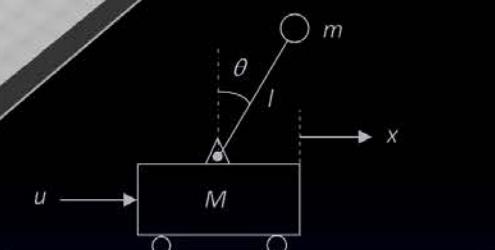
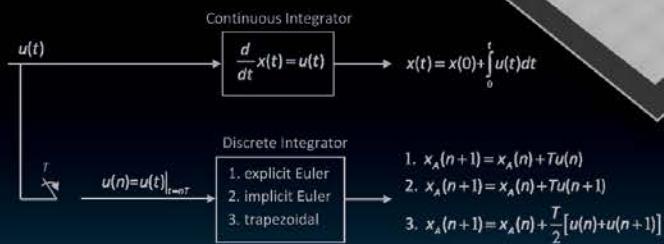
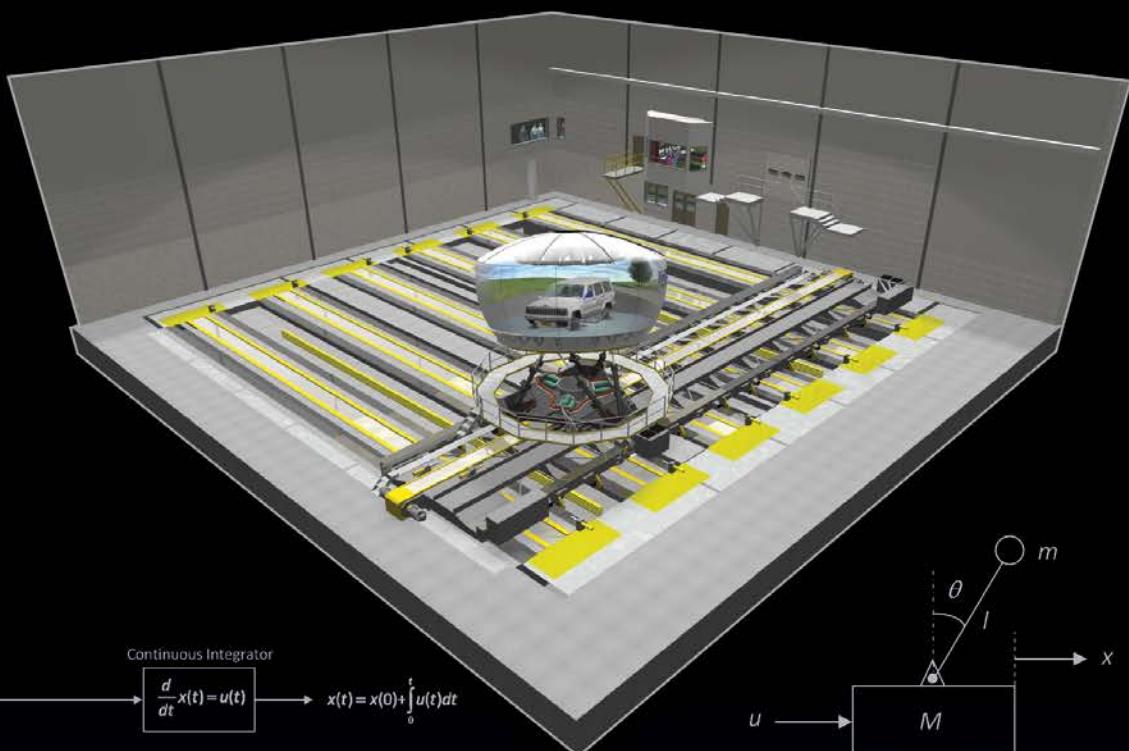


THIRD EDITION

Simulation of Dynamic Systems with MATLAB® and Simulink®

Harold Klee • Randal Allen



CRC Press
Taylor & Francis Group

Simulation of Dynamic Systems with MATLAB® and Simulink®

Third Edition



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

Simulation of Dynamic Systems with MATLAB® and Simulink®

Third Edition

Harold Klee and Randal Allen



CRC Press

Taylor & Francis Group

Boca Raton London New York

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

MATLAB® and Simulink® are a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® and Simulink® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® and Simulink® software.

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

© 2018 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper

International Standard Book Number-13: 978-1-4987-8777-2 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Klee, Harold, author. | Allen, Randal, 1964- author.
Title: Simulation of dynamic systems with MATLAB and Simulink / Harold Klee and Randal Allen.
Description: Third edition. | Boca Raton : Taylor & Francis, CRC Press, 2018.
| Includes bibliographical references and index.
Identifiers: LCCN 2017035511 | ISBN 9781498787772 (hardback) | ISBN 9781315154176 (ebook)
Subjects: LCSH: Computer simulation. | SIMULINK. | MATLAB.
Classification: LCC QA76.9.C65 K585 2018 | DDC 003/.3--dc23
LC record available at <https://lccn.loc.gov/2017035511>

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

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

*To Andrew, Cassie, and in loving memory of their
mother and my devoted wife, Laura.*

Harold Klee

*To my wife, Christine, whose inner beauty radiates with the warmth of a
sunny day at the beach. Thank you for your everlasting love and support.*

Randal Allen



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

Contents

Foreword	xiii
Preface.....	xv
About the Authors.....	xix
Chapter 1 Mathematical Modeling	1
1.1 Introduction	1
1.1.1 Importance of Models	1
1.2 Derivation of A Mathematical Model	4
1.3 Difference Equations.....	10
1.4 First Look at Discrete-Time Systems	19
1.4.1 Inherently Discrete-Time Systems	19
1.5 Case Study: Population Dynamics (Single Species).....	22
Chapter 2 Continuous-Time Systems.....	29
2.1 Introduction	29
2.2 First-Order Systems.....	29
2.2.1 Step Response of First-Order Systems	30
2.3 Second-Order Systems	36
2.3.1 Conversion of Two First-Order Equations to a Second-Order Model	41
2.4 Simulation Diagrams	45
2.4.1 Systems of Equations	51
2.5 Higher-Order Systems	54
2.6 State Variables.....	57
2.6.1 Conversion from Linear State Variable Form to Single Input–Single Output Form	62
2.6.2 General Solution of the State Equations	63
2.7 Nonlinear Systems.....	66
2.7.1 Friction	68
2.7.2 Dead Zone and Saturation.....	71
2.7.3 Backlash	72
2.7.4 Hysteresis	72
2.7.5 Quantization	76
2.7.6 Sustained Oscillations and Limit Cycles	77
2.8 Case Study: Submarine Depth Control System.....	85
Chapter 3 Elementary Numerical Integration.....	91
3.1 Introduction	91
3.2 Discrete-Time System Approximation of a Continuous First-Order System.....	92
3.3 Euler Integration	98
3.3.1 Explicit Euler Integration	99
3.3.2 Implicit Euler Integration	100
3.4 Trapezoidal Integration	104

3.5	Discrete Approximation of Nonlinear First-Order Systems	112
3.6	Discrete State Equations.....	116
3.7	Improvements to Euler Integration.....	127
3.7.1	Improved Euler Integration	127
3.7.2	Modified Euler Integration.....	131
3.7.3	Discrete-Time System Matrices	132
3.8	Case Study: Vertical Ascent of a Diver	146
Chapter 4	Linear Systems Analysis	155
4.1	Introduction	155
4.2	Laplace Transform.....	155
4.2.1	Properties of the Laplace Transform.....	156
4.2.2	Inverse Laplace Transform.....	163
4.2.3	Laplace Transform of the System Response	164
4.2.4	Partial Fraction Expansion	166
4.3	Transfer Function.....	173
4.3.1	Impulse Function.....	173
4.3.2	Relationship between Unit Step Function and Unit Impulse Function.....	173
4.3.3	Impulse Response.....	175
4.3.4	Relationship between Impulse Response and Transfer Function	179
4.3.5	Systems with Multiple Inputs and Outputs	182
4.3.6	Transformation from State Variable Model to Transfer Function....	190
4.4	Stability of Linear Time Invariant Continuous-Time Systems	194
4.4.1	Characteristic Polynomial	195
4.4.2	Feedback Control System.....	200
4.5	Frequency Response of LTI Continuous-Time Systems.....	206
4.5.1	Stability of Linear Feedback Control Systems Based on Frequency Response	216
4.6	z -Transform	222
4.6.1	Discrete-Time Impulse Function.....	226
4.6.2	Inverse z -Transform.....	232
4.6.3	Partial Fraction Expansion	233
4.7	z -Domain Transfer Function.....	242
4.7.1	Nonzero Initial Conditions	243
4.7.2	Approximating Continuous-Time System Transfer Functions.....	245
4.7.3	Simulation Diagrams and State Variables.....	250
4.7.4	Solution of Linear Discrete-Time State Equations.....	256
4.7.5	Weighting Sequence (Impulse Response Function).....	261
4.8	Stability of LTI Discrete-Time Systems	267
4.8.1	Complex Poles of $H(z)$	271
4.9	Frequency Response of Discrete-Time Systems.....	280
4.9.1	Steady-State Sinusoidal Response	280
4.9.2	Properties of the Discrete-Time Frequency Response Function	282
4.9.3	Sampling Theorem	287
4.9.4	Digital Filters	293
4.10	Control System Toolbox	300
4.10.1	Transfer Function Models	301
4.10.2	State-Space Models	302
4.10.3	State-Space/Transfer Function Conversion	303

4.10.4	System Interconnections	305
4.10.5	System Response	307
4.10.6	Continuous-/Discrete-Time System Conversion	309
4.10.7	Frequency Response	311
4.10.8	Root Locus	313
4.11	Case Study: Longitudinal Control of an Aircraft.....	319
4.11.1	Digital Simulation of Aircraft Longitudinal Dynamics.....	333
4.11.2	Simulation of State Variable Model	335
4.12	Case Study: Notch Filter for Electrocardiograph Waveform.....	338
4.12.1	Multinotch Filters	339
Chapter 5	Simulink®	349
5.1	Introduction	349
5.2	Building a Simulink Model	349
5.2.1	The Simulink Library.....	349
5.2.2	Running a Simulink Model	353
5.3	Simulation of Linear Systems	357
5.3.1	Transfer Fcn Block	357
5.3.2	State-Space Block.....	363
5.4	Algebraic Loops	371
5.4.1	Eliminating Algebraic Loops.....	373
5.4.2	Algebraic Equations	375
5.5	More Simulink Blocks.....	380
5.5.1	Discontinuities.....	385
5.5.2	Friction	386
5.5.3	Dead Zone and Saturation.....	387
5.5.4	Backlash	389
5.5.5	Hysteresis	389
5.5.6	Quantization	391
5.6	Subsystems	394
5.6.1	PHYSBE.....	395
5.6.2	Car-Following Subsystem	396
5.6.3	Subsystem Using Fcn Blocks	398
5.7	Discrete-Time Systems	402
5.7.1	Simulation of an Inherently Discrete-Time System	403
5.7.2	Discrete-Time Integrator	406
5.7.3	Centralized Integration.....	409
5.7.4	Digital Filters	412
5.7.5	Discrete-Time Transfer Function	414
5.8	MATLAB and Simulink Interface	422
5.9	Hybrid Systems: Continuous- and Discrete-Time Components	431
5.10	Monte Carlo Simulation	435
5.10.1	Monte Carlo Simulation Requiring Solution of a Mathematical Model.....	439
5.11	Case Study: Pilot Ejection	448
5.12	Case Study: Kalman Filtering	453
5.12.1	Continuous-Time Kalman Filter	453
5.12.2	Steady-State Kalman Filter	454
5.12.3	Discrete-Time Kalman Filter	454
5.12.4	Simulink Simulations	455

5.12.5	Summary	468
5.13	Case Study: Cascaded Tanks with Flow Logic Control.....	469
Chapter 6	Intermediate Numerical Integration.....	475
6.1	Introduction	475
6.2	Runge–Kutta (RK) (One-Step Methods).....	475
6.2.1	Taylor Series Method	476
6.2.2	Second-Order Runge–Kutta Method	477
6.2.3	Truncation Errors	479
6.2.4	High-Order Runge–Kutta Methods.....	484
6.2.5	Linear Systems: Approximate Solutions Using RK Integration	486
6.2.6	Continuous-Time Models with Polynomial Solutions.....	488
6.2.7	Higher-Order Systems	490
6.3	Adaptive Techniques	500
6.3.1	Repeated RK with Interval Halving.....	500
6.3.2	Constant Step Size ($T = 1$ min).....	505
6.3.3	Adaptive Step Size (Initial $T = 1$ min)	505
6.3.4	RK–Fehlberg	505
6.4	Multistep Methods.....	512
6.4.1	Explicit Methods	513
6.4.2	Implicit Methods	515
6.4.3	Predictor–Corrector Methods	518
6.5	Stiff Systems.....	523
6.5.1	Stiffness Property in First-Order System.....	524
6.5.2	Stiff Second-Order System	526
6.5.3	Approximating Stiff Systems with Lower-Order Nonstiff System Models	529
6.6	Lumped Parameter Approximation of Distributed Parameter Systems.....	546
6.6.1	Nonlinear Distributed Parameter System	550
6.7	Systems with Discontinuities.....	555
6.7.1	Physical Properties and Constant Forces Acting on the Pendulum Bob	563
6.8	Case Study: Spread of an Epidemic	573
Chapter 7	Simulation Tools.....	581
7.1	Introduction	581
7.2	Steady-State Solver.....	582
7.2.1	Trim Function.....	584
7.2.2	Equilibrium Point for a Nonautonomous System.....	586
7.3	Optimization of Simulink Models.....	596
7.3.1	Gradient Vector	605
7.3.2	Optimizing Multiparameter Objective Functions Requiring Simulink Models	607
7.3.3	Parameter Identification	610
7.3.4	Example of a Simple Gradient Search	611
7.3.5	Optimization of Simulink Discrete-Time System Models	620
7.4	Linearization	630
7.4.1	Deviation Variables	631
7.4.2	Linearization of Nonlinear Systems in State Variable Form	639

7.4.3	Linmod Function.....	643
7.4.4	Multiple Linearized Models for a Single System.....	648
7.5	Adding Blocks to The Simulink Library Browser	659
7.5.1	Introduction	659
7.5.2	Summary	665
7.6	Simulation Acceleration	665
7.6.1	Introduction	665
7.6.2	Profiler.....	667
7.6.3	Summary	668
7.7	Black Swans.....	668
7.7.1	Introduction	668
7.7.2	Modeling Rare Events	668
7.7.3	Measurement of Portfolio Risk	669
7.7.4	Exposing Black Swans	673
7.7.4.1	Percent Point Functions (PPFs).....	673
7.7.4.2	Stochastic Optimization	673
7.7.5	Summary	676
7.7.6	Acknowledgements	676
7.7.7	References	676
7.7.8	Appendix—Mathematical Properties of the Log-Stable Distribution.....	676
7.8	The SIPmath Standard	677
7.8.1	Introduction	677
7.8.2	Standard Specification.....	677
7.8.3	SIP Details.....	678
7.8.4	SLURP Details.....	678
7.8.5	SIPs/SLURPs and MATLAB.....	679
7.8.6	Summary	680
7.8.7	Appendix	681
7.8.8	References	682
Chapter 8	Advanced Numerical Integration	683
8.1	Introduction	683
8.2	Dynamic Errors (Characteristic Roots, Transfer Function)	683
8.2.1	Discrete-Time Systems and the Equivalent Continuous-Time Systems	684
8.2.2	Characteristic Root Errors.....	687
8.2.3	Transfer Function Errors	697
8.2.4	Asymptotic Formulas for Multistep Integration Methods	704
8.2.5	Simulation of Linear System with Transfer Function $H(s)$	708
8.3	Stability of Numerical Integrators.....	714
8.3.1	Adams–Bashforth Numerical Integrators	714
8.3.2	Implicit Integrators.....	722
8.3.3	Runga–Kutta (RK) Integration.....	726
8.4	Multirate Integration	738
8.4.1	Procedure for Updating Slow and Fast States: Master/Slave = RK-4/RK-4	742
8.4.2	Selection of Step Size Based on Stability	743
8.4.3	Selection of Step Size Based on Dynamic Accuracy	745
8.4.4	Analytical Solution for State Variables	748

8.4.5	Multirate Integration of Aircraft Pitch Control System.....	750
8.4.6	Nonlinear Dual Speed Second-Order System.....	753
8.4.7	Multirate Simulation of Two-Tank System	760
8.4.8	Simulation Trade-Offs with Multirate Integration.....	763
8.5	Real-Time Simulation.....	766
8.5.1	Numerical Integration Methods Compatible with Real-Time Operation	769
8.5.2	RK-1 (Explicit Euler).....	770
8.5.3	RK-2 (Improved Euler).....	771
8.5.4	RK-2 (Modified Euler)	771
8.5.5	RK-3 (Real-Time Incompatible)	771
8.5.6	RK-3 (Real-Time Compatible).....	772
8.5.7	RK-4 (Real-Time Incompatible).....	772
8.5.8	Multistep Integration Methods.....	772
8.5.9	Stability of Real-Time Predictor–Corrector Method	774
8.5.10	Extrapolation of Real-Time Inputs	776
8.5.11	Alternate Approach to Real-Time Compatibility: Input Delay	783
8.6	Additional Methods of Approximating Continuous-Time System Models....	790
8.6.1	Sampling and Signal Reconstruction	790
8.6.2	First-Order Hold Signal Reconstruction	796
8.6.3	Matched Pole-Zero Method.....	796
8.6.4	Bilinear Transform with Prewarping	799
8.7	Case Study: Lego Mindstorms™ NXT	803
8.7.1	Introduction	803
8.7.2	Requirements and Installation.....	805
8.7.3	Noisy Model	806
8.7.4	Filtered Model	810
8.7.5	Summary	815
References		817
Index		821

Foreword

MATLAB is used for so many applications, it defies attempts at categorization. This book demonstrates some of that interesting diversity.

As you read and use this book, you will find two kinds of knowledge. You may hope to find insight to the use of MATLAB and Simulink. That hope will be richly fulfilled, I think. But you should be mindful of another kind of knowledge; how others have solved problems. The rich collection of examples and methods go far beyond the software toolset. These span different technical disciplines and industries.

The authors show how modeling, simulation, and analysis gets done across a wide range of applications and industries, including financial markets. Work within and among various professional societies further broaden this perspective. Their university work in teaching budding scientists and engineers has honed the ability to make complexity approachable.

This book gives readers a chance to look outside their own discipline or industry, to collect ideas from afar.

I hope your imagination will be fired while your modeling and simulation skills are being honed.

Steve Roemeran
Chairman & CEO of Lone Star Analysis
Dallas, TX

Simulation has come a long way since the days analog computers filled entire rooms. Yet, it is more important than ever that simulations be constructed with care, knowledge, and a little wisdom, lest the results be gibberish or, worse, reasonable but misleading. Used properly, simulations can give us extraordinary insights into the processes and states of a physical system. Constructed with care, simulations can save time and money in today's competitive marketplace.

One major application of simulation is the simulator, which provides interaction between a model and a person through some interface. The earliest simulator, Ed Link's Pilot Maker aircraft trainer, did not use any of the simulation techniques described in this book. Modern simulators, however, such as the National Advanced Driving Simulator (NADS), cannot be fully understood without them.

The mission of the NADS is a lofty one: to save lives on U.S. highways through safety research using realistic human-in-the-loop simulation. This is an example of the importance simulation has attained in our generation. The pervasiveness of simulation tools in our society will only increase over time; it will be more important than ever that future scientists and engineers be familiar with their theory and application.

The content for Simulation of Dynamic Systems with MATLAB® and Simulink® is arranged to give the student a gradual and natural progression through the important topics in simulation. Advanced concepts are added only after complete examples have been constructed using fundamental methods. The use of MATLAB and Simulink provides experience with tools that are widely adopted in industry and allow easy construction of simulation models.

May your experience with simulation be enjoyable and fruitful and extend throughout your careers.

Chris Schwarz, PhD
Iowa City, Iowa



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

Preface

In the first article of *SIMULATION* magazine in the Fall of 1963, the editor John McLeod proclaimed simulation to mean “the act of representing some aspects of the real world by numbers or symbols which may be easily manipulated to facilitate their study.” Two years later, it was modified to “the development and use of models for the study of the dynamics of existing or hypothesized systems.” More than 40 years later, the simulation community has yet to converge upon a universally accepted definition. Either of the two cited definitions or others that followed convey a basic notion, namely, that simulation is intended to reinforce or supplement one’s understanding of a system. The definitions vary in their description of tools and methods to accomplish this.

The field of simulation is experiencing explosive growth in importance because of its ability to improve the way systems and people perform, in a safe and controllable environment, at a reduced cost. Understanding the behavior of complex systems with the latest technological innovations in fields such as transportation, communication, medicine, aerospace, meteorology, etc., is a daunting task. It requires an assimilation of the underlying natural laws and scientific principles that govern the individual subsystems and components. A multifaceted approach is required, one in which simulation can play a prominent role, both in validation of a system’s design and in training of personnel to become proficient in its operation.

Simulation is a subject that cuts across traditional academic disciplines. Airplane crews spend hours flying simulated missions in aircraft simulators to become proficient in the use of onboard subsystems during normal flight and possible emergency conditions. Astronauts spend years training in shuttle and orbiter simulators to prepare for future missions in space. Power plant and petrochemical process operators are exposed to simulation to obtain peak system performance. Economists resort to simulation models to predict economic conditions of municipalities and countries for policymakers. Simulations of natural disasters aid in preparation and planning to mitigate the possibility of catastrophic events.

While the mathematical models created by aircraft designers, nuclear engineers, and economists are application specific, many of the equations are analogous in form despite the markedly different phenomena described by each model. Simulation offers practitioners from each of these fields the tools to explore solutions of the models as an alternative to experimenting with the real system.

This book is meant to serve as an introduction to the fundamental concepts of continuous system simulation, a branch of simulation applied to dynamic systems whose signals change over a continuum of points in time or space. Our concern is with mathematical models of continuous-time systems (electric circuits, thermal processes, population dynamics, vehicle suspension, human physiology, etc.) and the discrete-time system models created to simulate them. The continuous system mathematical models consist of a combination of algebraic and ordinary differential equations. The discrete-time system models are a mix of algebraic and difference equations.

Systems that transition between states at randomly occurring times are called discrete-event systems. Discrete-event simulation is a complementary branch of simulation, separate from continuous system simulation, with a mathematical foundation rooted in probability theory. Examples of discrete-event systems are facilities such as a bank, a tollbooth, a supermarket, or a hospital emergency room, where customers arrive and are then serviced in some way. A manufacturing plant involving multiple production stages of uncertain duration to generate a finished product is another candidate for discrete-event simulation.

Discrete-event simulation is an important tool for optimizing the performance of systems that change internally at unpredictable times due to the influence of random events. Industrial engineering programs typically include a basic course at the undergraduate level in discrete-event simulation. Not surprisingly, a number of excellent textbooks in the area have emerged for use by the academic community and professionals.

In academia, continuous simulation has evolved differently than discrete-event simulation. Topics in continuous simulation such as dynamic system response, mathematical modeling, differential equations, difference equations, and numerical integration are dispersed over several courses from engineering, mathematics, and the natural sciences. In the past, the majority of courses in modeling and simulation of continuous systems were restricted to a specific field like mechanical, electrical, and chemical engineering or scientific areas like biology, ecology, and physics.

A transformation in simulation education is underway. More universities are beginning to offer undergraduate and beginning graduate courses in the area of continuous system simulation designed for an interdisciplinary audience. Several institutions now offer master's and PhD programs in simulation that include a number of courses in both continuous and discrete-event simulation. A critical mass of students are now enrolled in continuous simulation-related courses and there is a need for an introductory unifying text.

The essential ingredient needed to make simulation both interesting and challenging is the inclusion of real-world examples. Without models of real-world systems, a first class in simulation is little more than a sterile exposition of numerical integration applied to differential equations.

Modeling and simulation are inextricably related. While the thrust of this text is continuous simulation, mathematical models are the starting point in the evolution of simulation models. Analytical solutions of differential equation models are presented, when appropriate, as an alternative to simulation and a simple way of demonstrating the accuracy of a simulated solution. For the most part, derivations of the mathematical models are omitted and references to appropriate texts are included for those interested in learning more about the origin of the model's equations.

New and revised topics in the third edition are discussed in the later paragraphs dedicated to the content of each chapter. However, certain changes appearing in the third edition apply to the entire book, [Chapters 1 through 8](#). These changes consist of the following:

1. All MATLAB script and function .m files have been renamed and the references to them in the text have been changed to reflect the new file names. This eliminates the confusion present in the second edition which retained the MATLAB file names from the first edition based on the old system for naming chapter sections, figures, tables and exercise problems. Simulink model .mdl file names remain unchanged since they do not contain chapter or section references in their names.
2. With very few exceptions, nearly every graph generated in MATLAB has been redone to improve its appearance in printed form. Specifically, all line plots and markers are produced with a heavier weight, annotation and titles of most graphs have been changed to better communicate the significance of each graph. Whilst the graphs are in black and white in the text, every graph generated in MATLAB appears on screen in vivid colors to enhance their appearance. Updated MATLAB and Simulink files are accessible from CRC Press.
3. Simulink diagrams have been updated to be compatible with version R2016a of MATLAB/ Simulink. Diagrams with numerous Simulink blocks have been expanded to reveal the details of each block and their interconnections.
4. Certain non-graph figures have been eliminated as a result of being unnecessary, while others have been modified to be more informative.

