workloads and configuration. The focus of this book is to distill and make accessible the essentials of control theory needed by computing practitioners to address these dynamics.

The dynamics of computing systems are important considerations in ensuring the profitability and availability of many businesses. For example, e-commerce sites frequently contend with workloads that change so rapidly that service degradations and failures result. Experienced designers know that leaving the management of dynamics to operators is not acceptable because many changes occur too rapidly for humans to be able to respond in a timely manner. As a result, ad hoc automation is frequently deployed with surprising results, such as wild oscillations or very slow responses to changes in workloads. Our belief is that by understanding the essential elements of control theory, computing practitioners can design systems that adapt in a more reliable manner.

A second audience for this book comprises researchers in the fields of computer science and controls. Today, very few computer science researchers have familiarity with control theory. As a result, many resource management schemes fail to address concepts that are well understood in control, such as the effect of measurement and system delays on stability and other aspects of control performance. Similarly, researchers in control fields rarely appreciate the issues particular to computing systems, such as considerations for policy-based management, service-level agreements, and the implications of modifying computing systems to provide sensors and actuators that are appropriate for control purposes. To address this second audience, we show through numerous examples how control theoretic techniques can be applied to computer systems, and describe the many challenges that remain.

Much effort has been devoted to making this book accessible to computer scientists. First, the examples focus on computer systems and their components, such as Web servers, caching, and load balancing. Second, our approach to modeling draws heavily on insights from queueing systems and their dynamics (as opposed to Newton's laws). Third, we focus almost entirely on discrete-time systems rather than continuous-time systems (as is traditional in controls books). There are two reasons for this: (1) performance measurements of computer systems are solicited on a sampled basis, which is best described by a discrete-time model; and (2) computer scientists are quite comfortable thinking in terms of difference equations, and much less comfortable thinking in terms of differential equations.

## Prerequisites

The book assumes background in series and their convergence, all of which is common in an undergraduate engineering and mathematics curriculum. Some prior exposure to Z-transforms (or Laplace transforms) is also of benefit, although not required. Also helpful is experience with developing statistical models, especially using linear regression.

Having appropriate software tools is immensely helpful in developing statistical models as well as for control analysis and design. Throughout the book, we use `MATLAB`®, a very powerful analysis environment that is arguably the standard for control engineers[1]. In Appendix E we provide an introduction to `MATLAB` (including the Control System Toolbox). However, access to `MATLAB` is not required for the vast majority of the book, only the optional section (indicated by *) at the end of each chapter.

## Outline of the Book

The book is divided into three parts. Part I, Background, consists of one chapter introducing the control problem and giving an overview of the area. Part II, Modeling, contains six chapters and covers modeling of dynamic systems in discrete time using difference equations, Z-transforms, block diagrams, transfer functions, and transient analysis. The focus is on single-input, single-output first- and second-order systems, although Chapter 7 is devoted to multiple-input, multiple-output (MIMO) systems. Part III, Control, has four chapters. In the first chapter we describe proportional control and pole placement design. In the next chapter we consider integral and differential control as well, including PID

---

[1] `MATLAB` is a registered trademark of The MathWorks, Inc.

(proportional–integral–differential) control tuning techniques. In the third chapter we address state-space feedback control, including the application of pole placement to MIMO systems and design using linear quadratic regulators. In the last chapter we discuss a variety of advanced topics, such as adaptive control, gain scheduling, minimum-variance control, and fuzzy control. In all three parts, examples are used extensively to illustrate the problems addressed, the techniques employed, and the value provided by the techniques.

Several appendixes are provided to make the book more useful as a reference and more self-contained. Appendix A summarizes the mathematical notation used, Appendix B lists key acronyms, and Appendix C contains key results developed in the book. Anothertwo appendixes contain supplemental material. Appendix D describes results from linear algebra that are used in Chapters 7 and 10. In Appendix E we provide an overview of the facilities in `MATLAB` for doing control analysis and design along with a brief `MATLAB` tutorial.

Considerable thought was given to the choice of examples. We sought examples that both aid in communicating key concepts and provide a basis for modeling systems encountered in practice (especially based on our experience at IBM and that of colleagues elsewhere in industry and academia). Our most basic example is a single-server queueing system with exponential interarrival and service times and a finite-size buffer ($M/M/1/K$), which provides a means to study the dynamics of admission control and proportional scheduling. The e-mail example based on the IBM Lotus® Domino™ Server[2] provides insight into challenges faced in system identification. The Apache HTTP Server[3] example serves as a vehicle for studying MIMO control. Additional examples include caching with differentiated service and load balancing.

## Roadmaps of the Book

The book may be approached in many ways depending on the interests of the reader. As depicted in Figure P.1, computer scientists interested in the basics of control theory should read Chapters 1 and 4 in detail. Chapters 2, 3, and 5 should be skimmed to gain insight into the nature of control system modeling, and Chapter 8 can be read in modest detail to understand the essence of control system design. Chapter 11 will also be of interest since it discusses other areas of control theory that are potentially applicable to computing systems.