Simulation is best learned by doing. Accordingly, the material is presented in a way that permits the reader to begin exploring simulation, starting with a mathematical model in [Chapter 1](#). The notation used to represent discrete-time variables has been simplified in the new edition making it easier to comprehend the difference equations developed to approximate the dynamics of continuous-time systems. The latter part of Section 1.1 and all of Sections 1.2 through 1.5 have been rewritten to better explain the underlying concepts.

[Chapters 2](#) and [4](#) remain basically unchanged. They present a condensed treatment of linear, continuous-time, and discrete-time dynamic systems, normally covered in an introductory linear systems course. The instructor can skip some or all of the material in these chapters if the students' background includes a course in signals and systems or linear control theory.

Numerical integration is at the very core of continuous system simulation. Instead of treating the subject in one exhaustive chapter, coverage is distributed over three chapters. Elementary numerical integration in [Chapter 3](#) is an informal introduction to the subject, which includes discussion of several elementary methods for approximating the solutions of first order differential equations. Presentation of the topics in [Chapter 3](#) has been completely revised. Much of the material in [Chapter 3](#) from the second edition appears in a reorganized format while some material has been deleted and new material added.

Simulink, from The MathWorks, is the featured simulation program because of its tight integration with MATLAB, the de facto standard for scientific and engineering analysis, and data visualization software. [Chapter 5](#) takes the reader through the basic steps of creating and running Simulink models. Monte Carlo simulation for estimation of system parameters and probability of events occurring in dynamic systems is covered. A new case study is introduced in Section 5.13 involving logically-controlled flows between two interconnected tanks.

[Chapter 6](#) delves into intermediate-level topics of numerical integration, including a formal presentation of One-Step (Runga–Kutta) and multistep methods, adaptive techniques, truncation errors and a brief mention of stability.

[Chapter 7](#) highlights some advanced features of Simulink useful in more in-depth simulation studies. Section 7.7 was added to demonstrate rare event modeling and portfolio risk measurement, thereby exposing potential Black Swans as they may pertain to the financial markets. Section 7.8 was added to introduce SIPmath as a means for efficiently representing uncertainty as probability distributions, enabling legacy and future simulation models to communicate with each other.

[Chapter 8](#) is for those interested in more advanced topics on continuous simulation. Coverage includes a discussion of dynamic errors, stability, real-time compatible numerical integration and multirate integration algorithms for simulation of stiff systems.

The basic minimum requirement for anyone using this text is a first course in Ordinary Differential Equations. An outline for a one-semester, preferably senior-level course in continuous system simulation is subject to the individual requirements of the instructor as well as the prior education of the students. As a starting point, some basic recommendations by the authors for a one-semester course are:

	Chapter 1 Sections	Chapter 2 Sections	Chapter 3 Sections	Chapter 4 Sections	Chapter 5 Sections	Chapter 6 Sections
For students knowledgeable in linear systems theory	1–4	1–8	1–8	Review of 1–8	1–3, 5, 6, 8, 9, 11 or 13	1–5, 8
For students not well-versed in linear systems theory	1–4	1–8	1–8	1–5	1–3, 5, 6, 8, 11 or 13	1–4, 8

All remaining sections are appropriate for a second course in a two-semester sequence, either at the senior, or more appropriately graduate level. The material in [Chapters 7](#) and [8](#) is well suited as a reference for practicing engineers and researchers involved in more advanced simulation endeavors.

The first and second editions of this text has been field-tested for nearly a decade. Despite numerous revisions based on the scrutiny and suggestions of students and colleagues, some errors manage to go undetected. Further suggestions for improvement and revelations of inaccuracies can be brought to the attention of the authors at aerospace321@outlook.com and klee.harold@gmail.com.

Numerous individuals deserve our thanks and appreciation for making the third edition possible. Thanks to Nora Konopka at Taylor & Francis/CRC Press for committing to the third edition

and Kyra Lindholm, also with Taylor & Francis/CRC Press, for facilitating the transition from the second to the third edition.

MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098 USA Tel: 508 647 7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com

About the Authors

Dr. Harold Klee received his PhD in systems science from Polytechnic Institute of Brooklyn in 1972, his MS in systems engineering from Case Institute of Technology in 1968, and his BSME from The Cooper Union in 1965.

Dr. Klee has been a faculty member in the College of Engineering at the University of Central Florida (UCF) since 1972. During his tenure at UCF, he has been a five-time recipient of the college's Outstanding Teacher Award. He has been instrumental in the development of simulation courses in both the undergraduate and graduate curricula. He is a charter member of the Core Faculty, which is responsible for developing the interdisciplinary MS and PhD programs in simulation at UCF. Dr. Klee served as graduate coordinator in the Department of Computer Engineering from 2003 to 2006. Two of his PhD students received the prestigious Link Foundation Fellowship in Advanced Simulation and Training. Both are currently enjoying successful careers in academia.

Dr. Klee has served as the director of the UCF Driving Simulation Lab for more than 15 years. Under the auspices of the UCF Center for Advanced Transportation Systems Simulation, the lab operates a high-fidelity motion-based driving simulator for conducting traffic engineering-related research. He also served as editor-in-chief for the *Modeling and Simulation* magazine for three years, a publication for members of the Society for Modeling and Simulation International.

Dr. Randal Allen is an aerospace and defense consultant working under contract to provide 6DOF aerodynamic simulation modeling, analysis, and design of navigation, guidance, and control systems. His previous experience includes launch systems integration and flight operations for West Coast Titan-IV missions, propulsion modeling for the Iridium satellite constellation, and field applications engineering for MATRIXx. He also chairs the Central Florida Section of the American Institute of Aeronautics and Astronautics (AIAA).

Dr. Allen is certified as a modeling and simulation professional (CMSP) by the Modeling and Simulation Professional Certification Commission (M&SPCC) under the auspices of the National Training and Simulation Association (NTSA). He is also certified to deliver FranklinCovey's Focus and Execution track, which provides training on achieving your highest priorities.

Dr. Allen's academic background includes a PhD in mechanical engineering from the University of Central Florida, an engineer's degree in aeronautical and astronautical engineering from Stanford University, an MS in applied mathematics, and a BS in engineering physics from the University of Illinois (Urbana–Champaign). He also serves as an adjunct professor at the University of Central Florida in Orlando, Florida.



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

1 Mathematical Modeling

1.1 INTRODUCTION

1.1.1 IMPORTANCE OF MODELS

Models are an essential component of simulation. Before a new prototype design for an automobile braking system or a multimillion dollar aircraft is tested in the field, it is commonplace to “test drive” the separate components and the overall system in a simulated environment based on some form of model. A meteorologist predicts the expected path of a tropical storm using weather models that incorporate the relevant climatic variables and their effect on the storm’s trajectory. An economist issues a quantitative forecast of the U.S. economy predicated based on key economic variables and their interrelationships with the help of computer models. Before a nuclear power plant operator is “turned loose” at the controls, extensive training is conducted in a model-based simulator where the individual becomes familiar with the plant’s dynamics under routine and emergency conditions. Health care professionals have access to a human patient simulator to receive training in the recognition and diagnosis of disease. Public safety organizations can plan for emergency evacuations of civilians from low-lying areas using traffic models to simulate vehicle movements along major access roads.

The word “model” is a generic term referring to a conceptual or physical entity that resembles, mimics, describes, predicts, or conveys information about the behavior of some process or system. The benefit of having a model is to be able to explore the intrinsic behavior of a system in an economical and safe manner. The physical system being modeled may be inaccessible or even nonexistent as in the case of a new design for an aircraft or automotive component.

Physical models are often scaled-down versions of a larger system of interconnected components as in the case of a model airplane. Aerodynamic properties of airframe and car body designs for high-performance airplanes and automobiles are evaluated using physical models in wind tunnels. In the past, model boards with roads, terrain, miniaturized models of buildings, and landscape, along with tiny cameras secured to the frame of ground vehicles or aircraft, were prevalent for simulator visualization. Current technology relies almost exclusively on computer-generated imagery.

In principle, the behavior of dynamic systems can be explained by mathematical equations and formulae, which embody either scientific principles or empirical observations, or both, related to the system. When the system parameters and variables change continuously over time or space, the models consist of coupled algebraic and differential equations. In some cases, lookup tables containing empirical data are employed to compute the parameters. Equations may be supplemented by mathematical inequalities, which constrain the variation of one or more dependent variables. The aggregation of equations and numerical data employed to describe the dynamic behavior of a system in quantitative terms is collectively referred to as a mathematical model of the system.

Partial differential equation models appear when a dependent variable is a function of two or more independent variables. For example, electrical parameters such as resistance and capacitance are distributed along the length of conductors carrying electrical signals (currents and voltages). These signals are attenuated over long distances of cabling. The voltage at some location x measured from an arbitrary reference is written $v(x, t)$ instead of simply $v(t)$, and the circuit is modeled accordingly.

A mathematical model for the temperature in a room would necessitate equations to predict $T(x, y, z, t)$ if a temperature probe placed at various points inside the room reveals significant variations in temperature with respect to x, y, z in addition to temporal variations. Partial differential

equations describing the cable voltage $v(x, t)$ and room temperature $T(x, y, z, t)$ are referred to as “distributed parameter” models.

The mathematical models of dynamic systems where the single independent variable is “time” comprise ordinary differential equations. The same applies to systems with a single spatial independent variable; however, these are not commonly referred to as dynamic systems since variations of the dependent variables are spatial as opposed to temporal in nature. Ordinary differential equation models of dynamic systems are called “lumped parameter” models because the spatial variation of the system parameters is negligible or else it is being approximated by lumped sections with constant parameter values. In the room temperature example, if the entire contents of the room can be represented by a single or lumped thermal capacitance, then a single temperature $T(t)$ is sufficient to describe the room. We focus exclusively on dynamic systems with lumped parameter models, hereafter referred to simply as mathematical models.

A system with a lumped parameter model is illustrated in [Figure 1.1](#). The key elements are the system inputs $u_1(t), u_2(t), \dots, u_r(t)$, which make up the system input vector $\underline{u}(t)$, the system outputs $y_1(t), y_2(t), \dots, y_p(t)$, which form the output vector $\underline{y}(t)$, and the parameters p_1, p_2, \dots, p_m constituting the parameter vector \underline{p} . The parameters are shown as constants; however, they may also vary with time.

Our interest is in mathematical models of systems consisting of coupled algebraic and differential equations relating the outputs and inputs with coefficients expressed in terms of the system parameters. For steady-state analyses, transient responses are irrelevant, and the mathematical models consist of purely algebraic equations relating the system variables.

An example of a mathematical model for a system with two inputs, three outputs, and several parameters is

$$p_1 \frac{d^2}{dt^2} y_1(t) + p_2 p_3 \frac{d}{dt} y_1(t) + p_4 y_1(t) + p_5 \frac{d}{dt} y_2(t) + p_6 y_2(t) = p_7 u_1(t) \quad (1.1)$$

$$p_8 \frac{d}{dt} y_2(t) + \frac{p_9}{p_{10}} y_2(t) + p_{11} y_1(t) y_2(t) = p_{12} \frac{d}{dt} u_1(t) + p_{13} u_1(t) + p_{14} u_2(t) \quad (1.2)$$

$$p_{15} y_3(t) = \frac{p_{16} y_1^{p_{17}}(t)}{y_2(t)} \quad (1.3)$$

The order of a model is equal to the sum of the highest derivatives of each of the dependent variables, in this case $y_1(t), y_2(t), y_3(t)$, and the order is therefore $2 + 1 + 0 = 3$. Equation 1.1 is a linear differential equation. Equation 1.2 is a nonlinear differential equation because of the term involving the product of $y_1(t)$ and $y_2(t)$. The mathematical model is nonlinear due to the presence of the nonlinear differential equation and the nonlinear algebraic equation (Equation 1.3). It is to be borne in mind that it is the nature of the equations that determines whether a math model is linear

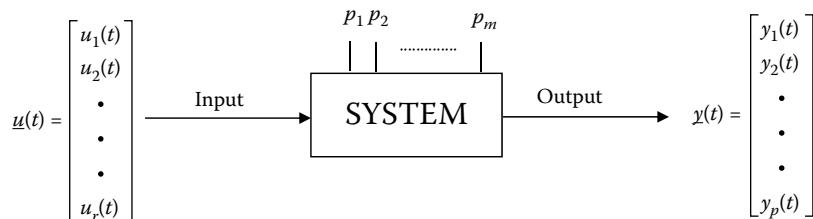


FIGURE 1.1 A system with a lumped parameter model.

or nonlinear. An adjective such as linear or nonlinear applies to the mathematical model as opposed to the actual system.

It is important to distinguish between the system being modeled and the model itself. The former is unique, even though it may exist only at the design stage, while the mathematical model may assume different forms. For example, a team of modelers may be convinced that the lead term in Equation 1.1 is likely to be insignificant under normal operating conditions. Consequently, two distinct models of the system exist, one third order and the other second order. The third-order model includes the second derivative term to accurately reflect system behavior under unusual or nontypical conditions (e.g., an aircraft exceeding its flight envelope or a ground vehicle performing an extreme maneuver). The simpler second-order model ignores what are commonly referred to as higher-order effects. Indeed, there may be a multitude of mathematical models to represent the same system under different sets of restricted operating conditions. Regardless of the detail inherent in a mathematical model, it nevertheless represents an incomplete and inexact depiction of the system.

A model's intended use will normally dictate its level of complexity. For example, models for predicting vehicle handling and responsiveness are different from those intended to predict ride comfort. In the first case, accurate equations describing lateral and longitudinal tire forces are paramount in importance, whereas passenger comfort relies more on vertical tire forces and suspension system characteristics.

Mathematical modeling is an inexact science, relying on a combination of intuition, experience, empiricism, and the application of scientific laws of nature. Trade-offs between model complexity and usefulness are routine. Highly accurate microclimatic weather models that use current atmospheric conditions to predict the following day's weather are of limited value if they require 48 h on a massively parallel or supercomputer system to produce results. At the extreme opposite, overly simplified models can be grossly inaccurate if significant effects are overlooked.

The difference between a mathematical model and a simulation model is open to interpretation. Some in the simulation community view the two as one and the same. Their belief is that a mathematical model embodies the attributes of the actual system and simulation refers to solutions of the model equations, albeit generally approximate in nature. Exact analytical solutions of mathematical model equations are nonexistent in all but the simplest cases.

Others maintain a distinction between the two and express the view that simulation model(s) originate from the mathematical model. According to this line of thinking, simulating the dynamics of a system requires a simulation model that is different in nature from a mathematical model. A reliable simulation model must be capable of producing numerical solutions in reasonably close agreement with the actual (unknown) solutions to the math model. Simulation models are commonly obtained from discrete-time approximations of continuous-time mathematical models. Much of this book is devoted to the process of obtaining simulation models in this way. More than one simulation model can be developed from a single mathematical model of a system.

Stochastic models are important when dealing with systems whose inputs and parameters are best modeled using statistical methods. Discrete event models are used to describe processes that transit from one state to another at randomly spaced points in time. Probability theory plays a significant role in the formulation of discrete event models for describing the movement of products and service times at different stages in manufacturing processes, queuing systems, and the like. In fact, the two pillars of simulation are continuous system simulation, the subject of this book, and discrete event simulation.

There is a great deal more to be said about modeling. Entire books are devoted to properly identifying model structure and parameter values for deterministic and stochastic systems. Others concentrate more on derivation of mathematical models from diverse fields and methods of obtaining solutions under different circumstances. The reader is encouraged to check the references section at the end of this book for additional sources of material related to modeling.

Modeling is essential to the field of simulation. Indeed, it is the starting point of any simulation study. The emphasis, however, in this book is on the presentation of simulation fundamentals.

Accordingly, derivation of mathematical models is not a prominent component. For the most part, the math models are taken from documented sources listed in the references section, some of which include step-by-step derivations of the model equations. The derivation is secondary to a complete understanding of the model, that is, its variables, parameters, and knowledge of conditions that may impose restrictions on its suitability for a specific application.

Simulation of complex systems requires a team effort. The modeler is a subject expert responsible for providing the math model and interpreting the simulation results. The simulationist produces the simulation model and performs the simulation study. For example, an aerodynamicist applies principles of boundary layer theory to obtain a mathematical model for the performance of a new airfoil design. Starting with the math model, simulation skills are required to produce a simulation model capable of verifying the efficacy of the design based on numerical results. Individuals with expert knowledge in a particular field are oftentimes well versed in the practice of simulation and may be responsible for formulation of alternative mathematical models of the system in addition to developing and running simulations.

A simple physical system is introduced in the next section, and the steps involved in deriving an idealized math model are presented. In addition to benefiting from seeing the process from start to finish, the ingredients for creating a simulation model are introduced. Hence, by the end of this chapter, the reader will be able to perform rudimentary simulation.

1.2 DERIVATION OF A MATHEMATICAL MODEL

We begin our discussion of mathematical modeling with a simple derivation of the mathematical model representing the dynamic behavior of an open tank containing a liquid that flows in the top and is discharged from the bottom. Referring to [Figure 1.2](#), the primary input is the liquid flow rate $F_1(t)$, an independent variable measured in appropriate units such as cubic feet per minute (volumetric flow rate) or pounds per hour (mass flow rate). Responding to changes in the input are dependent variables $H(t)$ and $F_0(t)$ the fluid level, and flow rate from the tank.

Once the derivation is completed, we can use the model to predict the outflow and fluid level response to a specific input flow rate $F_1(t)$, $t \geq 0$. Note that we have restricted the set of possible inputs to $F_1(t)$ and in the process relegated the remaining independent variables, that is, other variables which affect $F_0(t)$ and $H(t)$, to second-order importance. Our assumption is that the eventual



FIGURE 1.2 Tank as a dynamic system with input and outputs.

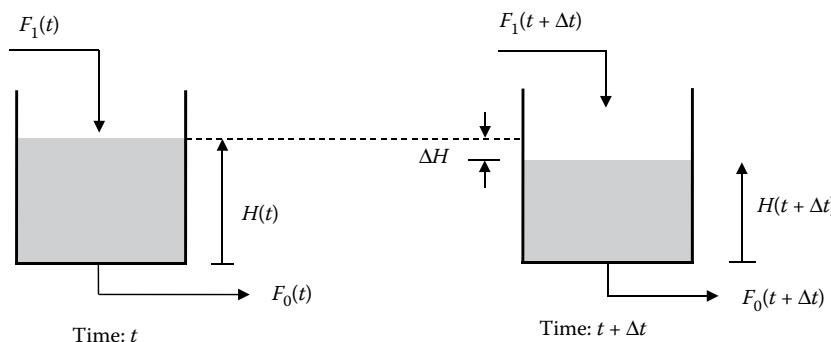


FIGURE 1.3 A liquid tank at two points in time.

model will be suitable for its intended application. It must be borne in mind that if extremely accurate predictions of the level $H(t)$ are required, it may be necessary to include second-order effects such as evaporation and hence introduce additional inputs related to ambient conditions, namely, temperature, humidity, air pressure, wind speed, and so forth.

The derivation is based on conditions of the tank at two discrete points in time, as if snapshots of the tank were available at times “ t ” and “ $t + \Delta t$,” as shown in [Figure 1.3](#).

The following notation is used with representative units given for clarity:

$F_1(t)$: Input flow at time t , ft³/min

$H(t)$: Liquid level at time t , ft

$F_0(t)$: Output flow at time t , ft³/min

A : Cross-sectional area of tank, ft²

At time $t + \Delta t$, from the physical law of conservation of volume,

$$V(t + \Delta t) = V(t) + \Delta V \quad (1.4)$$

where

$V(t)$ is the volume of liquid in the tank at time t

ΔV is the change in volume from time t to $t + \Delta t$

The volume of liquid in the tank at times t and $t + \Delta t$ is given by

$$V(t) = AH(t) \quad (1.5)$$

$$V(t + \Delta t) = AH(t + \Delta t) \quad (1.6)$$

Equations 1.5 and 1.6 assume constant cross-sectional area of the tank, that is, A is independent of H .

The change in volume from t to $t + \Delta t$ is equal to the volume of liquid flowing in during the interval t to $t + \Delta t$ minus the volume of liquid flowing out during the same period of time. The liquid volumes are the areas under the input and output volume flow rates from t to $t + \Delta t$ as shown in [Figure 1.4](#).

Expressing these areas in terms of integrals,

$$\Delta V = \int_t^{t+\Delta t} F_1(t)dt - \int_t^{t+\Delta t} F_0(t)dt \quad (1.7)$$

The integrals in Equation 1.7 can be approximated by assuming $F_1(t)$ and $F_0(t)$ are constant over the interval t to $t + \Delta t$ (see [Figure 1.4](#)). Hence,

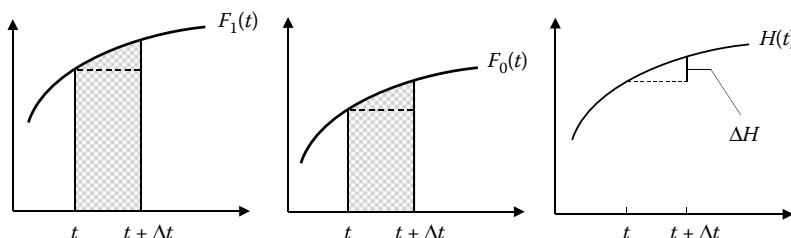


FIGURE 1.4 Volumes of liquid flowing in and out of tank from t to $t + \Delta t$.

$$\int_t^{t+\Delta t} F_l(t) dt \approx F_l(t) \Delta t \quad (1.8)$$

$$\int_t^{t+\Delta t} F_0(t) dt \approx F_0(t) \Delta t \quad (1.9)$$

Equations 1.8 and 1.9 are reasonable approximations provided Δt is small. Substituting Equations 1.8 and 1.9 into Equation 1.7 yields

$$\Delta V \approx F_l(t) \Delta t - F_0(t) \Delta t \quad (1.10)$$

Substituting Equations 1.5, 1.6, and 1.10 into Equation 1.4 gives

$$AH(t + \Delta t) \approx AH(t) + [F_l(t) - F_0(t)] \Delta t \quad (1.11)$$

$$\Rightarrow A[H(t + \Delta t) - H(t)] \approx [F_l(t) - F_0(t)] \Delta t \quad (1.12)$$

$$\Rightarrow A \left[\frac{\Delta H}{\Delta t} \right] \approx F_l(t) - F_0(t) \quad (1.13)$$

where ΔH is the change in liquid level over the interval $(t, t + \Delta t)$. Note that $\Delta H/\Delta t$ is the average rate of change in the level H over the interval $(t, t + \Delta t)$. It is the slope of the secant line from pt A to pt B in [Figure 1.5](#).

In the limit as Δt approaches zero, pt B approaches pt A, and the average rate of change in H over the interval $(t, t + \Delta t)$ becomes the instantaneous rate of change in H at time t , that is,

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta H}{\Delta t} = \frac{dH}{dt} \quad (1.14)$$

where dH/dt is the first derivative of $H(t)$. From the graph, it can be seen that dH/dt is equal to the slope of the tangent line of the function $H(t)$ at t (pt A).

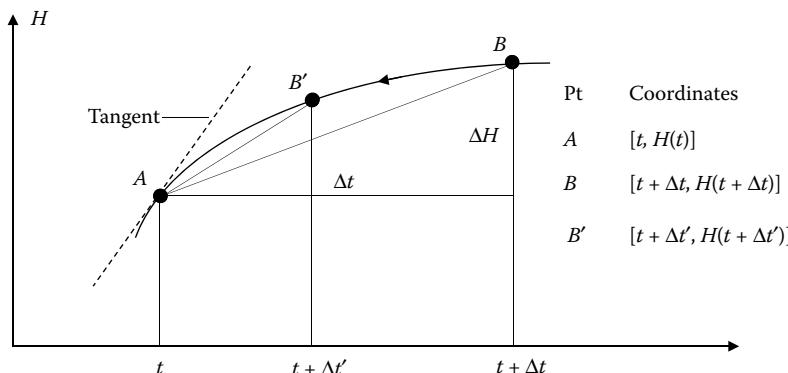


FIGURE 1.5 Average rate of change $\Delta H/\Delta t$ as Δt gets smaller.

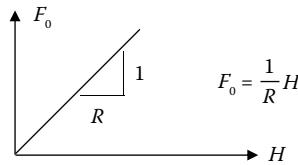


FIGURE 1.6 A tank with out-flow proportional to fluid level.

Taking the limit as Δt approaches zero in Equation 1.13 and using the definition of the derivative in Equation 1.14 give

$$\lim_{\Delta t \rightarrow 0} \left[\frac{\Delta H}{\Delta t} \right] = \lim_{\Delta t \rightarrow 0} [F_1(t) - F_0(t)] \quad (1.15)$$

$$\Rightarrow A \left[\frac{dH}{dt} \right] = F_1(t) - F_0(t) \quad (1.16)$$

Since there are two dependent variables, a second equation or constraint relating F_0 and H is required in order to solve for either one given the input function $F_1(t)$. It is convenient at this point to assume that F_0 is proportional to H , that is, $F_0 = \text{constant} \times H$ (see Figure 1.6). The constant of proportionality is expressed as $1/R$ where R is called the fluid resistance of the tank. At a later point, we will revisit this assumption.

$$F_0 = \frac{1}{R}H \quad (1.17)$$

Equations 1.16 and 1.17 constitute the mathematical model of the liquid tank, namely,

$$A \frac{dH}{dt} + F_0 = F_1 \quad \text{and} \quad F_0 = \frac{1}{R}H$$

where F_1 , F_0 , H , and dH/dt are short for $F_1(t)$, $F_0(t)$, $H(t)$, and $(d/dt)H(t)$.

In this example, the model is a coupled set of equations. One is a linear differential equation and the other is an algebraic equation, also linear. The differential equation is first order since only the first derivative appears in the equation and the tank dynamics are said to be first order.

The outflow F_0 can be eliminated from the model equations by substituting Equation 1.17 into Equation 1.16 resulting in

$$A \frac{dH}{dt} + \frac{1}{R}H = F_1 \quad (1.18)$$

Before a particular solution to Equation 1.18 for some $F_1(t)$, $t \geq 0$ can be obtained, the initial tank level $H(0)$ must be known.

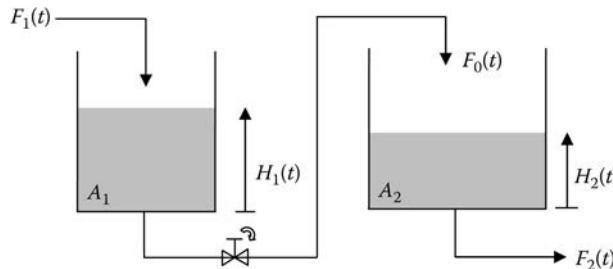
There are several reasons why an analytical approach to solving Equation 1.18 may not be the preferred method. Even when the analytical solution is readily obtainable, for example, when the system model is linear, as in the present example, the solution may be required for a number of different inputs or forcing functions. Recall from studying differential equations what happens when the right-hand side of the equation changes. A new particular solution is required that can be time-consuming, especially if the process is repeated for a number of nontrivial forcing functions.

Second, the input $F_1(t)$ may not even be available in analytical form. Suppose the input function $F_1(t)$ is unknown except as a sequence of measured values at regularly spaced points in time.

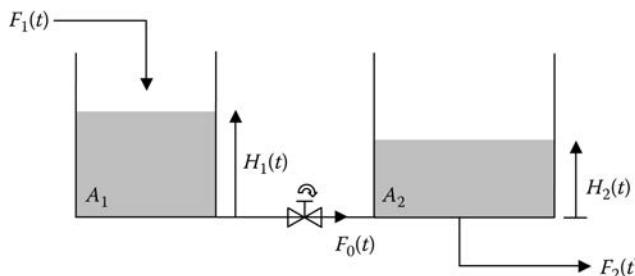
An exact solution to the differential equation model is out of the question since the input is not expressible as an analytic function of time.

EXERCISES

- 1.1 A system consists of two tanks in series in which the outflow from the first tank is the inflow to the second tank as shown in the figure below.



- Find the algebraic and differential equations comprising the mathematical model of the two tank system. Assume both tanks are linear, i.e. the outflows are proportional to the liquid levels and R_1, R_2 are the fluid resistances of the tanks.
 - Eliminate the flows $F_0(t)$ and $F_2(t)$ from the model to obtain a model in the form of two differential equations involving the system input $F_1(t)$ and the tank levels $H_1(t)$ and $H_2(t)$.
 - Obtain the model differential equations when $F_0(t)$ and $F_2(t)$ are present instead of $H_1(t)$ and $H_2(t)$.
 - The initial fluid levels in the tanks are $H_1(0)$ and $H_2(0)$. Suppose the flow in to the first tank is constant, $F_1(t) = \bar{F}_1$, $t \geq 0$. Obtain expressions for $H_1(\infty)$ and $H_2(\infty)$, the eventual fluid levels in Tanks 1 and 2. Do $H_1(\infty)$ and $H_2(\infty)$ depend on the initial fluid levels? Explain.
 - Find the ratio of tank resistances R_1/R_2 if $H_1(\infty) = 2H_2(\infty)$.
 - Suppose the flow between the two tanks is reduced to zero by closing the valve in the line. Show that this is equivalent to $R_1 = \infty$ and determine the values of $H_1(\infty)$ and $H_2(\infty)$ assuming the inflow to the first tank is still constant.
- 1.2 The two tanks in Exercise 1.1 are said to be non-interacting because the flow rate from the first tank only depends on the fluid level in the first tank and is independent of the fluid level in the second tank. Suppose the discharged fluid from the first tank enters the second tank at the bottom instead of the top as shown in the figure below.



The flow between the tanks is now a function of the fluid levels in both tanks. The driving force for the inter-tank flow is the difference in fluid levels and for the time being we can assume the two quantities are proportional. That is,

$$F_0(t) \propto [H_1(t) - H_2(t)] \Rightarrow F_0(t) = \frac{H_1(t) - H_2(t)}{R_{12}}$$

where R_{12} represents a fluid resistance involving both tanks. The fluid resistance of the second tank is still R_2 .

- a. The general form of the differential equation model for the system of interacting tanks is

$$\frac{dH_1}{dt} + a_{11}H_1 + a_{12}H_2 = b_1F_1$$

$$\frac{dH_2}{dt} + a_{21}H_1 + a_{22}H_2 = b_2F_1$$

Note: H_1 , H_2 and F_1 are short for $H_1(t)$, $H_2(t)$ and $F_1(t)$.

Find expressions for a_{11} , a_{12} , a_{21} , a_{22} , b_1 , b_2 in terms of the system parameters A_1 , A_2 , R_{12} , and R_2 .

- b. The tanks are initially empty, $H_1(0) = 0$ and $H_2(0) = 0$. The flow in to the first tank is constant, $F_1(t) = \bar{F}_1$, $t \geq 0$. Show that the final fluid levels in both tanks after a sufficient period of time has elapsed, $H_1(\infty)$ and $H_2(\infty)$, can be obtained from the solution of the following system of equations:

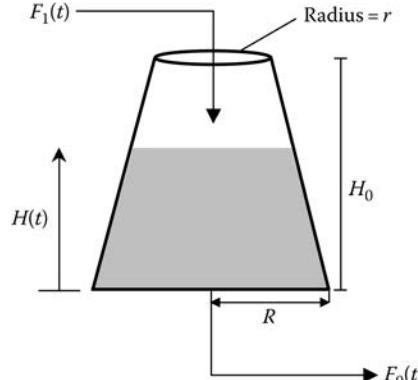
$$\begin{aligned} a_{11}H_1(\infty) + a_{12}H_2(\infty) &= b_1\bar{F}_1 \\ a_{21}H_1(\infty) + a_{22}H_2(\infty) &= b_2\bar{F}_1 \end{aligned}$$

- c. Solve for $H_1(\infty)$ and $H_2(\infty)$ in terms of the system parameters A_1 , A_2 , R_{12} , R_2 and the constant inflow \bar{F}_1 . Are the results different if the tanks are not initially empty? Explain.
- d. Using the following baseline values unless otherwise stated:

$$A_1 = A_2 = 25 \text{ ft}^2, \quad R_{12} = 3 \text{ ft per ft}^3/\text{min}, \quad R_2 = 1 \text{ ft per ft}^3/\text{min}, \quad \bar{F}_1 = 5 \text{ ft}^3/\text{min}$$

Find the eventual fluid levels $H_1(\infty)$ and $H_2(\infty)$ and flows $F_0(\infty)$ and $F_2(\infty)$.

- e. Repeat Part (d) with $A_2 = 75 \text{ ft}^2$.
- f. The valve between the tanks is opened some resulting in $R_{12} = 2 \text{ ft per ft}^3/\text{min}$. The remaining baseline values remain the same. Find $H_1(\infty)$, $H_2(\infty)$ and flows $F_0(\infty)$ and $F_2(\infty)$.
- g. Suppose Tank 1 initially holds 10 ft of liquid and Tank 2 has 4 ft. Find the initial rates of change in level for both tanks.
- h. Is it possible for the fluid level in Tank 2 to exceed the level in Tank 1? Explain.
- i. How does the model change if there is a separate flow, say $F_3(t)$ directly in to the top of Tank 2?
- 1.3 Consider a cone shaped tank with circular cross sectional area like the one shown in the figure below.



- a. How does this affect the derivation of the mathematical model?
- b. Find the math model for this case.

1.3 DIFFERENCE EQUATIONS

The tank model from the previous section, given in Equation 1.18, is a first-order differential equation. First-order systems, that is, systems governed by a first-order differential equation are treated in detail in [Chapter 2](#). Simulation of a first-order system requires finding an approximate solution to the differential equation.

A first-order system is shown in [Figure 1.7](#), where $u(t)$, $t \geq 0$ is the input; $y(t)$, $t \geq 0$ is the output; $y(0)$ is the initial condition, i.e. $y(t)$ at $t = 0$; $f[y(t), u(t)]$ is the mathematical function model of the dynamic system.

Examples of $f[y(t), u(t)]$ are:

- | | |
|--------------------------------------|--|
| 1. $f[y(t), u(t)] = ay(t)$ | linear first-order system with no input |
| 2. $f[y(t), u(t)] = ay(t) + bu(t)$ | linear first-order system with input $u(t)$ |
| 3. $f[y(t), u(t)] = ay^2(t) + bu(t)$ | nonlinear first-order system with input $u(t)$ |

Given the system model,

$$\frac{d}{dt} y(t) = f[y(t), u(t)] \quad (1.19)$$

An approximate solution for the response $y(t)$, $t \geq 0$, given the input $u(t)$, $t \geq 0$ and the initial condition $y(0)$, can be obtained at discrete points in time $t_n = nT$, $n = 0, 1, 2, \dots$. The discrete points are separated from each other by the step size T .

The approximate solution is $y_A(n)$, $n = 0, 1, 2, \dots$ where $y_A(n) \approx y(t_n) = y(nT)$, $n = 0, 1, 2, \dots$

[Figure 1.8](#) illustrates the difference between $y(t)$, $t \geq 0$ and $y_A(n)$, $n = 0, 1, 2, \dots$. It is useful for deriving an equation which can be solved to generate $y_A(n)$, $n = 0, 1, 2, \dots$.

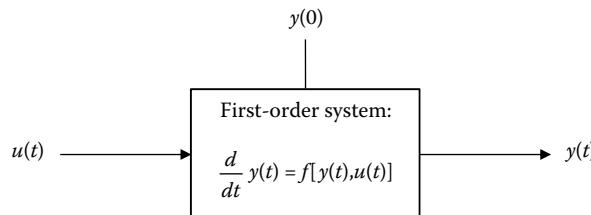


FIGURE 1.7 A first-order system with input $u(t)$ and output $y(t)$.

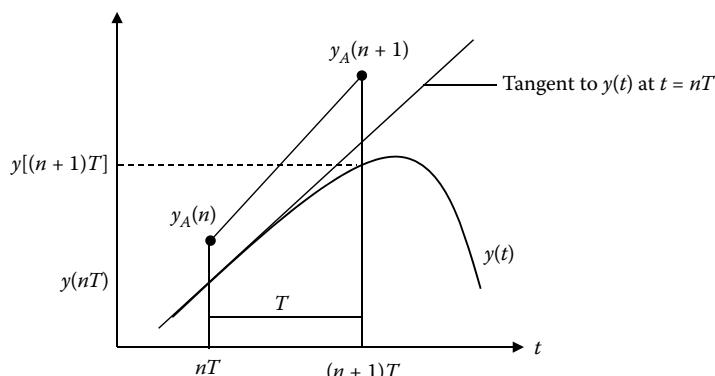


FIGURE 1.8 Illustration of the difference between $y(t)$ and $y_A(n)$.

The tangent to $y(t)$ at $t = nT$ is shown in [Figure 1.8](#). Its slope is numerically equal to the first derivative $\frac{dy}{dt}$ at $t = nT$. From Equation 1.19,

$$\text{At } t = nT, \quad \left. \frac{d}{dt} y(t) \right|_{t=nT} = f[y(nT), u(nT)] \quad (1.20)$$

Approximating $\left. \frac{d}{dt} y(t) \right|_{t=nT}$ by the slope of the line connecting $y_A(n)$ and $y_A(n + 1)$,

$$f[y(nT), u(nT)] \approx \frac{y_A(n+1) - y_A(n)}{T} \quad (1.21)$$

Replacing $y(nT)$ with $y_A(n)$, and writing $u(nT)$ as $u(n)$ for short,

$$f[y_A(n), u(n)] = \frac{y_A(n+1) - y_A(n)}{T} \quad (1.22)$$

Note the use of the equality in Equation 1.22, which enables $y_A(n)$ to be solved for, giving

$$y_A(n+1) = y_A(n) + Tf[y_A(n), u(n)], \quad n = 0, 1, 2, \dots \quad (1.23)$$

Equation 1.23 is called a difference equation, and given the initial condition $y_A(0)$, is easily solved in a recursive manner. To illustrate, consider the first-order system

$$\frac{d}{dt} y(t) + 2y(t) = u(t) = 3t, \quad y(0) = 1 \quad (1.24)$$

$$\frac{dy}{dt} = f(y, u) = -2y + u \quad (1.25)$$

Using Equation 1.23, the difference equation for obtaining $y_A(n)$, $n = 0, 1, 2, \dots$ is

$$y_A(n+1) = y_A(n) + T[-2y_A(n) + u(n)], \quad n = 0, 1, 2, \dots \quad (1.26)$$

$$= (1 - 2T)y_A(n) + Tu(n), \quad n = 0, 1, 2, \dots \quad (1.27)$$

Letting $\alpha = (1 - 2T)$,

$$y_A(n+1) = \alpha y_A(n) + Tu(n), \quad n = 0, 1, 2, \dots \quad (1.28)$$

where

$$u(n) = u(nT) = u(t) \Big|_{t=nT} = 3t \Big|_{t=nT} = 3nT, \quad n = 0, 1, 2, \dots \quad (1.29)$$

Replacing $u(n)$ in Equation 1.28 with $u(n)$ in Equation 1.29 gives

$$y_A(n+1) = \alpha y_A(n) + 3nT^2, \quad n = 0, 1, 2, \dots \quad (1.30)$$

Equation 1.30 is the difference equation which is solved recursively to generate the approximate solution $y_A(n)$, $n = 0, 1, 2, \dots$.

Starting with $n = 0$,

$$n = 0: \quad y_A(1) = \alpha y_A(0) + 3(0)T^2 \quad (1.31)$$

Choosing $y_A(0) = y(0)$ gives

$$n = 0: \quad y_A(1) = \alpha y(0) \quad (1.32)$$

$$n = 1: \quad y_A(2) = \alpha y_A(1) + 3(1)T^2 \quad (1.33)$$

$$= \alpha[\alpha y(0)] + 3T^2 \quad (1.34)$$

$$= \alpha^2 y(0) + 3T^2 \quad (1.35)$$

$$n = 2: \quad y_A(3) = \alpha y_A(2) + 3(2)T^2 \quad (1.36)$$

$$= \alpha[\alpha^2 y(0) + 3T^2] + 6T^2 \quad (1.37)$$

$$= \alpha^3 y(0) + 3(\alpha + 2)T^2 \quad (1.38)$$

$$n = 3: \quad y_A(4) = \alpha y_A(3) + 3(3)T^2 \quad (1.39)$$

$$= \alpha[\alpha^3 y(0) + 3(\alpha + 2)T^2] + 9T^2 \quad (1.40)$$

$$= \alpha^4 y(0) + 3T^2[\alpha(\alpha + 2) + 3] \quad (1.41)$$

The smaller the time step T , the closer $y_A(n)$, $n = 0, 1, 2, \dots$ will be to the exact solution $y(t)$ at $t = 0, T, 2T, \dots$. Finding an approximate solution for $y(t)$ on the interval $0 \leq t \leq t_{final}$ requires $\frac{t_{final}}{T}$ iterations of the difference equation. There is a trade-off between accuracy of the approximate solution $y_A(n)$, $n = 0, 1, 2, \dots$ and the computational effort to generate it.

Equation 1.25 with input $u(t) = 3t$ and initial condition $y(0) = 1$ is easily solved by analytical methods. The exact solution is given by

$$y(t) = 1.75e^{-2t} + 1.5t - 0.75, \quad t \geq 0 \quad (1.42)$$

EXAMPLE 1.1

- Find the approximate solution to the differential equation (Equation 1.24) over the interval $0 \leq t \leq 2$ by recursive solution of the difference equation (Equation 1.30) for $T = 0.01, 0.025, 0.05$ and 0.1 . Plot the approximate solution for each value of T and the exact solution given in Equation 1.42.
- Compare approximate and exact solutions at $t = 0, 0.2, 0.5, 1, 1.5, 2$ when $T = 0.01$ and $T = 0.1$.
- Matlab program “Ch1_Ex1_1.m” solves the difference equation (Equation 1.30) recursively and also computes points along the exact solution (Equation 1.42). The results are shown in [Figure 1.9](#).

Note, the approximate solution is known only at times $t_n = nT$, $n = 0, 1, 2, \dots$. For purposes of clarity, only a subset of the discrete points are plotted except for the case when $T = 0.1$.

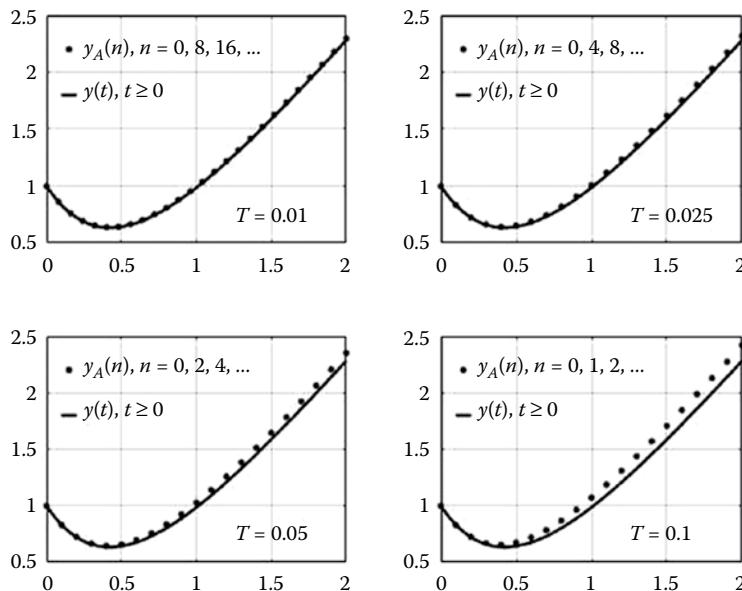
Exact and approximate solutions for various numerical values of the time step T **FIGURE 1.9** Comparison of exact and approximate solution of first-order system response for different values of T .

Figure 1.9 illustrates how the accuracy of the approximate solution can be improved by decreasing the time step T (at the expense of requiring additional calculations). The approximate solutions are seen to converge to the exact solution as the time step decreases. Further it appears that setting the time step below $T = 0.025$ may not be necessary unless extreme accuracy is required.

- b. **Tables 1.1** and **1.2** compare the approximate and exact solution at $t = 0, 0.2, 0.5, 1, 1.5$ and 2 when $T = 0.01$ and $T = 0.1$, respectively.

Note the percent error,

$$\% \text{ Error} = 100 \times \frac{y_A(n) - y(t_n)}{y(t_n)} \quad (1.43)$$

is roughly 10 times greater when the time step $T = 0.1$ compared to when $T = 0.01$.

Let us now apply what we have learned about finding approximate solutions of first-order system response to the tank example in Section 1.2. Recall that the tank dynamics are governed by the first-order differential equation (Equation 1.18),

$$A \frac{dH}{dt} + \frac{1}{R} H = F_1 \quad (1.44)$$

Solving for the derivative function $f(F_1, H)$ gives

$$f(F_1, H) = \frac{dH}{dt} = \frac{1}{A} \left(F_1 - \frac{1}{R} H \right) \quad (1.45)$$

TABLE 1.1

Comparison of Exact and Approximate ($T = 0.01$) Solution of First-Order System Response at Different Times

n	$t_n = nT$	$y_A(n)$	$y(t_n)$	% Error
0	0	1	1	0
20	0.2	0.7233	0.7231	0.03
50	0.5	0.6468	0.6438	0.47
100	1.0	0.9951	0.9868	0.84
150	1.5	1.5988	1.5871	0.74
200	2.0	2.2955	2.2821	0.59

TABLE 1.2

Comparison of Exact and Approximate ($T = 0.1$) Solution of First-Order System Response at Different Times

n	$t_n = nT$	$y_A(n)$	$y(t_n)$	% Error
0	0	1	1	0
2	0.2	0.7240	0.7231	0.13
5	0.5	0.6743	0.6438	4.74
10	1.0	1.0718	0.9868	8.61
15	1.5	1.7063	1.5871	7.51
20	2.0	2.4148	2.2821	5.98

Based on Equation 1.23, the difference equation for approximating the tank level response is

$$H_A(n+1) = H_A(n) + T f_i(n), H_A(n) \quad (1.46)$$

$$= H_A(n) + T \cdot \frac{1}{A} \left(f_i(n) - \frac{1}{R} H_A(n) \right) \quad (1.47)$$

$$= \left(1 - \frac{T}{AR} \right) H_A(n) + \left(\frac{T}{A} \right) f_i(n) \quad (1.48)$$

Given the input flow $f_i(t)$, $t \geq 0$ and the initial tank level $H_A(0)$, Equation 1.48 is the difference equation which can be solved recursively to obtain the approximate tank level response $H_A(n)$, $n = 0, 1, 2, \dots$. Starting with $n = 0$,

$$n = 0: \quad H_A(1) = \left(1 - \frac{T}{AR} \right) H_A(0) + \left(\frac{T}{A} \right) f_i(0) \quad (1.49)$$

$$n = 1: \quad H_A(2) = \left(1 - \frac{T}{AR} \right) H_A(1) + \left(\frac{T}{A} \right) f_i(1) \quad (1.50)$$

$$= \left(1 - \frac{T}{AR} \right) \left[\left(1 - \frac{T}{AR} \right) H_A(0) + \left(\frac{T}{A} \right) f_i(0) \right] + \left(\frac{T}{A} \right) f_i(1) \quad (1.51)$$

$$= \left(1 - \frac{T}{AR} \right)^2 H_A(0) + \left(1 - \frac{T}{AR} \right) \left(\frac{T}{A} \right) f_i(0) + \left(\frac{T}{A} \right) f_i(1) \quad (1.52)$$

$$n = 2: \quad H_A(3) = \left(1 - \frac{T}{AR}\right) \left[\left(1 - \frac{T}{AR}\right)^2 H_A(0) + \left(1 - \frac{T}{AR}\right) \left(\frac{T}{A}\right) F_1(0) + \left(\frac{T}{A}\right) F_1(1) \right] + \left(\frac{T}{A}\right) F_1(2) \quad (1.53)$$

$$= \left(1 - \frac{T}{AR}\right)^3 H_A(0) + \left(1 - \frac{T}{AR}\right)^2 \left(\frac{T}{A}\right) F_1(0) + \left(1 - \frac{T}{AR}\right) \left(\frac{T}{A}\right) F_1(1) + \left(\frac{T}{A}\right) F_1(2) \quad (1.54)$$

Based on the first few iterations, a general formula for finding $H_A(n)$, for any n , is

$$H_A(n) = \left(1 - \frac{T}{AR}\right)^n H(0) + \left(\frac{T}{A}\right) \sum_{k=0}^{n-1} \left(1 - \frac{T}{AR}\right)^{n-k-1} F_1(k), \quad n = 0, 1, 2, \dots \quad (1.55)$$

where $H_A(0)$ is replaced by the initial tank level $H(0)$.

When specific values of $H_A(n)$ are required, say $H_A(100)$, Equation 1.55 eliminates the need for recursive solution of Equation 1.48 to find $H_A(n)$, $n = 0, 1, 2, \dots, 99$. The summation in Equation 1.55 requires some effort; however, the z-transform introduced in [Chapter 4](#) provides a way to avoid the sum altogether.

The quantity $\left(1 - \frac{T}{AR}\right)$ must be less than 1 for both stable and accurate results. Keeping this in mind, Equation 1.55 demonstrates the diminishing influence of the initial tank level as time progresses.

Equation 1.55 also indicates that all past inputs $F_1(k)$, $k = 0, 1, 2, \dots, n-1$ are needed to obtain $H_A(n)$, a property known as infinite memory. However, recent input flow values are weighted higher than older values (see Equation 1.54).

EXAMPLE 1.2

A tank with cross-sectional area of 10 ft^2 receives a constant input flow of $5 \text{ ft}^3/\text{min}$. The fluid resistance of the tank is 2 ft per ft^3/min , and the tank is initially filled to a level of 4 ft .

- Find the difference equation for obtaining an approximate solution for the level $H(t)$ using a time step of $T = 0.25 \text{ min}$.
- Solve the difference equation recursively to obtain the approximate fluid level $H_A(n)$, $n = 1, 2, 3$.
- Use Equation 1.55 to find $H_A(3)$ and compare your answer to the result from part (b).

$$\text{a. } \left(\frac{T}{A}\right) = \frac{0.25}{10} = 0.025, \quad \left(1 - \frac{T}{AR}\right) = 1 - \frac{0.25}{10(2)} = 0.9875$$

$$H_A(0) = H(0) = 4, \quad F_1(n) = 5, \quad n = 0, 1, 2, \dots$$

The difference equation (Equation 1.48) is

$$H_A(n+1) = 0.9875H_A(n) + (0.025)5, \quad n = 0, 1, 2, \dots \quad (1.56)$$

- b. $H_A(n)$, $n = 1, 2, 3$ are computed as follows:

$$\begin{aligned} n = 0: \quad H_A(1) &= 0.9875H_A(0) + 0.125 \\ &= 4.0750 \end{aligned} \quad (1.57)$$

$$\begin{aligned} n = 1: \quad H_A(2) &= 0.9875H_A(1) + 0.125 \\ &= 0.9875(4.0750) + 0.125 \\ &= 4.1491 \end{aligned} \quad (1.58)$$

$$\begin{aligned}
 n = 2: \quad H_A(3) &= 0.9875H_A(2) + 0.125 \\
 &= 0.9875(4.1491) + 0.125 \\
 &= 4.2222
 \end{aligned} \tag{1.59}$$

c. From Equation 1.55 with $n = 3$,

$$\begin{aligned}
 H_A(3) &= (0.9875)^3(4) + 0.025 \sum_{k=0}^2 (0.9875)^{3-k-1}(5) \\
 &= 3.8519 + 0.025[(0.9875)^2(5) + (0.9875)(5) + 5] \\
 &= 4.2222
 \end{aligned}$$

Due to the simple nature of the input, that is, $F_1(t) = F$, $t \geq 0$, the analytical solution of the first-order differential equation model (Equation 1.44) is given by

$$H(t) = R\bar{F} + [H(0) - R\bar{F}]e^{-t/\Delta t} \tag{1.60}$$

It is instructive to compare the approximate solution based on the difference equation approach with the exact solution shown in Equation 1.60. Table 1.3 includes both solutions at equally spaced intervals for the first 2 min of the response.