Designers of computing systems who want to apply control theory in practice should proceed as shown in Figure P.2 by readying Chapters 1 through 6 and Chapters 8 and 9. State-space techniques, which are described in Chapters 7 and 10, should be approached only after there is a solid understanding of the other chapters. Considerable effort has been made to include worked examples that can be the basis for more extensive analysis and design studies. Also, all of these chapters include a section of extended examples that should stimulate ideas about the range of applications of control theory to computing systems.

---

[2] IBM Lotus Domino is a registered trademark of IBM Corporation.

[3] Apache is a trademark of The Apache Software Foundation and is used with permission.

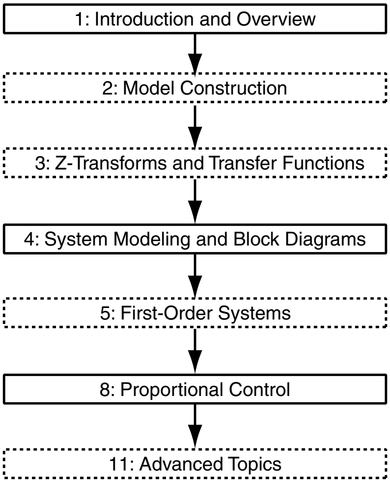

***Fig. P.1*** *Roadmap for computer scientists interested in the basics of control theory. Dashed boxes indicate chapters that should be skimmed.*

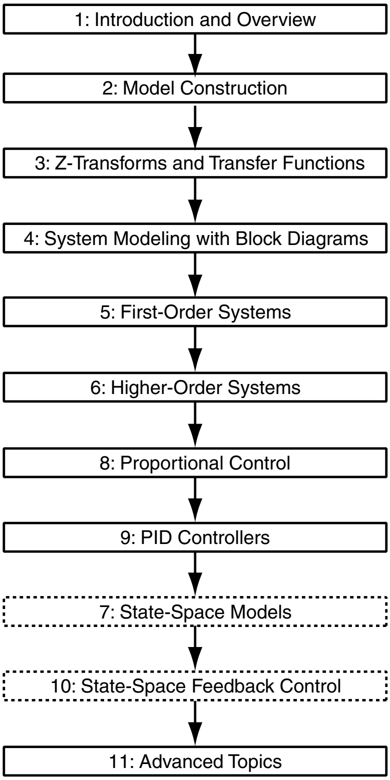

***Fig. P.2*** *Roadmap for computer scientists interested in applying control theory. Dashed boxes indicate chapters that should be skimmed.*

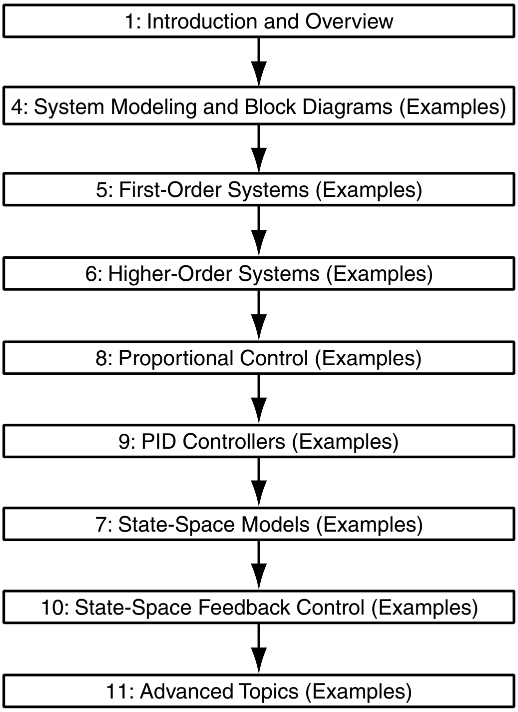

***Fig. P.3*** *Roadmap for control theorists interested in applications to computing systems. The focus should be on the examples, both the short in-chapter examples and the extended examples at the end of each chapter.*

Control theorists interested in computing system applications should proceed as depicted in Figure P.3. Desirable properties of controllers in computing systems and many examples of computing systems are described in Chapter 1. Chapters 4 through 11 contain a rich set of control problems based on these examples, especially the extended examples at the end of chapters.

## Errata and Additions

We intend to post errata and various additions to the book on the Web site *http://www.research.ibm.com/fbcs/*. For example, several of us are currently teaching a class based on the book at Columbia University. This has resulted in a number of new ideas about how to present the material.

## Acknowledgments

We wish to acknowledge the many colleagues who have helped with this book in various ways. Xichu Chen at the University of Michigan, Freeman Rawson at IBM Research in Austin, Texas, and Jose Renato Santos at Hewlett-Packard

Laboratory provided detailed comments on the text. David Patterson at the University of California–Berkeley and Armando Fox at Stanford aided us in better focusing the book for a computer science audience. Nagui Halim at IBM Research in Hawthorne, New York, gave strong support for this project from the start and provided constant enthusiasm throughout.