Graphs of the exact response $H(t)$ and the approximate response $H_A(n)$, $n = 0, 8, 16, \dots$ are shown in Figure 1.10.

By observation of Figure 1.10, it appears that the exact and approximate solutions for the tank level are in close agreement. The step size T is the determining factor in terms of how close the two solutions are at the discrete points in time where the approximate solution is defined.

Generally speaking, an assessment of whether the numerical value selected for T is reasonable cannot be made on the basis of comparing the approximate solution with the exact solution to the differential equation. Analytical solutions are rare due to the complexity of most real-world system models. A logical approach to finding an acceptable step size is to obtain approximate solutions with different step sizes (an order of magnitude apart) and comparing the results. If the approximate solutions are substantially identical, the smaller step size is eliminated from consideration. Conversely, if the approximate solutions are not close, the larger value of T is discarded. Eventually, a value for T will be found, which balances accuracy and computational requirements. This point will be revisited in greater detail after the subject of numerical integration is discussed.

TABLE 1.3
Comparison of Approximate and
Exact Tank Level Response

n	$t_n = nT$	$H_A(n)$	$H(t_n)$
0	0	4.0	4.0
1	0.25	4.0750	4.0745
2	0.5	4.1491	4.1481
3	0.75	4.2222	4.2208
4	1.0	4.2944	4.2926
5	1.25	4.3657	4.3635
6	1.5	4.4362	4.4335
7	1.75	4.5057	4.5027
8	2.0	4.5744	4.5710

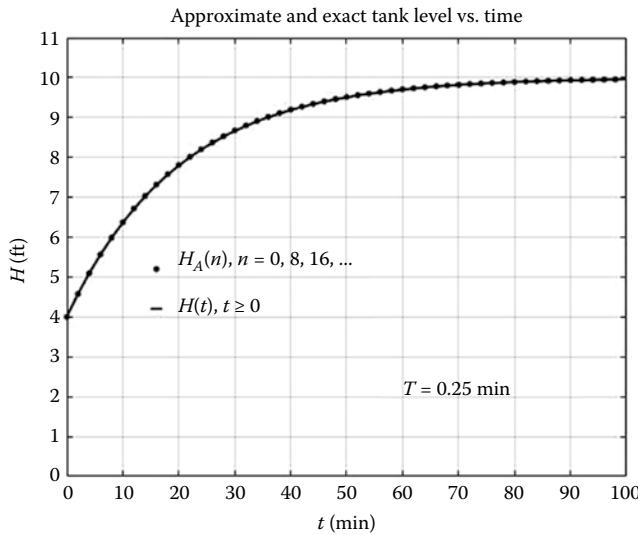


FIGURE 1.10 Approximate and exact solutions for tank level vs. time.

EXERCISES

- 1.4 Find the difference equation, similar to Equation 1.48, relating $F_0(n+1)$ to $F_0(n)$ and $F_1(n)$.
- 1.5 A tank with cross sectional area $A = 5 \text{ ft}^2$ is initially filled to a level of 10 ft. The flow out is given by $F_0 = H/R$, $R = 1 \text{ ft per ft}^3/\text{min}$. There is no flow in to the tank.
 - a. Find $H_A(n)$, $n = 0, 1, 2, \dots, 10$ when $T = 2.5 \text{ min}$.
 - b. Find $H_A(n)$, $n = 0, 1, 2, \dots, 25$ when $T = 1 \text{ min}$.
 - c. Find $H_A(n)$, $n = 0, 1, 2, \dots, 100$ when $T = 0.25 \text{ min}$.
 - d. Plot the results and comment on the differences.
- 1.6 Repeat Exercise 1.5 for the case where the outflow is described by $F_0 = cH^{1/2}$, $c = 3 \text{ ft}^3/\text{min per ft}^{1/2}$.
- 1.7 Rework Example 2.1 using the Trial and Error method for determining a suitable value of T . Start with $T = 10 \text{ min}$ and calculate $H_A(n)$, $n = 0, 1, 2, \dots, n_f$ where $n_f T = 100 \text{ min}$. Repeat the steps with $T = 5 \text{ min}$, 2.5 min , 1.25 min , etc. until the approximations of $H(10)$, $H(20)$, $H(30)$, ..., $H(100)$ are in agreement to at least one place after the decimal point. Use Table E1.7 for comparisons. Extend the table to smaller values of T if necessary.

TABLE E1.7

n	$H_A(n)$		$H_A(n)$		$H_A(n)$		$H_A(n)$	
	$T = 10$	n	$T = 5$	n	$T = 2.5$	n	$T = 1.25$	
0	0	0	0	0	0	0	0	0
1	2	2	4	4	8	8	16	16
2	4	4	8	8	16	16	32	32
3	6	6	12	12	24	24	48	48
4	8	8	16	16	32	32	64	64
5	10	10	20	20	40	40	80	80
6	12	12	24	24	48	48	96	96
7	14	14	28	28	56	56	112	112
8	16	16	32	32	64	64	128	128
9	18	18	36	36	72	72	144	144
10	20	20	40	40	80	80	160	160

1.8 An alternate model of the tank relates the outflow and liquid level according to

$$F_0(t) = \alpha [H(t)]^{1/2}$$

- Develop a new discrete-time model of the tank using the above relationship in conjunction with the differential equation $A(dH/dt) + F_0 = F_1$. The tank cross-sectional area is 10 ft^2 and the input flow is constant at $5 \text{ ft}^3/\text{min}$. The tank is initially filled to a level of 4 ft . Assume $\alpha = 2 \text{ ft}^3/\text{min per ft}^{1/2}$.
- Calculate the approximate tank level for the first minute using a step size $T = 0.25 \text{ min}$.
- Consider the same tank with zero flow in and an initial fluid level of 25 ft . Write a program to calculate the approximate level of the tank as it empties. Choose $T = 0.1 \text{ min}$.
- The analytical solution for the level $H(t)$ when $F_1(t) = 0, t \geq 0$ is given by

$$H(t) = \left(H_0^{1/2} - \frac{\alpha t}{2A} \right)^2$$

where H_0 is the initial tank level. Compare the results from Part (c) to the exact solution. Present the comparison of results in tabular and graphical form.

1.9 A holding tank serves as an effective way of smoothing variations in the flow of a liquid. For example, suppose the liquid flow rate from an upstream process is

$$F_1(t) = \bar{F} + f \sin\left(\frac{2\pi t}{T}\right), \quad t \geq 0$$

where \bar{F} is an average flow, f is the fluctuation about the average flow, and T is the period of the fluctuations. Nominal parameter values for the input flow rate are $\bar{F} = 250 \text{ ft}^3/\text{min}$, $f = 50 \text{ ft}^3/\text{min}$ and $T = 15 \text{ min}$.

A holding tank is placed between the source $F_1(t)$ and a downstream process which requires a more constant input flow rate, $F_0(t)$ as shown in Figure E1.9. The downstream process requires that the sustained fluctuations in the flow $F_0(t)$ be no larger than $10 \text{ ft}^3/\text{min}$. Assume the tank is linear and the fluid resistance $R = 0.25 \text{ ft per ft}^3/\text{min}$.

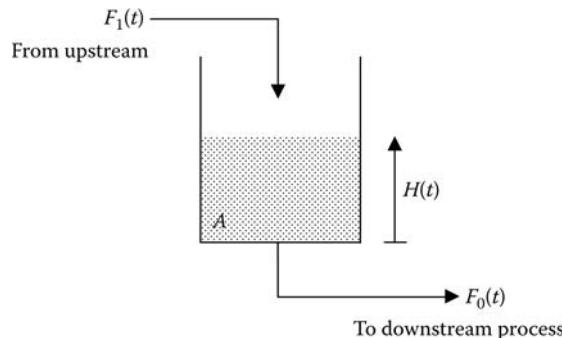


FIGURE E1.9

- Find the difference equation for $F_{0,A}(n)$, $n = 0, 1, 2, 3, \dots$. Leave the tank cross sectional area A as a parameter.
- Write a program to solve the difference equation with $\Delta t = 0.5 \text{ min}$ for a starting value of $A = 100 \text{ ft}^2$. Graph both $F_{0,A}(n)$ and $H_A(n)$, $n = 0, 1, 2, \dots$ for a period of time sufficient to determine if the design criterion is satisfied. Assume the tank is initially empty.

- c. Repeat part (b) with a new value of A until the design criterion is satisfied, i.e. the sustained fluctuations in $F_0(t)$ are equal to $10 \text{ ft}^3/\text{min}$.
- d. Graph the discrete-time signals $F_1(n)$ and $F_{0,A}(n)$, $n = 0, 1, 2, 3, \dots$ for the tank whose area is the value determined in Part (c).

1.4 FIRST LOOK AT DISCRETE-TIME SYSTEMS

The variables $F_1(t)$, $F_0(t)$, and $H(t)$ in the liquid tank shown in Figure 1.3 are referred to as continuous-time (or simply continuous) signals. The reason is because there is a continuum of values between any two points along the t -axis where the variables are defined. Equation 1.18 is a continuous-time model and the system is a continuous-time system because it involves only continuous-time variables.

In contrast to the continuous-time signals $F_1(t)$, $F_0(t)$, and $H(t)$, the sequence of sampled input flow values, $F_1(n)$, $n = 0, 1, 2, \dots$ and the sequence of approximate tank levels $H_A(n)$, $n = 0, 1, 2, \dots$ are classified as discrete-time (discrete for short) signals because the independent variable “ n ” is discrete in nature. The difference equation (Equation 1.48) is referred to as a discrete-time model, and the underlying system with purely discrete-time input and output signals is likewise a discrete-time system.

Figure 1.11 illustrates both the continuous and discrete representations of the tank dynamics.

1.4.1 INHERENTLY DISCRETE-TIME SYSTEMS

The discrete model of the tank dynamics relates $H_A(n)$, an approximation of the continuous tank level $H(t)$ and $F_1(n)$, the sampled version of the continuous flow $F_1(t)$. In contrast, inherently discrete-time systems involve discrete signals which are not related to continuous signals. For example, consider a discrete-time system governed by the difference equation

$$y(n) = \frac{1}{2} \left[y(n-1) + \frac{u(n)}{y(n-1)} \right], \quad n = 0, 1, 2, \dots \quad (1.61)$$

Equation 1.61 is simply a rule for transforming a discrete input signal $u(n)$ into an appropriate discrete output signal $y(n)$. Is this discrete-time system useful? Let us investigate its behavior for the case where $u(n)$ is a constant, for example, $u(n) = 25$, $n = 0, 1, 2, \dots$. Before we are able to compute $y(n)$, $n = 0, 1, 2, \dots$, a starting or initial value, in this case $y(-1)$ must be given. If we choose $y(-1) = 1$, the first few values of $y(n)$ are

$$n = 0: \quad y(0) = \frac{1}{2} \left[y(-1) + \frac{u(0)}{y(-1)} \right] = \frac{1}{2} \left[1 + \frac{25}{1} \right] = 13 \quad (1.62)$$

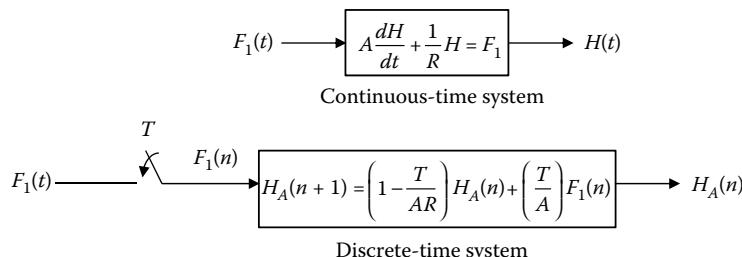


FIGURE 1.11 Continuous-time and discrete-time representations of tank dynamics.

$$n=1: \quad y(1) = \frac{1}{2} \left[y(0) + \frac{u(1)}{y(0)} \right] = \frac{1}{2} \left[13 + \frac{25}{13} \right] = 7.4615 \quad (1.63)$$

$$n=2: \quad y(2) = \frac{1}{2} \left[y(1) + \frac{u(2)}{y(1)} \right] = \frac{1}{2} \left[7.4615 + \frac{25}{7.4615} \right] = 5.4060 \quad (1.64)$$

$$n=3: \quad y(3) = \frac{1}{2} \left[y(2) + \frac{u(3)}{y(2)} \right] = \frac{1}{2} \left[5.4060 + \frac{25}{5.4060} \right] = 5.0152 \quad (1.65)$$

$$n=4: \quad y(4) = \frac{1}{2} \left[y(3) + \frac{u(4)}{y(3)} \right] = \frac{1}{2} \left[5.0152 + \frac{25}{5.0152} \right] = 5.0000 \quad (1.66)$$

Using different positive constants for $u(n)$ and other starting values for $y(-1)$ will reveal an interesting property of the system, namely

$$\lim_{n \rightarrow \infty} y(n) = \sqrt{u} \quad (1.67)$$

Hence, the primary purpose of the discrete-time system governed by Equation 1.61 is to compute the square root of its positive-valued constant input $u(n)$. The discrete signals $u(n)$ and $y(n)$ are plotted in Figure 1.12.

Another inherently discrete-time system is one we are all familiar with, namely, an interest-bearing account such as a bank account. The discrete signals of interest are $y(n)$, the account balance at the end of the n th interest period, and $u(n)$, the net deposit for the n th interest period. Figure 1.13 shows the discrete-time system.

Consider an account with an interest rate i (per interest period). The balance at the end of the n th interest period, $y(n)$ is the sum of

- the balance at the end of the $(n - 1)$ st period: $y(n - 1)$
- the interest earned for the n th interest period: $i \cdot y(n - 1)$
- the net deposit for the period: $u(n)$

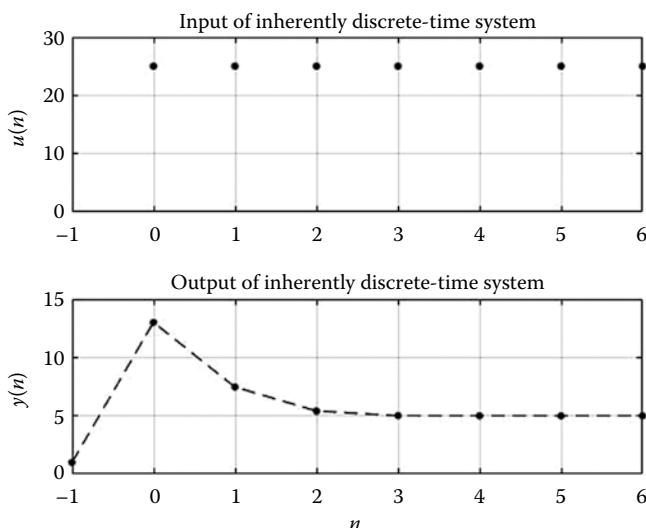


FIGURE 1.12 Illustration of using discrete-time system in Equation 1.61 to find a square root.

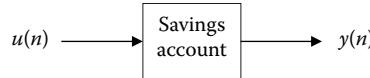


FIGURE 1.13 Example of an inherently discrete-time system.

Therefore, the model for this inherently discrete-time system is

$$y(n) = y(n-1) + iy(n-1) + u(n) \quad (1.68)$$

EXAMPLE 1.3

A college trust fund is set up with \$5000 on Jan 1, 2000. Starting on Jan 1, 2001 and every year thereafter, \$1000 is added to the fund, which earns 7.5% interest annually.

- a. Track the end of year fund balance for the first several years.
- b. Find the account balance at the end of Year 18th year.

- a. The discrete-time model is

$$y(n) = y(n-1) + 0.075y(n-1) + u(n), \quad n = 1, 2, 3, \dots \quad (1.69)$$

with input

$$u(n) = 1000, \quad n = 1, 2, 3, \dots \quad (1.70)$$

and initial condition $y(0) = 5000$.

The account balance at the end of years 1, 2, and 3 are computed as follows:

$$\begin{aligned} n = 1: \quad y(1) &= y(0) + 0.075y(0) + u(1) \\ &= 5000 + 0.075(5000) + 1000 \\ &= 6375 \end{aligned} \quad (1.71)$$

$$\begin{aligned} n = 2: \quad y(2) &= y(1) + 0.075y(1) + u(2) \\ &= 6375 + 0.075(6375) + 1000 \\ &= 7853.13 \end{aligned} \quad (1.72)$$

$$\begin{aligned} n = 3: \quad y(3) &= y(2) + 0.075y(2) + u(3) \\ &= 7853.13 + 0.075(7853.13) + 1000 \\ &= 9442.11 \end{aligned} \quad (1.73)$$

- b. The recursive solution could be continued for $n = 4, 5, 6, \dots, 18$ resulting in the fund's balance at the end of the 18th year. However, a general solution of the discrete-time model is preferable since it can be evaluated for any value of the variable n .

Expressing the discrete-time model in Equation 1.68 as

$$y(n) = (1+i)y(n-1) + A, \quad n = 1, 2, 3, \dots \quad (1.74)$$

$$= \alpha y(n-1) + A, \quad n = 1, 2, 3, \dots \quad (1.75)$$

where $\alpha = 1 + i$ and A is the constant net deposit each interest period.

$$n = 1: \quad y(1) = \alpha y(0) + A \quad (1.76)$$

$$n = 2: \quad y(2) = \alpha y(1) + A \quad (1.77)$$

$$= \alpha[\alpha y(0) + A] + A \quad (1.78)$$

$$= \alpha^2 y(0) + \alpha A + A \quad (1.79)$$

$$n = 3: \quad y(3) = \alpha y(2) + A \quad (1.80)$$

$$= \alpha[\alpha^2 y(0) + \alpha A + A] + A \quad (1.81)$$

$$= \alpha^3 y(0) + \alpha^2 A + \alpha A + A \quad (1.82)$$

suggesting the general expression for $y(n)$ is

$$y(n) = \alpha^n y(0) + (1 + \alpha + \alpha^2 + \cdots + \alpha^{n-1})A, \quad n = 0, 1, 2, \dots \quad (1.83)$$

Further simplification is possible using the closed form of the finite geometric series in Equation 1.83 resulting in

$$y(n) = \alpha^n y(0) + \left(\frac{1 - \alpha^n}{1 - \alpha} \right) A, \quad n = 0, 1, 2, \dots \quad (1.84)$$

The account balance after 18 years is computed from Equation 1.84 with $\alpha = 1.075$, $y(0) = 5000$ and $A = 1000$.

$$\begin{aligned} y(18) &= (1.075)^{18}(5000) + \left[\frac{1 - (1.075)^{18}}{1 - 1.075} \right] 1000 \\ &= 54,056.41 \end{aligned} \quad (1.85)$$

The results from part (a) can be verified using the general solution in Equation 1.84.

EXERCISES

- 1.10 Prove that the output of the discrete-time system in Equation 1.61 will approach the square root of the input, any positive constant “A”. In other words, show that

$$\lim_{n \rightarrow \infty} y(n) = \sqrt{A}$$

where $u(n) = A$, $n = 0, 1, 2, 3, \dots$

1.5 CASE STUDY: POPULATION DYNAMICS (SINGLE SPECIES)

The population of a country is under investigation. Unlike the liquid tank example, there is no scientific principle to serve as a foundation for deriving a mathematical model that can be used to predict future populations. Instead, empirical observations of historical birth and death rates, immigration and emigration patterns, and a host of other pertinent data are utilized.

One hundred years of observed population data, recorded at intervals of 10 years, is given in [Table 1.4](#).

TABLE 1.4
Population Data for 100 Years

t (years)	$P_{\text{obs}}(t)$, millions
0	3.0000
10	3.2276
20	4.5759
30	6.9570
40	8.7618
50	9.1536
60	11.2669
70	14.5153
80	16.5059
90	17.9563
100	19.5078

Based on the available data, researchers are convinced that the population is adequately modeled by the following differential equation, referred to in the literature as logistic growth (Haberman 1977).

$$\frac{dP}{dt} = cP(P_m - P) \quad (1.86)$$

$P = P(t)$ is the population “ t ” years after the initial population was recorded. The parameters c and P_m influence the specific growth pattern behavior. The model ignores immigration and emigration and all other external inputs, which influence dP/dt , the rate at which the population changes.

The system model in Equation 1.86 is said to be autonomous, meaning there are no additional terms independent of P , as might be the case if immigration or emigration inputs as a function of time were considered. The dynamics depend solely on initial conditions and the system parameters. It is also referred to as an unforced system since there are no external inputs.

Statistical analyses of the population data has resulted in estimated values for c and P_m to be 1.25×10^{-9} and 25 million, respectively. It is now 100 years since the initial population was measured. Government planners are interested in determining what the likely population will be over the next several decades. A method is needed to obtain an approximate solution for $P(t)$, that is, a difference equation for $P_A(n) \approx P(nT)$, $n = 0, 1, 2, \dots$ is required.

Employing the method from Section 1.2 for obtaining a difference equation to approximate the dynamics of a first-order system,

$$P_A(n+1) = P_A(n) + T \cdot f[P_A(n)] \quad (1.87)$$

where $f[P_A(n)]$ is the derivative function, that is,

$$f[P_A(n)] = cP_A(n)[P_m - P_A(n)] \quad (1.88)$$

Combining Equations 1.87 and 1.88 results in

$$P_A(n+1) = P_A(n) + TcP_A(n)[P_m - P_A(n)] \quad (1.89)$$

Note, Equation 1.89 could just as easily have been found by simply replacing $\frac{dP}{dt}$ in Equation 1.86 with the divided difference term $\frac{P_A(n+1) - P_A(n)}{T}$ and replacing P with $P_A(n)$ on the right hand side of the equation.

Simplifying Equation 1.89 leads to the required difference equation, namely

$$P_A(n+1) = \{1 + Tc[P_m - P_A(n)]\}P_A(n) \quad (1.90)$$

Since our interest is in predicting populations for 100 years and beyond, we need to solve Equation 1.90 over a suitable range of values for the discrete-time variable “ n ”. The appropriate integer values depend on the size of our time step T . For simplicity, we shall choose $T = 1$ year, necessitating the calculation of $P_A(101)$, $P_A(102)$, ..., $P_A(130)$ to obtain predictions for a 30-year time span.

A recursive solution seems like our only alternative, since a general solution is not easily achievable. The Matlab file “Ch1_CaseStudy.m” includes the necessary statements to solve Equation 1.90 recursively and produce the results shown in [Table 1.5](#). Note, the initial value $P_A(0)$ is the initial observed population of 3 million.

A comparison of the numbers for $P_{\text{obs}}(t)$ and $P_A(n)$ indicates that the modelers were justified in assuming logistic growth of the population for the 100 years corresponding to the recorded data. Naturally, this assumes that the approximate solution values $P_A(n)$, $n = 0, 1, 2, \dots, 100$ are reasonably close to the exact solution $P(t)$, $t = 0, 1, 2, \dots, 100$.

Ordinarily, dynamic population models are not amenable to exact solutions. However, the analytical solution to Equation 1.86 is known and given in Equation 1.91.

$$P(t) = \frac{P_m P(0)}{P(0) + [P_m - P(0)]e^{-cP_m t}}, \quad t \geq 0 \quad (1.91)$$

Knowing the exact solution, $P(t)$, to the continuous model differential equation, we can evaluate $P(t)$ at $t = 0, 10, 20, \dots, 100$ years for comparison with the discrete output $P_A(n)$, $n = 0, 10, 20, \dots, 100$ years obtained from solution of the difference equation. This allows us to determine if the current step size $T = 1$ year needs to be adjusted.

Comparing the last two columns in [Table 1.5](#) should convince us that the step size T does not need to be reduced. While it's possible to reduce the discrepancy between the approximate and exact solutions by lowering T , it's hardly justified in view of the fact that the logistic growth model, Equation 1.86, is itself only an approximate representation of the true population dynamics.

The data in [Table 1.5](#) are presented in graphical form in [Figure 1.14](#). The difference equation used to find $P_A(n)$, $n = 0, 1, 2, \dots, 100$ is used to predict the future population for the next 30 years. The projected populations for years 100, 120, and 130 are included in [Table 1.5](#) and appear as data points in [Figure 1.14](#).

The previous point relating to the accuracy of the approximate solution is worth reiterating. Extremely accurate discrete solutions of nonlinear differential equation models are generally not warranted unless the continuous models themselves are highly detailed with known accuracy.

Once we have an approximate solution of a continuous model response, how can we be certain if the two responses are in close agreement with each other? There is no simple answer; however, it's worth remembering that

1. the difference equations in the discrete-time model converge to the differential equations of the continuous-time model in the limiting case when the step size! approaches zero.
2. the discrete-time solutions approach the exact solutions of the continuous-time model as the step size T is reduced to zero.

Systematically reducing the step size T until the changes in the discrete-time outputs are within some tolerance demonstrates this convergence and is an effective way of selecting the step size for

TABLE 1.5
Comparison of Observed, Discrete ($T = 1$ year) and Continuous Populations

t (years)	$P_{obs}(t)$, millions	n	$P_A(n)$, millions	$P(t)$, millions
0	3.0000	0	3.0000	3.0000
10	3.2276	10	3.9161	3.9276
20	4.5759	20	5.0493	5.0759
30	6.9570	30	6.4129	6.4570
40	8.7618	40	8.0003	8.0618
50	9.1536	50	9.7778	9.8536
60	11.2669	60	11.6834	11.7669
70	14.5153	70	13.6325	13.7153
80	16.5059	80	15.5321	15.6059
90	17.9563	90	17.2976	17.3563
100	19.5078	100	18.8671	18.9078
110		110	20.2076	20.2310
120		120	21.3139	21.3226
130		130	22.2012	22.1990

future runs. We touched on this in the previous section as a way of choosing an appropriate value for the step size T .

On a cautionary note, the step size may have to be readjusted as conditions change, in particular, the frequency components of the inputs. We will have more to say about this when we consider the frequency response characteristics of continuous and discrete systems in [Chapter 4](#).

Further scrutiny of the logistic growth model, Equation 1.86, reveals several important and noteworthy characteristics of the underlying population dynamics. Expressing the model in a slightly different form,

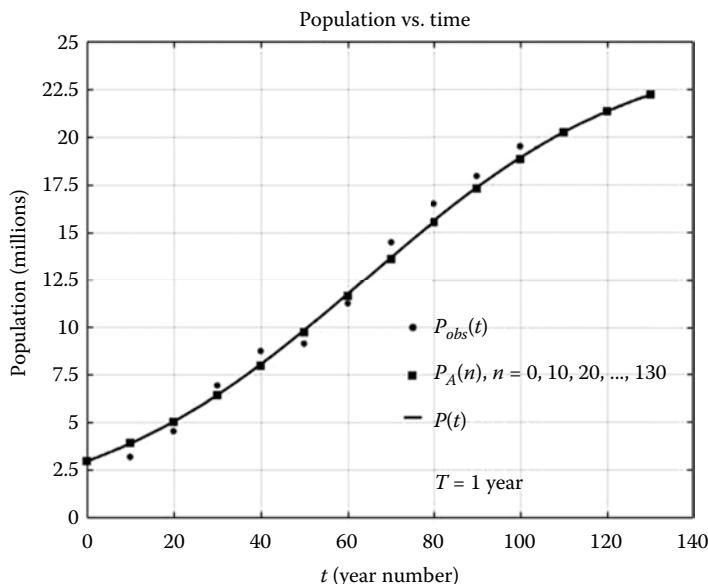


FIGURE 1.14 Observed, discrete (approximate) and continuous populations.

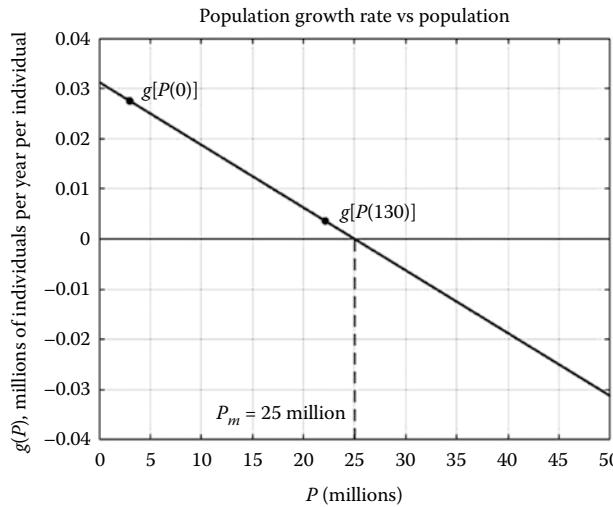


FIGURE 1.15 Plot of growth rate for a logistic growth population.

$$g(P) = \frac{1}{P} \frac{dP}{dt} = c(P_m - P) \quad (1.92)$$

where $g(P)$, the rate of change in population dP/dt divided by the population P , is called the population growth rate. Different population models are normally characterized by the terms appearing on the right hand side of Equation 1.92.

The growth rate for a logistic growth population model is shown in Figure 1.15.

We expect the population to be increasing whenever the growth rate is positive, since a positive growth rate implies the instantaneous rate of change in the population, that is, the first derivative is also positive. The logistic population growth rate declines linearly with increasing population,

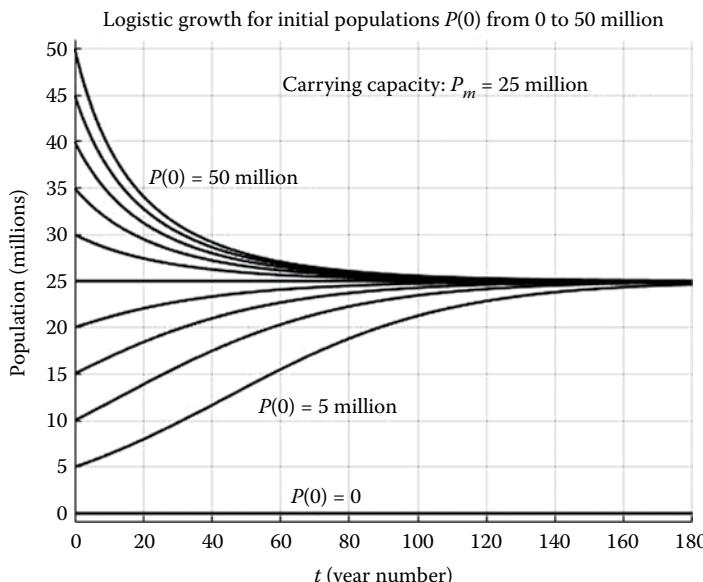


FIGURE 1.16 Logistic growth with different initial populations.

eventually reaching zero when the population reaches P_m . In logistic growth models, P_m is called the carrying capacity, which equals 25 million in this example.

Observe from [Figure 1.14](#) that the discrete and continuous model outputs for 130 years ranged from the initial population of 3 million people to somewhere around 22 million people. Looking at the line segment in [Figure 1.15](#), corresponding to this range of populations, we notice that the growth rate is positive. Hence the population should be monotonically increasing, as indeed it was.

Is it possible for a population $P(t)$ governed by a logistic growth model to ever assume values on both sides of its carrying capacity P_m ? For example, is the population growth shown in [Figure 1.14](#) capable of exceeding $P_m = 25$ million if we wait long enough? [Figure 1.16](#) shows population time histories for the logistic growth model considered previously ($c = 1.25 \times 10^{-9}$, $P_m = 25 \times 10^6$) with different starting populations.

It is clear that the population approaches its carrying capacity from below or above in asymptotic fashion. We should not be surprised if we consider what happens to the population growth rate $g(P)$ as the population approaches its carrying capacity from either direction (see [Figure 1.15](#)).

EXERCISES

- 1.11 Assume the logistic growth population model accurately predicts future populations.
- Some time in the future, the population will reach 98% of its carrying capacity. Find how many more years will it take for this to occur by using the difference equation given in Equation 1.89. Does it make a difference whether you start from $P_A(0) = 3$ million or $P_A(130) = 22.2012$ million from [Table 1.5](#)?
 - Compare the answer obtained in Part (a) with the analytical solution for $P(t)$.
 - The population growth rate $g(P)$ vs P in [Figure 1.15](#) does not explicitly involve time. Label the points on the growth rate curve corresponding to $\{t_i, P(t_i)\}$ where $t_0 = 0, t_1 = 25, t_2 = 50, t_3 = 75, t_4 = 100$.
 - The carrying capacity P_m in a logistic growth model is an equilibrium population, meaning that if the population at some point in time were equal to P_m it would remain there forever. Investigate whether its stable or not by supposing the population were slightly less or slightly more than P_m and determine if the population returns to the carrying capacity. Obtain several approximate solutions corresponding to different initial populations reasonably close to P_m .
 - Find the other equilibrium population of the logistic growth model and determine if its stable.
- 1.12 A simpler model for population growth of a species is one in which the growth rate is assumed constant, that is independent of the population. Mathematically, this is represented by

$$\text{Growth Rate} = g(P) = \frac{1}{P} \frac{dP}{dt} = k$$

Suppose a culture of bacteria is increasing in size according to the constant growth rate model above. The initial bacteria population is P_0 .

- Develop the difference equation for the discrete system approximation of the continuous model. Denote the discrete population as $P_A(n)$.
- Find the general solution for $P_A(n)$, $n = 0, 1, 2, 3, \dots$. Leave your answer in terms of k and P_0 . The constant growth rate $k = 0.01$ bacteria/min per bacteria and the initial number of bacteria is 10,000.
- Solve the difference equation recursively using a step size $T = 1$ min for $P_A(n)$, $n = 1, 2, 3, 4, 5$. Compare the result for $P_A(5)$ to the value obtained from the general solution found in Part (b).

- d. The analytical solution to the continuous model is $P(t) = P_0 e^{kt}$, $t \geq 0$. How long does it take for the population to reach 1 million?
- e. On the same graph, plot the continuous model output $P(t)$, $0 \leq t \leq 500$ and the discrete model output $P_A(n)$, $n = 0, 50, 100, 150, \dots, 1000$ when $T = 0.5$ min.
- f. Explain what would happen to a population with constant growth rate k , if k were negative.

2 Continuous-Time Systems

2.1 INTRODUCTION

Before we start our exploration of simulation, it is important for us to have some basic knowledge of how linear time-invariant (LTI) dynamic systems behave. The analysis of linear systems and how they respond to elementary types of inputs is straightforward. Linear systems appear as building blocks in more complex systems. Our intuitive understanding of the entire system is enhanced by recognizing the fundamental behavior of its linear components. Control systems, for example, are oftentimes composed of linear continuous-time components interconnected to produce a desirable response to commanded as well as uncontrollable or disturbance inputs.

Speaking of control systems, the mathematical model of the process being controlled is often nonlinear; however, a properly designed regulatory control system will limit excursions of the process variables. In fact, the design of the controller may be based on a linearized model of the nonlinear process owing to the wealth of tools available in the field of linear control theory. Simulation can play a valuable role here by shedding light on the validity of using a linearized mathematical model to approximate a nonlinear system model.

Modern simulation software contains user interfaces employing graphical icons that serve as building blocks for representing the linear continuous- and discrete-time components within a system. In order to exploit this feature, the simulation builder must understand the meaning and differences between the assortment of linear system blocks (integrators, first-order lags, second-order systems, transfer functions, and state space models) at his or her disposal. The material on first- and second-order system response, and state variables covered in this chapter and [Chapter 4](#), is intended as an introduction (or possibly a review) to the topic of linear continuous-time systems. There are literally dozens of excellent books on the subject of linear systems theory and linear control systems. Several are included in the references and the reader is encouraged to consult one or more as necessary.

In addition to the focus on linear systems in this chapter, one section includes several examples of nonlinear systems as well. A graphical illustration of how to linearize a nonlinear system model is presented as a preview of what is to come in [Chapter 7](#) where the subject is revisited in more detail.

Simulation of continuous-time systems is not discussed in detail until [Chapter 3](#) where the subject of numerical integration is introduced. However, a simulation model based on numerical differentiation, similar to what was done in [Chapter 1](#), is presented. At the conclusion of this chapter, the reader will be capable of representing simple continuous-time systems in state variable form and generate discrete-time model approximations of them, which can be solved in a recursive fashion.

2.2 FIRST-ORDER SYSTEMS

Continuous-time dynamic systems are said to be first order if the highest derivative of the dependent variable appearing in the mathematical model is first order. Systems in which a quantity of material or energy changes at a rate dependent on the amount of material or energy present are typically first order in nature. The general representation of a scalar first-order system is

$$\frac{dy}{dt} = f(t, y, u) \quad (2.1)$$

where

t is the continuous-time variable

$u = u(t)$ is the system input

$y = y(t)$ is the system output

$f(t, y, u)$ is the derivative function, which relates the rate of change in y to all three arguments

Not all three arguments will be present in every first-order model. Furthermore, it is possible for multiple inputs $u_1(t), u_2(t), \dots, u_r(t)$ to be present.

We begin our discussion of first-order systems with a special case, namely, where the derivative function is an explicit linear function of the input and output given by

$$f(t, y, u) = b_0 u(t) - a_0 y(t) \quad (2.2)$$

where a_0 and b_0 are constants. Combining Equations 2.1 and 2.2 gives

$$\frac{dy}{dt} + a_0 y(t) = b_0 u(t) \quad (2.3)$$

Equation 2.3 is a LTI, ordinary differential equation. In the time-varying case, one or both of the linear system parameters a_0 and b_0 are functions of the independent variable t . Equation 2.3 is commonly expressed as

$$\tau \frac{dy}{dt} + y(t) = Ku(t) \quad (2.4)$$

where τ and K are easily related to a_0 and b_0 by

$$\tau = \frac{1}{a_0} \quad \text{and} \quad K = \frac{b_0}{a_0} \quad (2.5)$$

Many simple real-world dynamic systems are modeled by the first-order differential equation (Equation 2.4). More complex systems often behave similarly to first-order systems under certain conditions. Furthermore, higher-order system models can be reduced to a system of coupled first-order models. Familiarity with first-order system response will prove useful later on when we undertake the task of simulating higher-order linear and nonlinear systems. For this reason, we explore some basic properties of first-order systems modeled by Equation 2.4.

2.2.1 STEP RESPONSE OF FIRST-ORDER SYSTEMS

When the input $u(t)$ is constant, that is, $u(t) = A, t \geq 0$, the solution to Equation 2.4 for $y(t)$ is obtained using Laplace transform or classical time-domain methods. It is given below:

$$y(t) = y(0)e^{-t/\tau} + KA(1 - e^{-t/\tau}), \quad t \geq 0 \quad (2.6)$$

where $y(0)$ is the initial value of the output $y(t)$. Several graphs of $y(t)$ are shown in Figure 2.1 for the cases where $y(0) = 0, K = 5, A = 2$, and $\tau = 0.5, 2, 5$, and 10 .

The graphs of $y(t)$ shown in Figure 2.1 are called the step response because the input resembles a step (changing from 0 to A at $t = 0$). Note that the initial condition is zero in all the step responses.

The constant A measures the amplitude of the input and is not an inherent system parameter. The system parameters are K and τ (or a_0 and b_0 from which they are computed). The first parameter

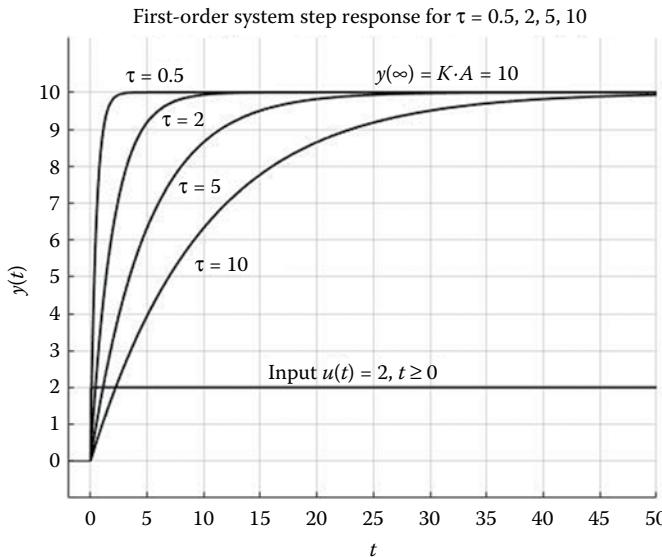


FIGURE 2.1 Step response of first-order system with different values of τ .

K is called the system DC or steady-state gain. It is so named because the final value of the output, $y(\infty)$, is easily computed from

$$y(\infty) = K \cdot u(\infty) = K \cdot A \quad (2.7)$$

which in this case is $y(\infty) = 5 \cdot 2 = 10$ (see Figure 2.1). The final value $y(\infty)$ is unaffected by the initial condition $y(0)$. However, the graph of $y(t)$ in Equation 2.6 certainly depends on $y(0)$, since that is where it starts. A first-order system like the one in Equation 2.4 is called a first-order lag because of the way the step response in Figure 2.1 lags the step input.

There are situations when the input to a first-order system is not a step; however, the input remains constant for a period of time that is largely relative to the parameter τ . Equation 2.7 enables us to readily compute the final output value prior to a change in the input. In essence, we are tracking the first-order system from one steady-state level to another, and the transient response (portion of the overall step response that decays to zero) is ignored. Even without knowledge of the transient response, it is possible to predict the amount of time necessary for the new steady state to be established.

In the first-order system modeled by Equation 2.4, the first derivative vanishes when the system is at steady state, leaving $y_{ss} = K\bar{u}$, where y_{ss} is the output at steady state in response to the constant input \bar{u} . A similar result is obtained from Equation 2.6 with A replaced by \bar{u} and t approaching ∞ .

The first-order system step responses shown in Figure 2.1 correspond to four distinct values for the parameter τ . It is apparent that while all approach the limiting value $y(\infty) = 10$, there is a noticeable difference in the amount of time required for each to get there. The individual step responses are correlated with the system parameter τ . This parameter is called the time constant of the first-order system. It is a measure of the speed of the step response as well as an indicator of the overall speed of the first-order system's dynamics. A "rule of thumb" for first-order systems is that the transient response vanishes after four or five time constants. The transient response component of the step response in Equation 2.6 with $y(0) = 0$ is

$$y_{tr}(t) = -KAe^{-t/\tau}, \quad t \geq 0 \quad (2.8)$$

when $t = 5\tau$,

$$y_{tr}(5\tau) = -KAe^{-5} = -KA(0.0067) \quad (2.9)$$

and the step response

$$y(5\tau) = KA(1 - e^{-5}) = 0.9933KA \quad (2.10)$$

is more than 99% complete. After four time constants have elapsed, the step response is slightly over 98% of its final value (see Figure 2.1).

First-order system models are commonplace in science, engineering, economics, business, etc. The liquid storage tank model in Section 1.2 and the population models considered in Section 1.5 are examples of first-order system models. Another example of a physical system described in terms of a first-order model is the simple electric circuit shown in Figure 2.2 along with the tank.

The circuit components are a capacitor C , a resistor R , and a voltage source $e_0(t)$. There is also a switch that connects the source to the rest of the circuit when it is in the closed position. Like the tank that stores its energy as a column of liquid, the circuit's capacitor stores energy in the form of electric charge. The potential energy of the fluid varies as the tank level changes and the electrical energy stored in the circuit varies with the amount of electrical charge stored in the capacitor. Both systems have a mechanism for dissipating energy. The tank does so whenever the level of fluid is dropping and the circuit dissipates energy in the resistor whenever there is current flowing.

The fluid resistance of the tank tells us the amount of effort, that is, height of liquid, required to produce a unit of flow from the tank. A typical unit for fluid resistance is ft per ft^3/min . The electrical counterpart is the electrical resistor that also measures the driving force, in this case, the voltage applied to the resistor, necessary to produce a unit of current flow, measured in amperes. The unit of electric resistance is volts/ampere, commonly called ohms.

Choosing the voltage across the capacitor $v_c(t)$ as the output, the circuit model is easily derived using basic principles of electrical circuits. The result is

$$RC \frac{d}{dt} v_c(t) + v_c(t) = e_0(t) \quad (2.11)$$

Comparison of Equation 2.11 with the standard form introduced in Equation 2.4 reveals the time constant of the circuit $\tau = RC$ and the steady-state gain $K = 1(\text{V}/\text{V})$. Hence, the transient response lasts for a period of time equal to approximately $5RC$. For a constant voltage applied to the circuit, that is, $e_0(t) = E_0$, $t \geq 0$, the steady-state voltage $v_c(\infty)$ is numerically equal to E_0 since $v_c(\infty) = KE_0 = 1 \cdot E_0$.

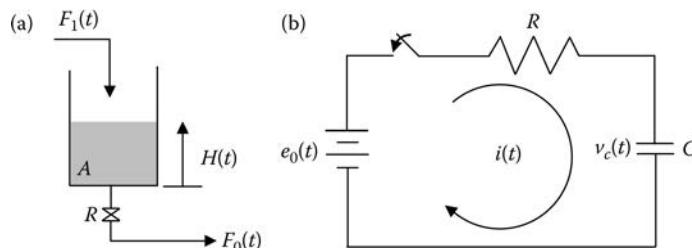


FIGURE 2.2 Examples of systems with first-order system models: (a) storage tank and (b) RC circuit.

The step response is obtained from Equation 2.6 with $y(0) = v_c(0) = 0$, $\tau = RC$, $K = 1$, and $A = E_0$. The result is

$$v_c(t) = E_0(1 - e^{-t/RC}), \quad t \geq 0 \quad (2.12)$$

The step response consists of the steady-state component

$$v_c(\infty) = E_0 \quad (2.13)$$

and the transient component

$$v_c(t) = -E_0 e^{-t/RC}, \quad t \geq 0 \quad (2.14)$$

The transient response involves the exponential $e^{-t/RC}$, which is called the natural mode of the system. To understand this, consider the circuit response with zero applied voltage ($E_0 = 0$) and a nonzero initial voltage across the capacitor $v_c(0)$. From Equation 2.6, the solution for $v_c(t)$ is

$$v_c(t) = v_c(0)e^{-t/RC}, \quad t \geq 0 \quad (2.15)$$

a constant times the natural mode. Natural modes of linear systems are exponential functions of time involving the parameters of the system, in this case, R and C . The natural modes do not depend on the system inputs. The unforced response of higher-order system models is referred to as the natural response of the system. It contains a linear combination of the natural modes (only one for the first-order system model). In general, the natural modes of linear system models appear in the transient response independent of whether the system is being forced (excited by inputs) or simply responding to initial conditions as in the case of an autonomous system.

EXAMPLE 2.1

A 12 V battery is used to charge the capacitor in the circuit shown in Figure 2.2. When the switch is closed at $t = 0$, the capacitor voltage is zero. Numerical values of the circuit parameters are $R = 5000 \Omega$ and $C = 0.125 \times 10^{-6} \text{ F}$ (1 F = 1 A per V/s).

- a. Find the time constant τ , steady-state gain K , and natural mode of the circuit.
 - b. Find the steady-state voltage $v_c(\infty)$ across the capacitor.
 - c. Determine how long it takes for the capacitor to charge up to 50% of $v_c(\infty)$.
 - d. Find and graph the transient component, steady-state component, and the complete response for the case where the capacitor is initially charged to 3 V.
- a. $\tau = RC = (5000 \Omega) \times 0.125 \times 10^{-6} \text{ F} = 0.000625 \text{ s} (625 \times 10^{-6} \text{ s})$
 $K = 1 \text{ V/V}$
Natural mode: $e^{-t/RC} = e^{-t/0.000625} = e^{-t/625 \times 10^{-6}}$, $t \geq 0$
- b. $v_c(\infty) = KE_0 = (1 \text{ V/V}) \times 12 \text{ V} = 12 \text{ V}$
- c. $v_c(t) = E_0(1 - e^{-t/RC}) \Rightarrow 6 = 12(1 - e^{-t/625 \times 10^{-6}})$, which can be solved using natural logarithms to give $t = 0.0004332 \text{ s}$
- d. From Equation 2.6 with initial condition $v_c(0) = 3 \text{ V}$, the complete response is

$$\begin{aligned} v_c(t) &= v_c(0)e^{-t/RC} + KE_0(1 - e^{-t/RC}), \quad t \geq 0 \\ &= 3e^{-t/625 \times 10^{-6}} + (1)(12)(1 - e^{-t/625 \times 10^{-6}}) \end{aligned}$$

The transient component is

$$\begin{aligned} V_c(t)_{tr} &= [v_c(0) - KE_0]e^{-t/RC}, \quad t \geq 0 \\ &= [3 - (1)(12)]e^{-t/625 \times 10^{-6}} \\ &= -9e^{-t/625 \times 10^{-6}} \end{aligned}$$

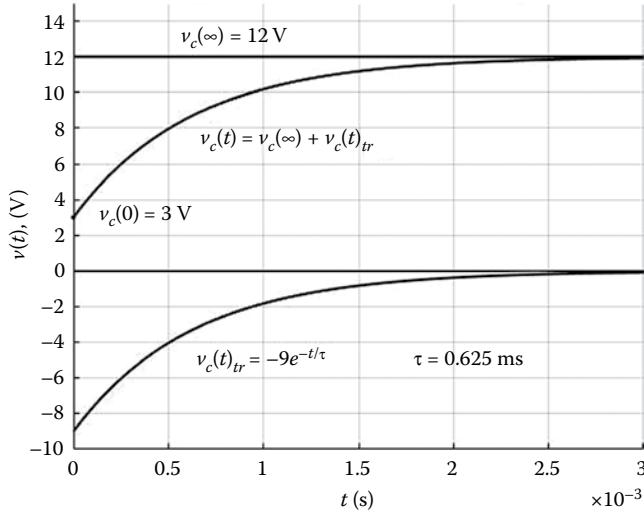


FIGURE 2.3 Steady-state, transient, and total response of an RC circuit.

and the steady-state component is

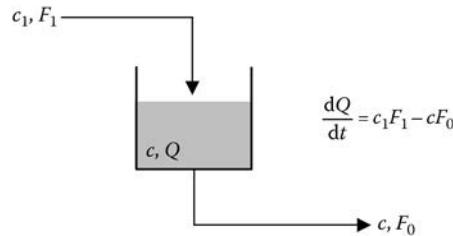
$$v_c(\infty) = KE = 1 \frac{V}{\sqrt{V}} (12 \text{ V}) = 12 \text{ V}$$

Graphs of the steady-state, transient, and complete responses are shown in [Figure 2.3](#).

Note that the transient response has decayed to essentially zero after five time constants ($5 \times 625 \times 10^{-6} = 3.125 \times 10^{-3}$) have elapsed.