J.L. HELLERSTEIN
*IBM Thomas J. Watson Research Center*
*Hawthorne, New York*
Y. DIAO
*IBM Thomas J. Watson Research Center*
*Hawthorne, New York*
S. PAREKH
*IBM Thomas J. Watson Research Center*
*Hawthorne, New York*
D.M. TILBURY
*Mechanical Engineering Department*
*University of Michigan*
*Ann Arbor, Michigan*

# *Part I*

# *Background*

# 1

# Introduction and Overview

This book is about feedback control of computing systems. The main idea of feedback control is to use measurements of a system's outputs, such as response times, throughputs, and utilizations, to achieve externally specified goals. This is done by adjusting the system control inputs, such as parameters that affect buffer sizes, scheduling policies, and concurrency levels. Since the measured outputs are used to determine the control inputs, and the inputs then affect the outputs, the architecture is called *feedback* or *closed loop*. Almost any system that is considered automatic has some element of feedback control. In this book we focus on the closed-loop control of computing systems and methods for their analysis and design.

## 1.1 THE NATURE OF FEEDBACK CONTROL

Feedback control is about regulating the characteristics of a system. We begin with some key concepts: the measured output, which is the characteristic to be regulated to a desired value; the control input, which is what influences the measured output; and a disturbance, which affects the way in which the input affects the output. These are illustrated in a later section.

The reader may be familiar with everyday feedback control systems, such as cruise control in an automobile, a thermostat in a house, or the human sensorimotor system. A cruise control system achieves the desired speed by adjusting the accelerator pedal based on a velocity measurement from the speedometer. Here,

---

*Feedback Control of Computing Systems*, by Joseph L. Hellerstein, Yixin Diao, Sujay Parekh, and Dawn M. Tilbury
ISBN 0-471-26637-X

the accelerator pedal adjustments are the control input that provides a means to regulate speed, the measured output. The desired speed is maintained even when the car goes up or down hills or encounters head or tail winds, all of which are examples of disturbances that affect the relationship between the control input and the measured output. A thermostat achieves the desired temperature (output) by adjusting the furnace cycle and fan (input). The desired temperature is maintained even when the outside temperature increases or decreases (disturbance). The sensorimotor system achieves the desired hand position (output) to pick up an object by adjusting the muscle tensions (inputs) based on the current position sensed by the eyes and touch.

These concepts of feedback control apply to computing systems as well. Consider a computing system with a desired output characteristic. For example, operators of computing systems, or *administrators*, may want each of three Apache HTTP Servers [24] to run at no greater than 66% utilization, so that if any one of them fails, the other two can immediately absorb the entire load. Here, the measured output is CPU utilization. In computing systems, the measured output typically depends on the nature of the requests being served, or *workload*. Workload is often characterized in terms of the arrival process (e.g., Poisson, self-similar), and the distribution of service times for the resources used (e.g., CPU, memory, and database locks) [20]. In our studies of the Apache HTTP Server, CPU utilization depends on the workload and the control input. The workload is characterized by the request rate and whether the requests are for static or dynamic hypertext pages. The control input is the maximum number of connections that the server permits as specified by the `MaxClients` parameter. The workload is uncontrolled and so can be viewed as a disturbance. The control input `MaxClients` can be manipulated by an administrator or an automatic controller to affect CPU utilization.

Much of feedback control deals with understanding how the control input and disturbance affect the measured output. Continuing with the Apache HTTP Server example, as `MaxClients` increases, the CPU utilization increases. However, the effect is not instantaneous. A larger `MaxClients` only *allows* more users to connect; the system must wait some time for the users to arrive. Similarly, when `MaxClients` is decreased, current users are not disconnected until their sessions have timed out. Further, the value of `MaxClients` that results in a 66% utilization depends on the current workload, which may be unknown *a priori* and/or may change over time. Feedback control provides a method for setting `MaxClients` *automatically* to achieve the desired utilization that takes into account these dynamics and the effects of disturbances.

With this context we can describe feedback control in more detail. However, before doing so, a change in perspective is required. In computing systems we think in terms of the flow of work units or data through a system. Thus, input–output relationships reflect how work is done and/or data are transformed. Control theory also relies heavily on input–output relationships. However, the semantics are different. In control analysis, the inputs and outputs are metric values (e.g., CPU utilization) and/or control settings (e.g., `MaxClients`).

67. The Math Works. *MATLAB, The Language of Technical Computing*. The Math Works, Inc., Natick, MA, 1984–2003.
68. L. Zhang, Z. Zhao, Y. Shu, L. Wang, and O. Yang. Load balancing of multipath source routing in ad hoc networks. In *International Conference on Communications*, 2002.
69. R. E. Ziemer, W. H. Tranter, and D. R. Fannin. *Signals and Systems: Continuous and Discrete*, 4th ed. Prentice Hall, Upper Saddle River, NJ, 1998.

# Index

---

*Feedback Control of Computing Systems*, by Joseph L. Hellerstein, Yixin Diao, Sujay Parekh, and Dawn M. Tilbury
ISBN 0-471-26637-X