EXERCISES

- 2.1 The tank shown in [Figure 2.2](#) has a constant cross-sectional area A and fluid resistance R .
 - a. Find expressions for the time constant τ and steady-state gain K of the tank in terms of the physical parameters A and R .
 - b. The empty tank is subject to a constant flow in of \bar{F} ft³/min. Obtain an expression for the liquid level step response of the tank.
 - c. The cross-sectional area of the tank is 20 ft², and the fluid resistance is 0.5 ft per ft³/min. How high must the tank be if the inflow is constant at $\bar{F} = 15$ ft³/min for it not to overflow.
 - d. How long will it take for the tank level to reach 50% of its final height?
 - e. What size tank is needed if the time required to fill up is increased by 10%?
- 2.2 Consider the first-order system: $(d/dt)y(t) + a_0y(t) = b_0u(t)$
 - a. Under what conditions does this system reduce to a pure integrator?
 - b. For the continuous-time integrator in part (a), express the output $y(t)$ in terms of the input $u(t)$. Assume the initial condition is $y(0) = y_0$.
 - c. When is a liquid storage tank a pure integrator?
- 2.3 The amount of salt Q in a well-stirred tank shown in the [Figure E2.3](#) depends on c_1 , the concentration of salt in the brine solution entering the tank, as well as the flow rates F_1 and F_0 into and out of the tank. The continuous-time model is based on conservation of salt. It equates dQ/dt , the instantaneous rate of change in the amount of salt in the tank to the difference in the rate of salt entering the tank, c_1F_1 , and the rate of salt flowing out of the tank, cF_0 .

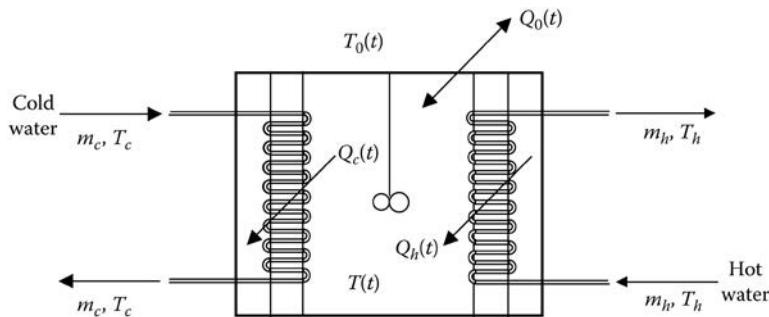
**FIGURE E2.3**

The tank initially contains 100 lb of salt-free water. The concentration of salt in the brine solution flowing in is 0.25 lb/ft³. Both the flow into and the flow out of the tank are both 1 ft³/min. Note that 1 ft³ of water weighs approximately 62.4 lb.

- Find $Q(t)$, the amount of salt in the tank as a function of time.
- Find the amount of salt in the tank at steady state.

$$\frac{dQ}{dt} = c_l F_l - c F_0$$

2.4 A temperature-controlled chamber is shown in Figure E2.4.

**FIGURE E2.4**

The air temperature inside the chamber is assumed to be the same everywhere, namely, $T(t)$. The chamber walls are insulated to reduce heat loss or gain with its surroundings. Temperature control is achieved by circulating hot or cold water through pipes located inside the chamber. Heat exchange occurs between the air inside the chamber and the circulating water in the pipes. The heat flow from the circulating hot water is $Q_h(t)$, and $Q_c(t)$ is the heat flow to the cold water. Heat exchange $Q_0(t)$ also occurs between the air inside and outside the chamber. Ambient temperature outside the chamber is denoted $T_0(t)$.

A suitable model for this thermal system is based on the conservation of energy.

$$c_A V \frac{dT}{dt} = Q_h - Q_c - Q_0$$

V is the volume (ft³) of air in the chamber, and c_A is the thermal capacitance of air (0.01375 Btu/°F/ft³). The heat flow terms on the right-hand side are given by

$$Q_h = \dot{m}_h c_p (T_h - T)$$

$$Q_c = \dot{m}_c c_p (T - T_c)$$

$$Q_0 = \frac{1}{R} (T - T_0)$$

where

- \dot{m}_h and \dot{m}_c are the mass flow rates (lb/min) of the hot and cold water
- c_p is the specific heat of water (1 Btu/lb/°F)
- R is the thermal resistance (°F/Btu/min) of the chamber walls

The expressions for Q_h and Q_c assume that the flow rates of the circulating fluids are great enough that both fluids exit at the same temperature at which they entered the chamber.

- a. Express the mathematical model in the form of a differential equation relating the output T and its derivative to the inputs T_h , T_c and T_0 .
- b. Find the time constant and the three steady-state gains of the system. Check the units to verify that the time constant is in minutes and the steady-state gains are dimensionless (°F/°F).
- c. Show that the air temperatures inside and outside the chamber eventually equalize after both the hot and cold circulating water flows are turned off.
- d. Suppose the chamber air temperature is required to be higher than the outside ambient air temperature, which remains constant, that is, $T_0(t) = \bar{T}_0$, $t \geq 0$. The hot water temperature entering the chamber is three times greater than the ambient temperature. The initial air temperature inside the chamber is the same as the outside ambient temperature. Find the analytical solution for $T(t)$, $t \geq 0$, the air temperature inside the chamber.
- e. Graph the solution for $T(t)$, $t \geq 0$ in part (d) using the following values:

$$V = 5000 \text{ ft}^3, R = 0.025 \text{ °F/Btu/min}, \dot{m}_h = 50 \text{ lb/min}, \text{ and } \bar{T}_0 = 60 \text{ °F}$$

2.3 SECOND-ORDER SYSTEMS

Input-output models of continuous-time dynamic systems where the highest derivative of the dependent variable is second order are classified as second-order systems. Second-order systems result when there are two energy storage elements present. Our interest for now is in linear second-order systems, which can be manipulated into the form shown in Equation 2.16 relating an output $y(t)$ to an input $u(t)$ involving generic system parameters, ζ , ω_n , and K .

$$\frac{d^2}{dt^2}y(t) + 2\zeta\omega_n \frac{d}{dt}y(t) + \omega_n^2 y(t) = K\omega_n^2 u(t) \quad (2.16)$$

For an actual second-order system (mechanical, electrical, biological, etc.), the generic parameters can be expressed in terms of the system's physical parameters. The importance of each will be explained shortly.

The unit step response of the second-order system is the solution for $y(t)$ in Equation 2.16 when $y(0) = 0$ and the input $u(t) = 1$, $t \geq 0$, hereafter denoted by $\hat{u}(t)$. It can be found in any text related to linear systems or controls (Palm 1983; Franklin et al. 2002; Dorf and Bishop 2005). The unit step response assumes one of three forms depending on the location of the roots of the algebraic equation

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad (2.17)$$

known as the characteristic equation of the system. The characteristic roots are the solution to Equation 2.17 and are given by

$$s_1, s_2 = -\zeta\omega_n \pm \sqrt{\zeta^2 - 1}\omega_n \quad (2.18)$$

The natural modes of the second-order system are $e^{s_1 t}$ and $e^{s_2 t}$. The step response depends on the value of the parameter ζ . There are three cases to consider.

Case 1: $\zeta > 1$

If we let $s_1 = -\zeta\omega_n - \sqrt{\zeta^2 - 1}\omega_n$ and $s_2 = -\zeta\omega_n + \sqrt{\zeta^2 - 1}\omega_n$, then both roots are negative (assuming $\omega_n > 0$) and $s_1 < s_2 < 0$. Introducing time constants τ_1 and τ_2 as the reciprocals of the characteristic roots s_1 and s_2 , respectively,

$$\tau_1 = -\frac{1}{s_1}, \quad \tau_2 = -\frac{1}{s_2} \quad (2.19)$$

The unit step response is

$$y(t) = K \left[1 + \frac{\tau_2 e^{-t/\tau_2} - \tau_2 e^{-t/\tau_1}}{\tau_1 - \tau_2} \right], \quad t \geq 0 \quad (2.20)$$

Case 2: $0 < \zeta < 1$

The characteristic roots are complex conjugates and can be expressed as

$$s_1, s_2 = -\zeta\omega_n \pm j\sqrt{1 - \zeta^2}\omega_n \quad (2.21)$$

It is convenient to define a new quantity ω_d in terms of ζ and ω_n according to

$$\omega_d = \sqrt{1 - \zeta^2}\omega_n \quad (2.22)$$

The unit step response is

$$y(t) = K \left[1 - e^{-\zeta\omega_n t} \left(\cos \omega_d t + \frac{\zeta\omega_n}{\omega_d} \sin \omega_d t \right) \right], \quad t \geq 0 \quad (2.23)$$

An alternate form of Equation 2.23 is

$$y(t) = K \left[1 - \frac{\omega_n}{\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t + \varphi) \right], \quad t \geq 0 \quad (2.24)$$

where the phase angle term φ is given by

$$\varphi = \tan^{-1} \frac{\omega_d}{\zeta\omega_n} = \tan^{-1} \left(\frac{\sqrt{1 - \zeta^2}}{\zeta} \right) \quad (2.25)$$

Case 3: $\zeta = 1$

From Equation 2.18, the characteristic roots are repeated, $s_1 = s_2 = -\omega_n$

The unit step response is

$$y(t) = K[1 - e^{-\omega_n t}(\omega_n t + 1)], \quad t \geq 0 \quad (2.26)$$

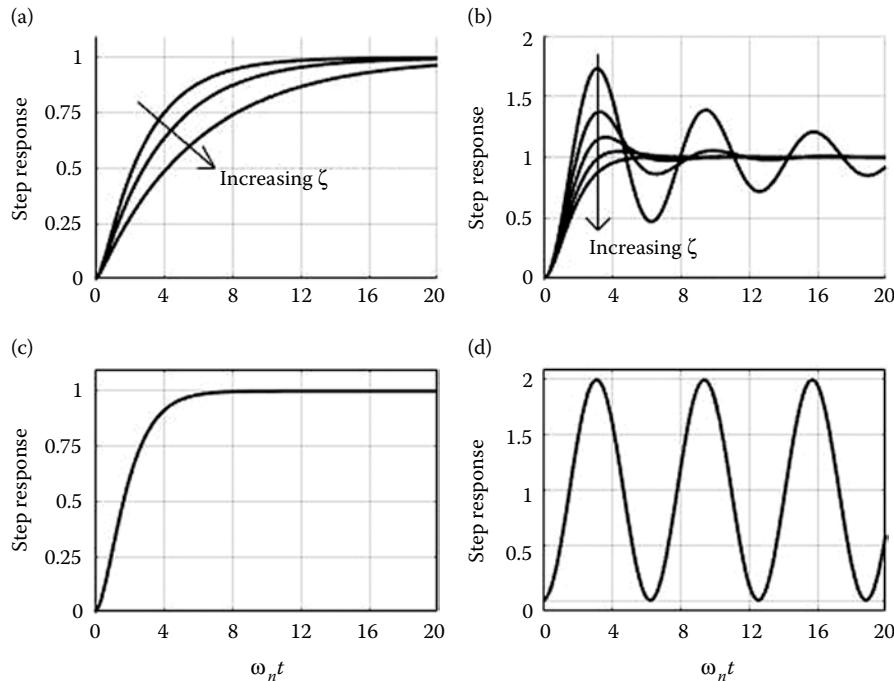


FIGURE 2.4 Unit step response of second-order system in Equation 2.16. (a) Overdamped, $\zeta = 1.5, 2, 3$. (b) Underdamped, $\zeta = 0.1, 0.3, \dots, 0.9$. (c) Critically damped, $\zeta = 1$. (d) Zero damping, $= 0$.

A graph of the unit step responses given in Equations 2.20, 2.23, and 2.26 with $K = 1$ is shown in Figure 2.4. The abscissa is $\omega_n t$, a dimensionless variable, which allows us to visualize the effect of the parameter on the step response independent of ω_n . Note that all three step responses start from zero. Furthermore, the initial slope given by $dy(0)/dt$ is also zero for all three cases (see Exercise 2.6).

There are no oscillations in Case 1 ($\zeta > 1$), that is, the response is monotonically increasing without overshooting the final value $y(\infty) = K\bar{u} = 1$ for a unit step input. The transient period increases with increasing ζ . The system is said to be overdamped.

An oscillatory step response occurs in Case 2 ($0 < \zeta < 1$), and the system is referred to as underdamped. As the value of ζ decreases, the oscillations become more pronounced, and the settling time for the transient component to die out becomes larger.

The case when $\zeta = 1$ represents the transition from Case 1 to Case 2 (or vice versa). The second-order system is called critically damped in this situation.

The graph in Figure 2.4d is the unit step response for the case when $\zeta = 0$. From Equation 2.23 with $\zeta = 0$,

$$y(t) = K(1 - \cos \omega_n t), \quad t \geq 0 \quad (2.27)$$

resulting in sustained oscillations from 0 to 2 when $K = 1$. The differential equation of the unforced system is

$$\frac{d^2}{dt^2} y(t) + \omega_n^2 y(t) = 0 \quad (2.28)$$

and the natural response resulting from the presence of initial conditions is that of harmonic motion, that is, sustained oscillations about zero at a frequency of ω_n rad/s.

Except for the case when $\zeta = 0$, the unit step response approaches the limiting or steady-state value $y(\infty) = K$, which means that K is the DC or steady-state gain of the second-order system in Equation 2.16. The parameter ζ , which determines the existence and extent of the oscillations as well as the duration of the transient response, is called the damping ratio of the system. The last two parameters ω_n and ω_d are the natural frequency and damped natural frequency of the second-order system, respectively. The first, ω_n , is the frequency of the sustained oscillations ($\zeta = 0$) in Equation 2.27, and the second, ω_d , is the frequency of the decaying oscillations ($0 < \zeta < 1$) in Equation 2.24. It follows from Equation 2.22 that $\omega_d < \omega_n$. The natural frequency ω_n is an indication of the speed of the step response (and the system in general) since the oscillatory natural modes are damped by the exponential term with time constant $1/\zeta\omega_n$ in Equation 2.23.

EXAMPLE 2.2

Figure 2.5 shows a delicate instrument placed on a table that moves as a result of a vertical force acting on it. Springs and dampers connect the table to the ground to limit the table's movement.

The combined mass of the table and instrument is m . The total stiffness of the springs is k and the total damping is c . The mechanical system is modeled by

$$m \frac{d^2}{dt^2} x(t) + c \frac{d}{dt} x(t) + kx(t) = f(t) \quad (2.29)$$

where

$x(t)$ is the displacement of the table (from its static equilibrium position)

$f(t)$ is the force acting on the platform resulting in the motion $x(t)$

- Find expressions for the steady-state gain K , the damping ratio ζ , and the natural frequency ω_n in terms of the physical parameters m , c , and k .
- Numerical values of the physical parameters are $m = 40 \text{ lb}_m$, $k = 45 \text{ lb}_f/\text{ft}$, and $c = 4 \text{ lb}_f \text{ s}/\text{ft}$. Find the response of the table when the platform is subjected to a sudden deflection due to a force of 12 lb_f .
- Graph the solution and estimate the duration of the transient.
- The instrument is not usable if it is moving faster than 0.04 ft/s . How long a period of time must pass after the force is applied before the instrument will function properly?
- Dividing Equation 2.29 by m for comparison with the standard form of a second-order system in Equation 2.16 gives

$$\frac{d^2}{dt^2} x(t) + \frac{c}{m} \frac{d}{dt} x(t) + \frac{k}{m} x(t) = \frac{1}{m} f(t) \quad (2.30)$$

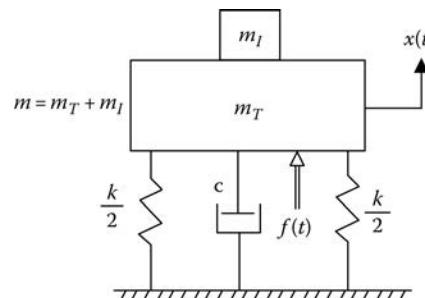


FIGURE 2.5 Mechanical system for Example 2.2.

$$\Rightarrow 2\zeta\omega_n \frac{c}{m}, \quad \omega_n^2 = \frac{k}{m}, \quad K\omega_n^2 = \frac{1}{m} \quad (2.31)$$

Solving for the parameters K , ω_n , and ζ yields

$$\omega_n = \sqrt{\frac{k}{m}}, \quad \zeta = \frac{c}{2\sqrt{km}}, \quad K = \frac{1}{k} \quad (2.32)$$

b. Substituting the given values for m (in slugs), k , and c ,

$$\begin{aligned}\omega_n &= \sqrt{\frac{k}{m}} = \sqrt{\frac{45}{40/32.2}} = 6.0187 \text{ rad/s} \\ \zeta &= \frac{c}{2\sqrt{km}} = \frac{4}{2\sqrt{45.40/32.2}} = 0.2675 \\ K &= \frac{1}{45} = 0.0222 \text{ in/lb}_f\end{aligned}$$

The damping ratio $\zeta = 0.2675$ indicates the system is underdamped. From Equation 2.22, the damped natural frequency is

$$\omega_d = \sqrt{1 - \zeta^2} \omega_n = \left(\sqrt{1 - 0.2675^2} \right) 6.0187 = 5.7994 \text{ rad/s}$$

and the response to a step input $f(t) = \bar{F} = 12 \text{ lb}_f$, $t \geq 0$ is

$$x(t) = K\bar{F} \left[1 - e^{-\zeta\omega_n t} \left(\cos \omega_d t + \frac{\zeta\omega_n}{\omega_d} \sin \omega_d t \right) \right], \quad t \geq 0 \quad (2.33)$$

Substituting the numerical values for K , \bar{F} , ζ , ω_n , and ω_d results in

$$x(t) = 0.2667 [1 - e^{-1.6100t} (\cos 5.7994t + 0.2776 \sin 5.7994t)], \quad t \geq 0 \quad (2.34)$$

c. A graph of the step response is generated in the script file “*Chap2_Ex3_1.m*” and shown in [Figure 2.6](#).

The transient period can be approximated from the graph as roughly 3 s, or it can be computed from the time constant of the exponential envelope as

$$\text{Transient period} \approx 5 \times \frac{1}{\zeta\omega_n} = 5 \times \frac{1}{0.2675(6.0187)} = 3.1056 \text{ s}$$

d. The first derivative is obtained by differentiation of the underdamped step response in Equation 2.24. The result is

$$\frac{d}{dt} y(t) = K \frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin \omega_d t, \quad t \geq 0 \quad (2.35)$$

Substituting the numerical values for the system parameters K , ζ , ω_n , and ω_d gives

$$\frac{d}{dt} y(t) = 0.1388 e^{-1.6100t} \sin 5.7994t, \quad t \geq 0 \quad (2.36)$$

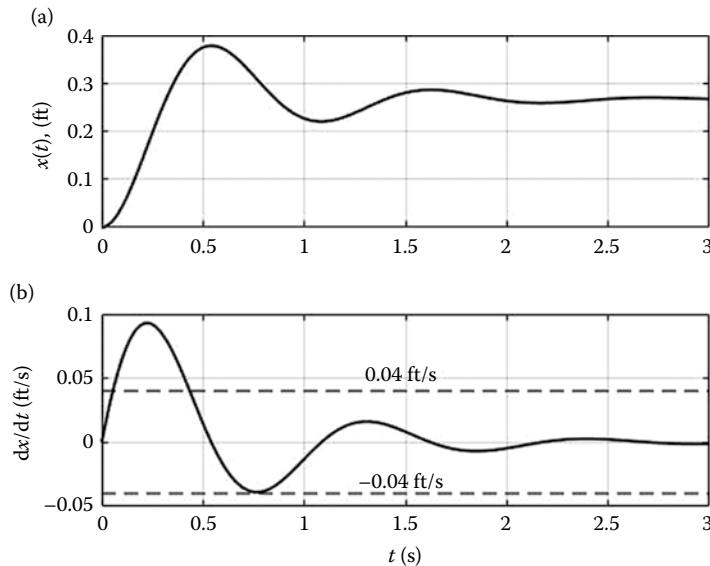


FIGURE 2.6 (a) Position and (b) velocity response of table and instrument ($\bar{F} = 12 \text{ lb}_p$).

The first derivative is graphed in the lower half of Figure 2.6. From the graph, it appears that approximately 0.77 s must elapse for the instrument to be usable, that is, the instrument is moving at less than 0.04 ft/s in either direction after that period of time. (A closeup of the response in the neighborhood of $\dot{x}/dt = -0.04 \text{ ft/s}$ reveals that the instrument's velocity actually falls a bit short of -0.04 ft/s .)

2.3.1 CONVERSION OF TWO FIRST-ORDER EQUATIONS TO A SECOND-ORDER MODEL

A linear second-order system is sometimes represented as a system of two first-order differential equations like those in Equations 2.37 and 2.38:

$$\frac{dx}{dt} = ax + by + f(t) \quad (2.37)$$

$$\frac{dy}{dt} = cx + dy + g(t) \quad (2.38)$$

Suppose a single equation relating the dependent variable $x = x(t)$ and the inputs $f = f(t)$ and $g = g(t)$ is required. The first step is to solve for $y = y(t)$ in Equation 2.37,

$$y = \frac{1}{b} \left(\frac{dx}{dt} - ax - f \right) \quad (2.39)$$

Differentiating Equation 2.39,

$$\frac{dy}{dt} = \frac{1}{b} \left(\frac{d^2x}{dt^2} - a \frac{dx}{dt} - \frac{df}{dt} \right) = cx + dy + g \quad (2.40)$$

Replacing y in Equation 2.40 with Equation 2.39 gives

$$\frac{1}{b} \left(\frac{d^2x}{dt^2} - a \frac{dx}{dt} - \frac{df}{dt} \right) = cx + d \left[\frac{1}{b} \left(\frac{dx}{dt} - ax - f \right) \right] + g \quad (2.41)$$

and simplifying leads to the second-order differential equation,

$$\frac{d^2x}{dt^2} - (a+d) \frac{dx}{dt} + (ad-bc)x = \frac{df}{dt} - df + bg \quad (2.42)$$

A similar procedure is used to eliminate x from Equations 2.37 and 2.38 to give a second-order differential equation in y .

EXAMPLE 2.3

The well-mixed tanks shown in Figure 2.7 contain uniform salt concentrations of $c_1 = c_1(t)$ and $c_2 = c_2(t)$, respectively. Concentration of salt in the input to the first tank is $c = c(t)$. The flow rates between the tanks are Q_1 and Q_2 , where $Q_1 > Q_2 > 0$. The liquid volumes in both tanks remain constant at V_1 and V_2 .

- Write the differential equations for the conservation of salt in each tank.
- Find the differential equation relating $c_2(t)$ and the input $c(t)$.
- Find expressions for the damping ratio, natural frequency, and steady-state gain.
- Find and plot the step response for c_2 under the following conditions:
 $Q_1 = 10 \text{ gal/min}$, $Q_2 = 5 \text{ gal/min}$, $V_1 = 15 \text{ gal}$, and $V_2 = 15 \text{ gal}$
 $c_1(0) = c_2(0) = 0 \text{ lb of salt/gal}$, $c(t) = \bar{c} = 0.25 \text{ lb salt/gal}$, $t \geq 0$
- Equating the accumulation of salt in each tank to the difference between the rates of salt in and out of the tanks,

$$\frac{d}{dt}(c_1 V_1) = Q_{in}c + Q_2 c_2 - Q_1 c_1 \quad (2.43)$$

$$\frac{d}{dt}(c_2 V_2) = Q_1 c_1 - Q_2 c_2 - Q_{out} c_2 \quad (2.44)$$

Since the holdup of liquid in both tanks is constant, the flows Q_{in} and Q_{out} are equal,

$$Q_{in} = Q_{out} = Q_1 - Q_2 \quad (2.45)$$

And, therefore, Equations 2.43 and 2.44 become

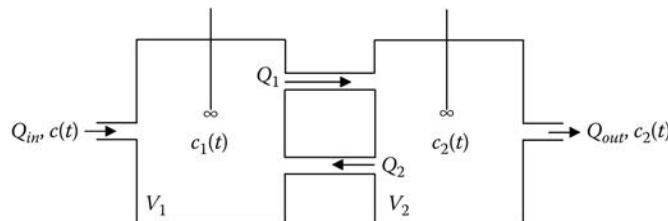


FIGURE 2.7 Two-tank mixing system.

$$V_1 \frac{dc_1}{dt} = (Q_1 - Q_2)c + Q_2 c_2 - Q_1 c_1 \quad (2.46)$$

$$V_2 \frac{dc_2}{dt} = Q_1 c_1 - Q_2 c_2 - (Q_1 - Q_2)c_2 \quad (2.47)$$

b. Rearranging Equations 2.46 and 2.47 into the form of Equations 2.37 and 2.38,

$$\frac{dc_2}{dt} = -\frac{Q_1}{V_2} c_2 + \frac{Q_1}{V_2} c_1 \quad (2.48)$$

$$\frac{dc_1}{dt} = \frac{Q_2}{V_1} c_2 - \frac{Q_1}{V_1} c_1 + \frac{(Q_1 - Q_2)}{V_1} c \quad (2.49)$$

Thinking of c_2 as x and c_1 as y , and comparing Equations 2.48 and 2.49 with Equations 2.37 and 2.38 implies

$$a = -\frac{Q_1}{V_2}, b = \frac{Q_1}{V_2}, c = \frac{Q_2}{V_1}, d = -\frac{Q_1}{V_1}, f(t) = 0, g(t) = \frac{(Q_1 - Q_2)}{V_1} c(t) \quad (2.50)$$

From Equation 2.42, the second-order differential equation relating c_2 and c is

$$\frac{d^2 c_2}{dt^2} + Q_1 \left(\frac{1}{V_1} + \frac{1}{V_2} \right) \frac{dc_2}{dt} + \frac{Q_1(Q_1 - Q_2)}{V_1 V_2} c_2 = \frac{Q_1(Q_1 - Q_2)}{V_1 V_2} c \quad (2.51)$$

c. Comparing the left-hand side of Equation 2.51 with the standard form in Equation 2.16 gives

$$2\zeta\omega_n = Q_1 \left(\frac{1}{V_1} + \frac{1}{V_2} \right), \quad \omega_n^2 = \frac{Q_1(Q_1 - Q_2)}{V_1 V_2} \quad (2.52)$$

$$\Rightarrow \omega_n = \left[\frac{Q_1(Q_1 - Q_2)}{V_1 V_2} \right]^{1/2}, \quad \zeta = \frac{(V_1 + V_2)}{2} \left[\frac{Q_1}{(Q_1 - Q_2)V_1 V_2} \right]^{1/2} \quad (2.53)$$

For $c(t) = \bar{c}$, the steady-state value of c_2 is obtained from Equation 2.51 by setting the derivatives equal to zero resulting in

$$\frac{Q_1(Q_1 - Q_2)}{V_1 V_2} (c_2)_{ss} = \frac{(Q_1 - Q_2)}{V_1} \left(\frac{Q_1}{V_2} \bar{c} \right) \quad (2.54)$$

$$\Rightarrow (c_2)_{ss} = \bar{c} \quad (2.55)$$

Hence, the steady-state gain $K = 1$ lb salt/lb salt as expected.

d. For the given conditions, that is, $Q_2 = Q = 5$, $Q_1 = 2Q = 10$, and $V_1 = V_2 = V = 15$

$$\omega_n = \left[\frac{2Q(2Q - Q)}{VV} \right]^{1/2} = (2)^{1/2} \frac{Q}{V} = (2)^{1/2} \frac{5}{15} = 0.4714 \text{ rad/min} \quad (2.56)$$

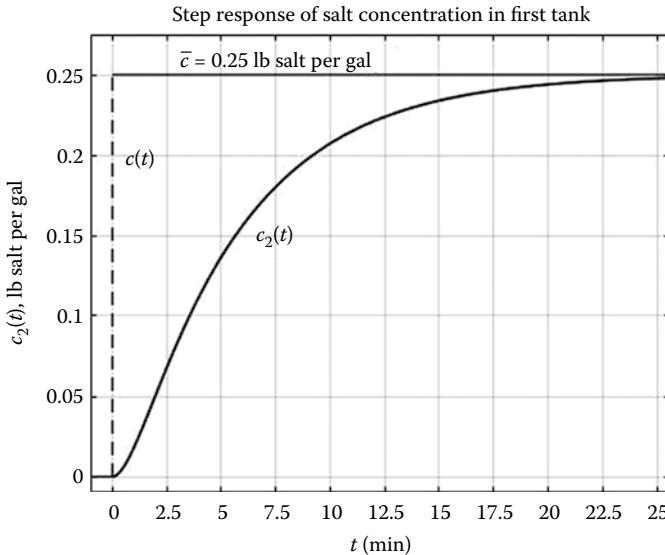


FIGURE 2.8 Response of salt concentration in second tank to step input $c(t) = 0.25$, $t \geq 0$.

$$\zeta = \frac{(V + V)}{2} \left[\frac{2Q}{(2Q - Q)V} \right]^{1/2} = (2)^{1/2} = 1.4142 \quad (2.57)$$

From Equation 2.18, the characteristic roots of the overdamped system are

$$s_1, s_2 = -\zeta\omega_n \pm \sqrt{\zeta^2 - 1\omega_n} = -1.1381 \text{ rad/min}, -0.1953 \text{ rad/min} \quad (2.58)$$

The time constants in Equation 2.19 are $\tau_1 = -1/s_1 = 0.8787$ min and $\tau_2 = -1/s_2 = 5.1213$ min, and from the unit step response in Equation 2.20, the response to a step of magnitude \bar{c} is

$$c_2(t) = K\bar{c} \left[1 + \frac{\tau_2 e^{-t/\tau_2} - \tau_1 e^{-t/\tau_1}}{\tau_1 - \tau_2} \right] \quad (2.59)$$

$$= 0.25 \left[1 - \left(\frac{5.1213 e^{-t/5.1213} - 0.8787 e^{-t/0.8787}}{4.2426} \right) \right], \quad t \geq 0 \quad (2.60)$$

The second-order differential equation in Equation 2.51 is in standard form; however, the second-order differential equation for $c_1(t)$ contains the first derivative dc/dt on the right-hand side of the equation (see Exercise 2.6). The implication of input derivatives in the system model will be discussed in a later section. A graph of the step response is shown in Figure 2.8.

EXERCISES

- 2.5 Starting with Equations 2.37 and 2.38, obtain the second-order differential equation relating the output $y = y(t)$ and its derivatives to the inputs $f = f(t)$ and $g = g(t)$.
- 2.6 In Example 2.3,
- Find the differential equation relating $c_1(t)$ and the input $c(t)$.

- b. Find the step response in $c_1(t)$ for the same initial conditions, system parameters, and input $c(t)$. Graph the step response for $c_1(t)$ and $c_2(t)$.
 - c. Show that the first derivative dc_1/dt is discontinuous at $t = 0$ while the first derivative dc_2/dt is continuous at $t = 0$.
- 2.7 The two-tank system in Exercise 1.2 is second order.
- a. Convert the model of the system from two first-order differential equations to one second-order differential equation with input $F_1(t)$ and output $H_2(t)$.
 - b. Find expressions for the damping ratio, natural frequency, and steady-state gain in terms of the physical parameters A_1 , A_2 , R_1 , and R_2 .
 - c. Use the results from part (b) to express the damping ratio in terms of the tank time constants $\tau_1 = A_1 R_1$ and $\tau_2 = A_2 R_2$.
 - d. Show that the system can never be underdamped.
- For parts (e) and (f), assume the following values for the system parameters:
- $$A_1 = 100 \text{ ft}^2, R_1 = 0.25 \text{ ft per ft}^3/\text{min}, A_2 = 50 \text{ ft}^2, \text{ and } R^2 = 0.1 \text{ ft per ft}^3/\text{min}$$
- e. Find and graph the response $H_2(t)$ of the unforced system, that is, $F_1(t) = 0$, $t \geq 0$ starting from $H_1(0) = 40$ ft and $H_2(0) = 0$ ft.
 - f. Find and graph the step response of $H_2(t)$ when $F_1(t) = 75$ ft $^3/\text{min}$. Both tanks are initially empty. Does the first tank achieve steady state in roughly $5\tau_1$? Does the second tank achieve steady state in roughly $5\tau_2$? Explain.
- 2.8 A fundamental difference between the step response of first- and second-order linear systems in standard form is the initial rate of change, that is, the first derivative at $t = 0$.
- a. Show that the first-order system step response undergoes the maximum rate of change at $t = 0$.
 - b. Show that the initial derivative of the second-order system step response is zero regardless of whether the system is underdamped, critically damped, or overdamped.

2.4 SIMULATION DIAGRAMS

In many cases, dynamic systems are composed of individual components and subsystems. The relationship of a system's components to each other and the role they serve in the overall system design are oftentimes easier to comprehend when presented in visual form rather than by inspection of the mathematical models. Control systems for ground vehicles, aircraft, robotic devices, building environments, and so forth are typically presented in graphical form as block diagrams. The blocks are both static and dynamic depending on the component it represents. Modern continuous-time system simulation languages include extensive libraries of special purpose blocks to represent the dynamics of commonly occurring components.

It is useful to reduce the blocks in a block diagram of a continuous-time dynamic system to a level that exposes the pure integrators. The simulationist is then given the flexibility of approximating individual integrators using different numerical algorithms. This is especially useful in applications where simulation code is developed manually instead of relying on a general purpose simulation language. This point will be revisited in [Chapter 3](#) following a discussion of numerical integration.

A block diagram of a continuous-time dynamic system comprising algebraic blocks and integrators is referred to as a simulation diagram. We begin with the first-order system of Equation 2.61:

$$\frac{dy(t)}{dt} + a_0 y(t) = b_1 \frac{du(t)}{dt} + b_0 u(t) \quad (2.61)$$

Equation 2.61 is a more general form than the first-order models introduced in Section 2.2 due to the presence of the first derivative term on the right-hand side.

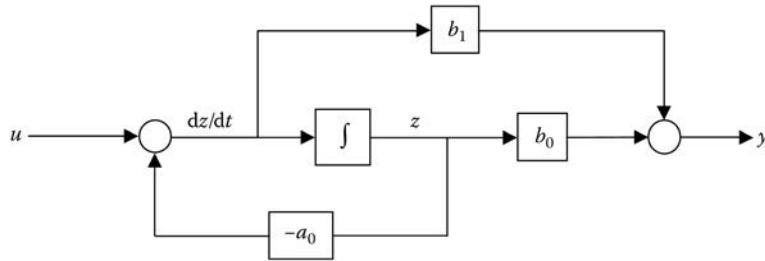


FIGURE 2.9 Simulation diagram of first-order system: $(d/dt)y(t) + a_0y(t) = b_1(d/dt)u(t) + b_0u(t)$.

If we introduce a new variable $z = z(t)$ where

$$\frac{d}{dt}z(t) + a_0z(t) = u(t) \quad (2.62)$$

the output y is related to z by

$$y(t) = b_0z(t) + b_1 \frac{d}{dt}z(t) \quad (2.63)$$

It is left as an exercise to show that Equations 2.62 and 2.63 are equivalent to Equation 2.61. In addition to the blocks required to implement Equations 2.62 and 2.63, an integrator block is needed to integrate the first derivative dz/dt to generate $z(t)$, that is,

$$z(t) = \int \frac{dz}{dt} dt \quad (2.64)$$

The simulation diagram in [Figure 2.9](#) is constructed by first drawing an integrator block and labeling the input dz/dt and output z corresponding to Equation 2.64. Next, we solve for the derivative term dz/dt in Equation 2.62 and draw a portion of the diagram to implement the result. Finally, the output y is generated from Equation 2.63 using the b_0 and b_1 gain blocks and a summing block.

The simulation diagram representation of the first-order system's dynamics involves a single dynamic block, namely, the integrator. The remaining blocks are sum blocks and gains that are algebraic in nature.

A block diagram for the same first-order system is shown in [Figure 2.10](#). The block diagram is a direct implementation of Equation 2.61 after solving for the first derivative dy/dt . An additional variable z is not required in this case. The diagram in [Figure 2.10](#) is not a simulation diagram because of the presence of the differentiator. In digital simulation, the differentiator (like the integrator) must be implemented using a numerical approximation. Numerical methods for approximating the

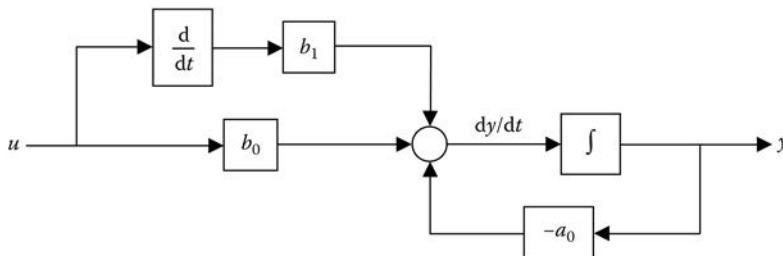


FIGURE 2.10 Block diagram of first-order system: $(d/dt)y(t) + a_0y(t) = b_1(d/dt)u(t) + b_0u(t)$.

derivative of a continuous-time function are available. However, they are rarely implemented in simulation applications due to their sensitivity to high-frequency noise components often present in continuous-time signals.

A final observation relates to the special case when b_1 in Equation 2.61 is zero. The input derivative is absent, and the first-order system assumes the simpler form of Equation 2.3 or 2.4. Recall that this form was sufficient to model the dynamics of the linear tank in [Chapter 1](#) and the simple RC circuit of Example 2.1.

EXAMPLE 2.4

Draw a simulation diagram of the linear tank modeled by

$$A \frac{d}{dt} H(t) + \frac{1}{R} H(t) = F_1(t) \quad (2.65)$$

The diagram is shown in [Figure 2.11](#).

Dividing Equation 2.65 by the parameter A and comparing the result to Equation 2.61 show

$$a_0 = \frac{1}{AR}, \quad b_0 = \frac{1}{A}, \quad b_1 = 0$$

leading to the simulation diagram shown in [Figure 2.11](#).

The simulation diagram in [Figure 2.11](#) is not unique. The “ $1/A$ ” block can be moved from the location where z is its input to the left of the summer where F_1 becomes its input. In that case, z and H are identical. The alternate simulation diagram can be obtained directly by solving the differential equation of the tank for the first derivative,

$$\frac{d}{dt} H(t) = \frac{1}{A} \left[F_1(t) - \frac{1}{R} H(t) \right] \quad (2.66)$$

and implementing Equation 2.66 directly. Integrating the derivative dH/dt to get H completes the diagram.

EXAMPLE 2.5

Suppose the current $i(t)$ in the RC circuit of [Figure 2.2](#) is considered the output. The differential equation for the circuit becomes

$$\frac{d}{dt} i(t) + \frac{1}{RC} i(t) = \frac{1}{R} \frac{d}{dt} e_0(t) \quad (2.67)$$

Draw the simulation diagram for the circuit described by Equation 2.67.

From Equation 2.61, a_0 , b_0 , and b_1 are

$$a_0 = \frac{1}{RC}, \quad b_0 = 0, \quad b_1 = \frac{1}{R}$$

and the simulation diagram is drawn in [Figure 2.12](#).

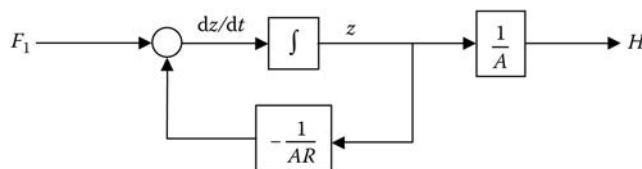


FIGURE 2.11 Simulation diagram of linear tank: $A(d/dt)H(t) + (1/R)H(t) = F_1(t)$.

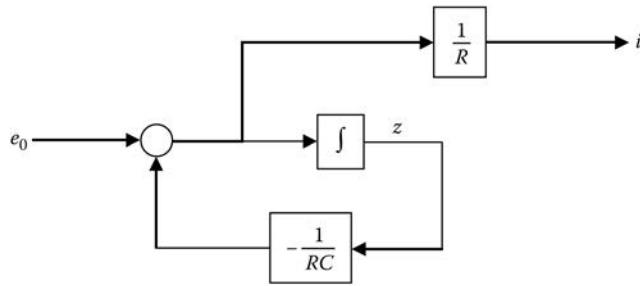


FIGURE 2.12 Simulation diagram for an RC circuit: $(d/dt)i(t) + (1/R)C)i(t) = (1/R)(d/dt)e_0(t)$.

When the differential equation model of a first-order system contains a term involving the first derivative of the input, a direct link or coupling exists from the input directly to the output. In other words, when $b_1 \neq 0$ in Equation 2.61, sudden changes in the input are immediately reflected in the output. Notice the path of heavy solid lines in Figure 2.12 illustrating the direct connection of algebraic components between the voltage input $e_0(t)$ and the output current $i(t)$.

In contrast, there is no direct connection from input to output in the simulation diagram shown in Figure 2.11 for the linear tank model. This is expected since changes in the inflow $F_1(t)$ must work their way through the tank dynamics, that is, the integrator, prior to affecting the output level $H(t)$. Hence, the tank prevents abrupt changes like a step or other inputs with high-frequency components from immediately causing any significant changes in the output $H(t)$. The tank behaves like a low-pass filter (see Exercise 1.10).

Obtaining a simulation diagram for a second-order system in the standard form

$$\frac{d^2}{dt^2}y(t) + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y(t) = K\omega_n^2 u(t) \quad (2.68)$$

is straightforward. We begin by drawing two consecutive integrators, labeling the input and output of the first with d^2y/dt^2 and dy/dt , respectively. The second integrator integrates the first derivative dy/dt producing y and is labeled accordingly. The next step is to solve for the second derivative term in Equation 2.68 resulting in

$$\frac{d^2}{dt^2}y(t) = K\omega_n^2 u(t) - 2\zeta\omega_n \frac{dy}{dt} - \omega_n^2 y(t) \quad (2.69)$$

Algebraic blocks (gains and summers) are used to implement Equation 2.69 leading to the simulation diagram shown in Figure 2.13.

The simulation diagram for a second-order system with first- or second-order derivatives of the input appearing in the differential equation model is not as straightforward. Starting with Equation 2.70

$$\frac{d^2}{dt^2}y(t) + a_1 \frac{dy}{dt} + a_0 y(t) = b_0 u(t) + b_1 \frac{du}{dt} + b_2 \frac{d^2}{dt^2}u(t) \quad (2.70)$$

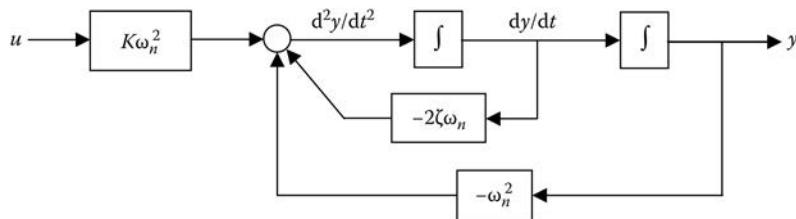


FIGURE 2.13 Simulation diagram of a second-order system in standard form.

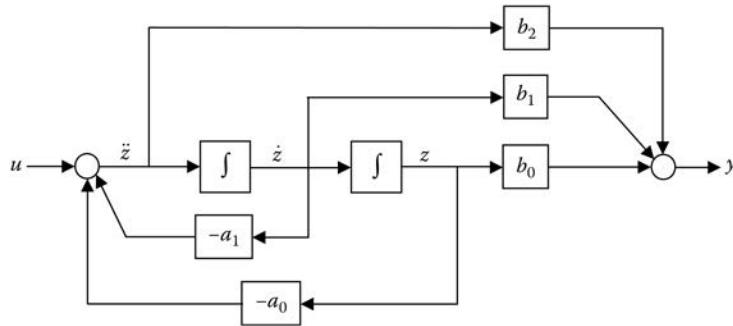


FIGURE 2.14 Simulation diagram for a second-order system with input derivatives present.

an approach similar to the method used for first-order systems with an input derivative term present is employed. An artificial variable $z(t)$ is introduced, and the output $y(t)$ is expressed as a linear combination of $z(t)$ and its two derivatives. The result is

$$\frac{d^2}{dt^2} z(t) + a_1 \frac{d}{dt} z(t) + a_0 z(t) = u(t) \quad (2.71)$$

$$y(t) = b_0 z(t) + b_1 \frac{d}{dt} z(t) + b_2 \frac{d^2}{dt^2} z(t) \quad (2.72)$$

The simulation diagram of the second-order system in Equation 2.70 is shown in Figure 2.14. Note the use of the dot notation, short for differentiation with respect to time. It is clear that a direct connection from the input $u(t)$ to the output $y(t)$ exists only when b_2 , the coefficient of the input second derivative in Equation 2.70, is nonzero.

Looking at the simulation diagrams in Figures 2.9 and 2.14 for the first- and second-order systems in Equations 2.61 and 2.70, a general pattern emerges for creating the simulation diagram of an n th-order system modeled by

$$\frac{dy^n}{dt^n} + a_{n-1} \frac{dy^{n-1}}{dt^{n-1}} + \cdots + a_1 \frac{dy}{dt} + a_0 y = b_0 u + b_1 \frac{du}{dt} + \cdots + b_{n-1} \frac{du^{n-1}}{dt^{n-1}} + b_n \frac{du^n}{dt^n} \quad (2.73)$$

The two equations equivalent to Equation 2.73 are

$$\frac{dz^n}{dt^n} + a_{n-1} \frac{dz^{n-1}}{dt^{n-1}} + \cdots + a_1 \frac{dz}{dt} + a_0 z = u \quad (2.74)$$

$$y = b_0 z + b_1 \frac{dz}{dt} + \cdots + b_{n-1} \frac{dz^{n-1}}{dt^{n-1}} + b_n \frac{dz^n}{dt^n} \quad (2.75)$$

The simulation diagram follows directly from Equations 2.74 and 2.75.

EXAMPLE 2.6

A unicycle is traveling over an uneven road as shown in Figure 2.15.

The input is the road elevation $x_r(t)$ above some reference. The output is the vertical movement $x(t)$ of the rider and seat combination (with respect to its equilibrium position). Ignoring the compliance

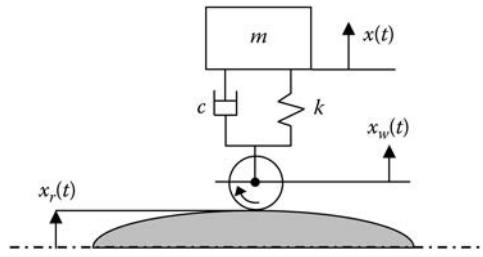


FIGURE 2.15 Unicycle traveling along an uneven road surface.

of the tire makes the wheel deflection $x_w(t) = x_r(t)$. Assume that the wheel remains in contact with the road surface. The mass of the rider and seat is m , and c and k are suspension parameters.

- Find the differential equation relating the output $x(t)$ and input $x_r(t)$.
- Draw a simulation diagram of the system.
- Is there a direct coupling between the input and output? Explain.
- The differential equation is obtained by equating the sum of the suspension forces acting on the rider and seat to the product of its mass and acceleration.

$$m \frac{d^2}{dt^2} x(t) = c \left[\frac{d}{dt} x_w(t) - \frac{d}{dt} x(t) \right] + k[x_w(t) - x(t)] \quad (2.76)$$

Replacing $x_w(t)$ with $x_r(t)$ gives

$$m \frac{d^2}{dt^2} x(t) = c \left[\frac{d}{dt} x_r(t) - \frac{d}{dt} x(t) \right] + k[x_r(t) - x(t)] \quad (2.77)$$

- Rearranging terms in Equation 2.77 gives

$$\frac{d^2}{dt^2} x(t) + \frac{c}{m} \frac{d}{dt} x(t) + \frac{k}{m} x(t) = \frac{k}{m} x_r(t) + \frac{c}{m} \frac{d}{dt} x_r(t) \quad (2.78)$$

Comparing Equations 2.78 and 2.70 leads to expressions for a_0 , a_1 , b_0 , b_1 , and b_2 in terms of the system parameters,

$$a_0 = \frac{k}{m}, \quad a_1 = \frac{c}{m}, \quad b_0 = \frac{k}{m}, \quad b_1 = \frac{c}{m}, \quad b_2 = 0 \quad (2.79)$$

and eventually the simulation diagram shown in [Figure 2.16](#).

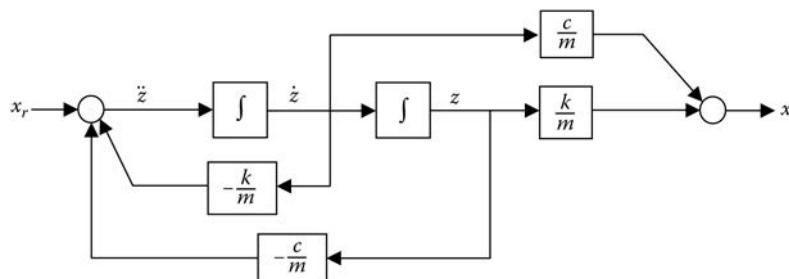


FIGURE 2.16 Simulation diagram for a unicycle suspension.

- c. Since both paths from x_r to x contain an integrator, there is no direct coupling between input and output. Consequently, an abrupt change in x_r , such as a vertical jump in the road surface height does not result in a similar type of displacement of the rider and seat combination.

2.4.1 SYSTEMS OF EQUATIONS

System models can assume the form of coupled differential and algebraic equations. The simulation diagram representation is straightforward.

EXAMPLE 2.7

A two-room building with temperatures $T_1(t)$ and $T_2(t)$ is shown in [Figure 2.17](#).

The simplified model relating the uniform room temperatures $T_1(t)$ and $T_2(t)$ to the heat supplied from the furnace $Q_f(t)$ and outside temperature $T_0(t)$ is based on conservation of energy. It consists of the following differential and algebraic equations:

$$C_1 \frac{d}{dt} T_1(t) = Q_f(t) - Q_1(t) - Q_{12}(t) \quad (2.80)$$

$$C_2 \frac{d}{dt} T_2(t) = Q_{12}(t) - Q_2(t) \quad (2.81)$$

$$Q_{12}(t) = \frac{T_1(t) - T_2(t)}{R_{12}} \quad (2.82)$$

$$Q_1(t) = \frac{T_1(t) - T_0(t)}{R_1} \quad (2.83)$$

$$Q_2(t) = \frac{T_2(t) - T_0(t)}{R_2} \quad (2.84)$$

where C_1 , C_2 , R_1 , R_2 , and R_{12} are thermal parameters of the system. The simulation diagram shown in [Figure 2.18](#) follows directly from Equations 2.80 through 2.84.

Combining Equations 2.80 through 2.84 and solving for the first derivatives give

$$\frac{d}{dt} T_1(t) = \frac{1}{C_1} \left[Q_f(t) - \frac{T_1(t) - T_0(t)}{R_1} - \frac{T_1(t) - T_2(t)}{R_{12}} \right] \quad (2.85)$$

$$= \frac{1}{C_1} \left[-\left(\frac{1}{R_1} + \frac{1}{R_{12}} \right) T_1(t) + \frac{1}{R_{12}} T_2(t) + \frac{1}{R_1} T_0(t) + Q_f(t) \right] \quad (2.86)$$

$$\frac{d}{dt} T_2(t) = \frac{1}{C_2} \left[\frac{T_1(t) - T_2(t)}{R_{12}} - \frac{T_2(t) - T_0(t)}{R_2} \right] \quad (2.87)$$

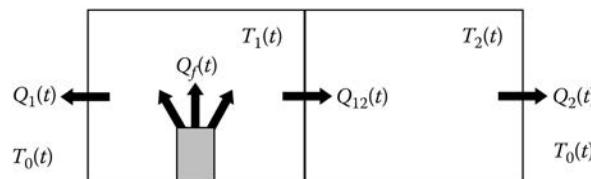


FIGURE 2.17 Heat flows and temperatures in a two-room building.

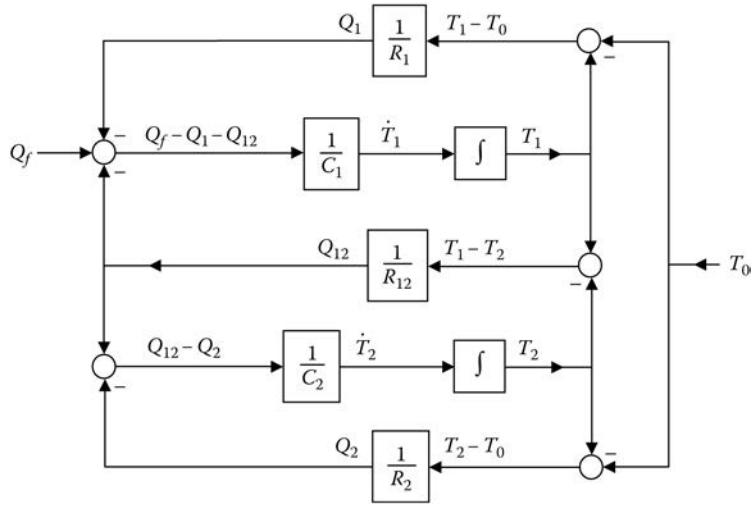


FIGURE 2.18 Simulation diagram for building room temperature model.

$$= \frac{1}{C_2} \left[\frac{1}{R_{12}} T_1(t) - \left(\frac{1}{R_2} + \frac{1}{R_{12}} \right) T_2(t) + \frac{1}{R_2} T_0(t) \right] \quad (2.88)$$

Equations 2.86 and 2.88 are of the form

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + b_{11}u_1 + b_{12}u_2 \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + b_{21}u_1 + b_{22}u_2 \end{aligned} \quad (2.89)$$

where $x_1 = T_1$, $x_2 = T_2$, $u_1 = T_0$, and $u_2 = Q_f$ and the coefficients a_{ij} and b_{ij} ($j = 1, 2$) depend on the system parameters according to

$$a_{11} = -\frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_{12}} \right), \quad a_{12} = \frac{1}{R_{12}C_1}, \quad b_{11} = \frac{1}{R_1 C_1}, \quad b_{12} = \frac{1}{C_1} \quad (2.90)$$

$$a_{21} = -\frac{1}{R_{12}C_2}, \quad a_{22} = \frac{1}{C_2} \left(\frac{1}{R_2} + \frac{1}{R_{12}} \right), \quad b_{21} = \frac{1}{R_2 C_2}, \quad b_{22} = 0 \quad (2.91)$$

Suppose we need to draw a simulation diagram for the system in Equation 2.89 with only x_1 or x_2 present. Using an approach similar to the one presented in Section 2.2 for converting two coupled first-order differential equations into a second-order differential equation, the second-order system in Equation 2.89 is equivalent to

$$\ddot{x}_1 + \alpha_1 \dot{x}_1 + \alpha_0 x_1 = \beta_{11} \dot{u}_1 + \beta_{10} u_1 + \beta_{21} \dot{u}_2 + \beta_{20} u_2 \quad (2.92)$$

where

$$\alpha_1 = -(a_{11} + a_{22}), \quad \alpha_0 = a_{11}a_{22} - a_{12}a_{21} \quad (2.93)$$

$$\beta_{11} = b_{11}, \quad \beta_{10} = a_{12}b_{21} - a_{22}b_{11}, \quad \beta_{21} = b_{12}, \quad \beta_{20} = a_{12}b_{22} - a_{22}b_{12} \quad (2.94)$$

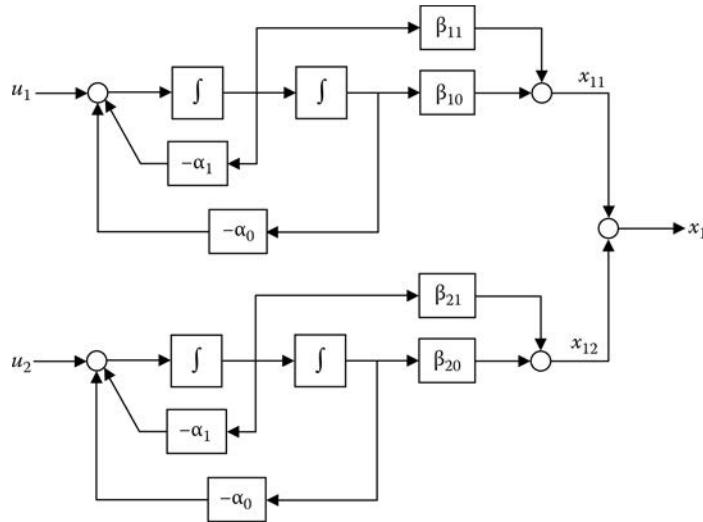


FIGURE 2.19 Simulation diagram for second-order system in Equation 2.92.

The simulation diagram for Equation 2.92 is constructed in two steps. From superposition, the output x_1 can be viewed as the sum of x_{11} and x_{12} where

$$\ddot{x}_{11} + \alpha_1 \dot{x}_{11} + \alpha_0 x_{11} = \beta_{11} \dot{u}_1 + \beta_{10} u_1 \quad (2.95)$$

$$\ddot{x}_{12} + \alpha_1 \dot{x}_{12} + \alpha_0 x_{12} = \beta_{21} \dot{u}_2 + \beta_{20} u_2 \quad (2.96)$$

Simulation diagrams for Equations 2.95 and 2.96 are drawn separately, and outputs x_{11} and x_{12} are added to yield the complete output x_1 . The result is shown in Figure 2.19.

Do not be misled into thinking that the simulation diagram shown in Figure 2.19 corresponds to a fourth-order system due to the presence of four integrators. There are two decoupled second-order systems, one with input u_1 and output x_{11} and the other with input u_2 and output x_{12} . In reality, they are the same system, that is, the second-order system governed by the second-order model in Equation 2.92.

On the other hand, if the feedback coefficients in the two systems are not identical, that is, α_0 and α_1 in both cases, the result is indeed a fourth-order system (see Exercise 2.13).

EXERCISES

2.9 Show that the system of equations

$$\frac{d}{dt}z(t) + a_0 z(t) = u(t) \quad \text{and} \quad y(t) = b_0 z(t) + b_1 \frac{d}{dt}z(t)$$

used to construct the simulation diagram for the first-order system

$$\frac{d}{dt}y(t) + a_0 y(t) = b_1 \frac{d}{dt}u(t) + b_0 u(t)$$

is equivalent to the first-order differential equation above.

Hint: The variable $z(t)$ must be eliminated from the two equations.

2.10 An alternate simulation diagram for the second-order system

$$\frac{d}{dt^2}y(t) + 2\zeta\omega_n \frac{d}{dt}y(t) + \omega_n^2 y(t) = K\omega_n^2 u(t)$$

when it is critically damped or overdamped is shown in [Figure E2.10](#) below:

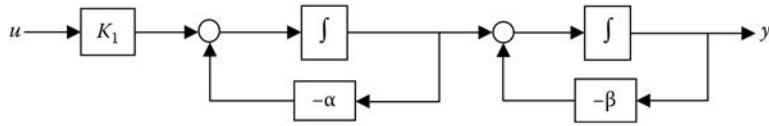


FIGURE E2.10

Find expressions for K_1 , α , and β in terms of the parameters ζ , ω_n , and K .

- 2.11 The circuit shown in [Figure E2.11](#) is governed by the differential equation:

$$RC \frac{d^2 v_C}{dt^2} + \frac{d}{dt} v_C + \frac{R}{L} v_C = \frac{d}{dt} e_S$$

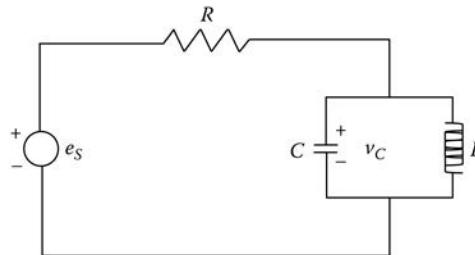


FIGURE E2.11

Draw a simulation diagram for the circuit.

- 2.12 Consider the building temperature example with room temperatures described by Equations 2.86 and 2.88.
- Find the second-order differential equation relating $T_2(t)$ and the system inputs $T_0(t)$ and $Q_f(t)$.
 - Draw a simulation diagram like the one shown in [Figure 2.19](#).
- 2.13 Simulation diagrams are shown in [Figure E2.13a](#) through [Figure E2.13c](#) below.
- Find the differential equation relating x and inputs u_1 and u_2 in [Figure E2.13a](#).
 - Find the differential equation relating x and input u in [Figure E2.13b](#).
 - Find the differential equation relating x and inputs u_1 and u_2 in [Figure E2.13c](#).
 - Comment on the differences between the systems represented by each diagram.

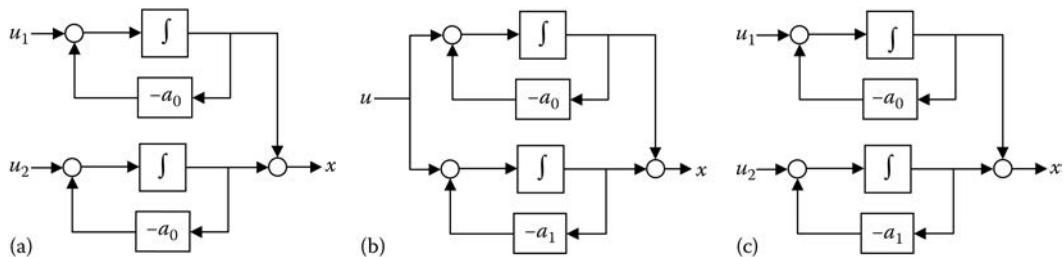


FIGURE E2.13

2.5 HIGHER-ORDER SYSTEMS

To this point, we have looked at linear continuous-time systems with first- and second-order dynamics only. Linear systems and linear controls texts include extensive coverage of lower-order

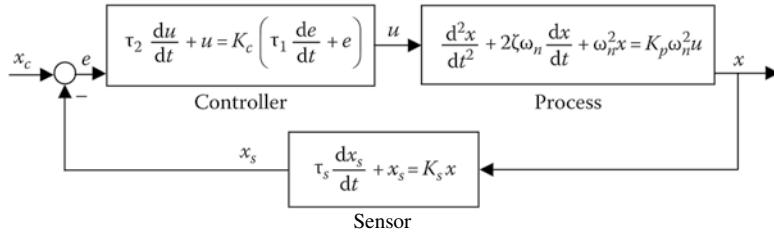


FIGURE 2.20 A control system consisting of first- and second-order components.

system response. In particular, the response of first- and second-order systems to impulse, step, and sinusoidal inputs is fully developed.

The dynamics of complex systems with linear differential equation models are invariably higher than second order. One may question why so much attention is devoted to first- and second-order systems. The explanation is simple.

High-order linear systems are oftentimes a collection of components or subsystems that are intrinsically first or second order. An electrical circuit with several capacitors and inductors is a good example. The circuit dynamics will depend on the number of these energy storage elements and their location in the circuit. In general, its order will be equal to the number of energy storage elements since each element is itself modeled as a first-order component. With n nonredundant energy storage elements, an n th-order differential equation involving an output (a voltage or current in the circuit) and an input (if an independent source is present) governs the behavior of the circuit. The same principle applies to fluid, thermal, mechanical, chemical, and so forth, systems made up of components analogous to the resistor, capacitor, and inductor of the electrical circuit.

The block diagram of a simple feedback control system is shown in Figure 2.20. The controller, process, and sensor are the subsystem components, which are individually modeled as either first or second order.

The control system model comprises the three coupled differential equations

$$\tau_2 \frac{du}{dt} + u = K_c \left(\tau_1 \frac{de}{dt} + e \right) \quad (2.97)$$

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = K_p \omega_n^2 u \quad (2.98)$$

$$\tau_s \frac{dx_s}{dt} + x_s = K_s x \quad (2.99)$$

and the summer equation

$$e = x_c - x_s \quad (2.100)$$

The command input $x_c = x_c(t)$ is the control system input, and the output of the process $x = x(t)$ is the control system output. Dependent variables $e(t)$, the error signal, $u(t)$, the output from the controller and input to the process, and $x_s(t)$, the sensor output are internal to the control system. Eliminating these variables produces a single fourth-order (1 + 2 + 1) differential equation model of the control system in the form

$$\frac{d^4x}{dt^4} + a_3 \frac{d^3x}{dt^3} + a_2 \frac{d^2x}{dt^2} + a_1 \frac{dx}{dt} + a_0 x = b_4 \frac{d^4x_c}{dt^4} + b_3 \frac{d^3x_c}{dt^3} + b_2 \frac{d^2x_c}{dt^2} + b_1 \frac{dx_c}{dt} + b_0 x_c \quad (2.101)$$

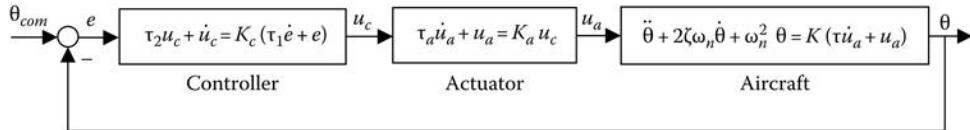


FIGURE 2.21 Control system for an aircraft pitch.

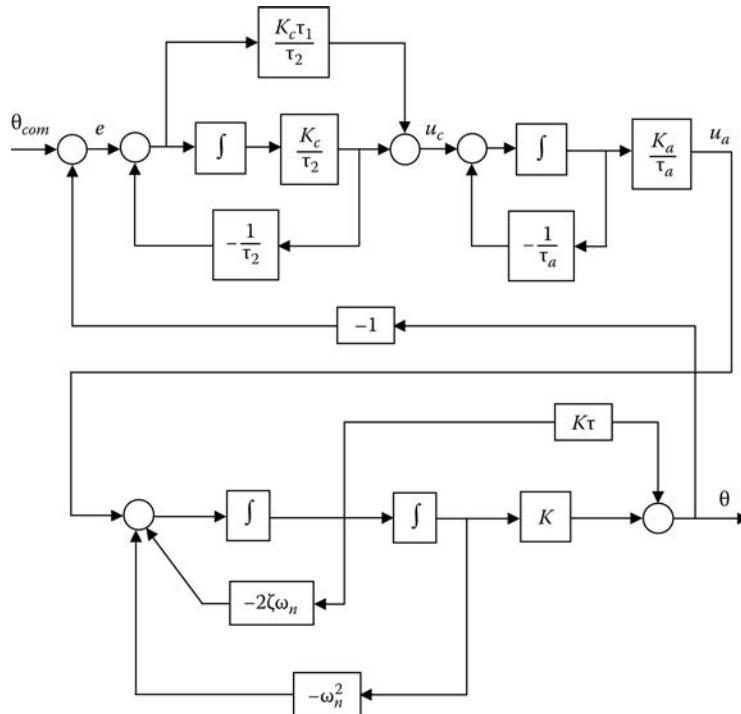


FIGURE 2.22 Simulation diagram for an aircraft pitch control system.

where several of the coefficients a_i , $i = 0, 1, 2, 3$ and b_i , $i = 0, 1, 2, 3, 4$ may be zero.

A simulation diagram of the control system can be obtained from Equation 2.101 using the procedure from the previous section. Alternatively, simulation diagrams can be developed for the individual components in [Figure 2.20](#) and properly connected to produce a simulation diagram for the control system. Simulation of the system based on a simulation diagram using the second approach is preferable since the internal variables are readily identifiable. We can check the simulation results to verify that inputs and outputs of the controller and sensor remain within proper operating ranges.

EXAMPLE 2.8

The control system for the pitch of an aircraft is shown in [Figure 2.21](#).

Draw a simulation diagram for the aircraft pitch control system block diagram.

Simulation diagrams of each component are connected to produce the simulation diagram of the entire control system shown in [Figure 2.22](#).

EXERCISES

2.14 For the control system shown in [Figure 2.20](#).

- Find the coefficients a_i , $i = 0, 1, 2, 3$ and b_i , $i = 0, 1, 2, 3, 4$ in Equation 2.101 in terms of the system parameters τ_1 , τ_2 , K_c , ω_n , K_p , τ_s and K_s .

Hint: The use of Laplace transforms (see Chapter 4) significantly reduces the amount of work necessary to eliminate the variables e , u , and x_s .

b. Draw a simulation diagram based on the fourth-order differential equation model.

- 2.15 Find the differential equation for the control system in Figure 2.21 relating the output θ and its derivatives to the input θ_{com} and its derivatives. Draw the simulation diagram based on the resulting differential equation.

Hint: The use of Laplace transforms (see Chapter 4) significantly reduces the amount of work necessary to eliminate the variables e , u_c , and u_a .

- 2.16 For the railroad cars shown in Figure E2.16,

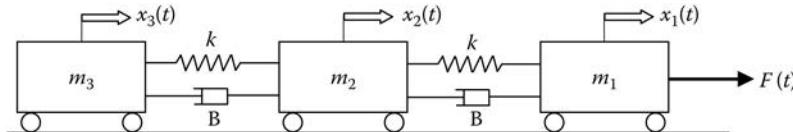


FIGURE E2.16

- Write the differential equation expressing $\sum_k F_{i,k} = m_i \ddot{x}_i, i = 1, 2, 3$ for each car where $F_{i,k}$ is the k th force acting on the i th car.
- Draw a simulation diagram of the system with integrators for x_i , \dot{x}_i , $i = 1, 2, 3$.
- Find the differential equation relating the input $F(t)$ and output $x_1(t)$.

Hint: The use of Laplace transforms (see Chapter 4) significantly reduces the amount of work necessary to eliminate the variables x_2 and x_3 .

2.6 STATE VARIABLES

In everyday terms, one's state of mind on a given day is determined by the history of numerous psychological factors that influence our mental well-being. The state of the national economy (weak, moderate, strong) depends on numerous factors such as energy prices, inflation, trade balances, employment, productivity, housing, tax policies, corporate earnings, transportation, agriculture, and so forth. Imagine that all the economic factors (inputs) affecting the national economy were measurable and the complex interrelationships among those variables that determine the state of the economy were fully understood. If the state of the economy were known at some point in time and the complete set of aforementioned economic factors were observed from that time forward, knowledgeable economists would (in principle) be able to predict the state of the national economy at future times.

The essential point is that if we know the state of a system at some point in time and wish to predict its future, then knowledge of the system inputs only from that time onward is required. The current state of a system reflects the effect of prior inputs that are responsible for the system's transition from some previous state to the current state.

Consider a simple spring-mass-damper system subject to an applied force acting on the mass like the one shown in Figure 2.23. The spring and mass are both capable of storing energy. At any time,

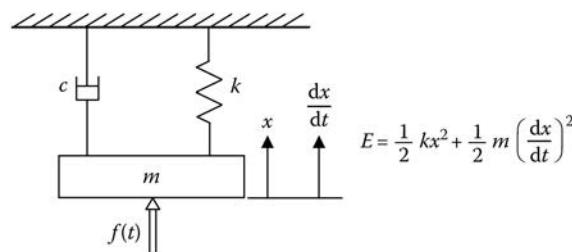


FIGURE 2.23 A spring-mass-damper system with applied force $f(t)$.

the instantaneous energy $E(t)$ stored in the system is given in Equation 2.102 where x is the position of the mass (relative to its equilibrium position) and dx/dt is the velocity of the mass.

$$E = \frac{1}{2} kx^2 + \frac{1}{2} m \left(\frac{dx}{dt} \right)^2 \quad (2.102)$$

A possible choice of state variables for the mechanical system is x and dx/dt . Given both state variables at time t_0 determines the energy $E(t_0)$. The applied force $f(t)$ for $t \geq t_0$ must be known to solve the initial value problem

$$m \frac{d^2}{dt^2} x(t) + c \frac{d}{dt} x(t) + kx(t) = f(t) \quad \text{given } x(t_0) \text{ and } \frac{dx}{dt}(t_0) \quad (2.103)$$

and determine both state variables x and dx/dt as well as $E(t)$ for $t \geq t_0$. The same cannot be said if only the position or the velocity of the mass were known at t_0 . In that case, the initial energy in the system $E(t_0)$ would be unknown, and it would be impossible to predict future values of x and dx/dt even if the force $f(t)$ were known for $t \geq t_0$. Consequently, x or dx/dt alone is not a suitable choice for the state of the system.

The situation is illustrated for the general case of a system with two state variables $x_1(t)$ and $x_2(t)$ and single input $u(t)$ in Figure 2.24. Given $x_1(t_0)$, $x_2(t_0)$, and $u(t)$, $t \geq t_0$, both states can be determined from t_0 on.

The choice of state variables for a dynamic system model is not unique; however, the number of state variables is limited to the minimum number of variables, which satisfy the requirement of predicting future states given the current state and future inputs. This number of state variables is equal to the number of independent energy storage components present in the system. It is advantageous to choose physical (measurable) quantities as in the case of the mechanical system in Figure 2.23 whenever possible.

A simulation diagram is a valuable tool when it comes to choosing the state variables of a system. The outputs of each integrator in a simulation diagram representation of a system is a valid choice for the state variables. The choice of which integrator output is x_1 , x_2 , and so forth is arbitrary.

Consider a second-order system governed by

$$\frac{d^2}{dt^2} y(t) + a_1 \frac{d}{dt} y(t) + a_0 y(t) = b_0 u(t) \quad (2.104)$$

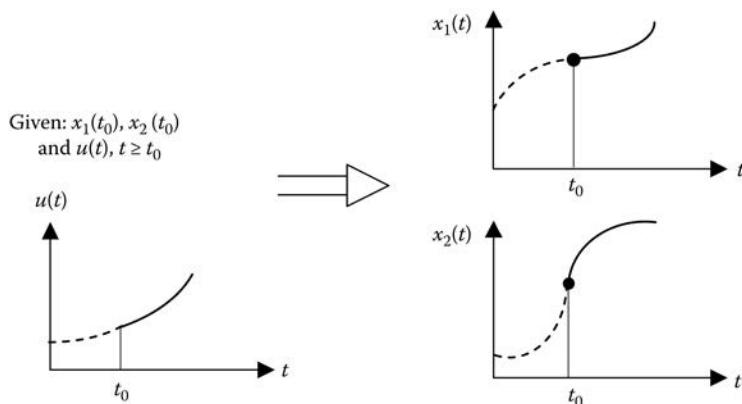


FIGURE 2.24 Dynamic system with state variables $x_1(t)$ and $x_2(t)$.

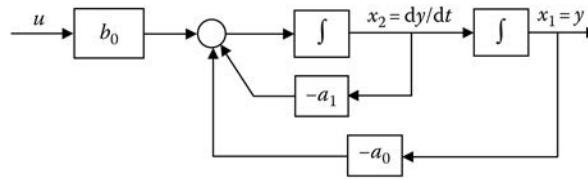


FIGURE 2.25 Simulation diagram of second-order system with state $x_1 = y$ and $x_2 = dy/dt$.

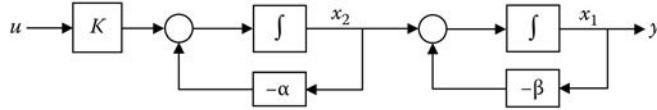


FIGURE 2.26 Simulation diagram for critically damped or overdamped second-order system using two first-order systems in a series.

A simulation diagram like the one shown in Figure 2.25 is easily constructed. State variables x_1 and x_2 are chosen as the output y and first derivative dy/dt , respectively.

The second-order system is critically damped or overdamped if $a_1^2 - 4a_0 \geq 0$. In this case, it is equivalent to two cascaded first-order systems as shown in Figure 2.26.

The parameters K , α , and β are related to a_0 , a_1 , and b_0 according to

$$K = b_0, \alpha = \frac{a_1 \pm \sqrt{a_1^2 - 4a_0}}{2}, \beta = \frac{a_1 \mp \sqrt{a_1^2 - 4a_0}}{2} \quad (2.105)$$

State variable x_1 is again the system output y ; however, the second state variable x_2 is no longer the output derivative dy/dt .

For an n th-order linear system model with constant coefficients, the state derivatives are expressible as a linear combination of the state variables and input(s). For example, from Figure 2.25, the state derivatives are equal to

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= b_0 u - a_0 x_1 - a_1 x_2 \end{aligned} \quad (2.106)$$

whereas in Figure 2.26, the appropriate expressions are

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 - \beta x_1 \\ \frac{dx_2}{dt} &= Ku - \alpha x_2 \end{aligned} \quad (2.107)$$

In the general linear case with n states x_1, x_2, \dots, x_n and r inputs,

$$\frac{dx_1}{dt} = f_1(\underline{x}, \underline{u}) = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + b_{11}u_1 + b_{12}u_2 + \dots + b_{1r}u_r \quad (2.108)$$

$$\frac{dx_2}{dt} = f_2(\underline{x}, \underline{u}) = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + b_{21}u_1 + b_{22}u_2 + \dots + b_{2r}u_r \quad (2.109)$$

$$\frac{dx_n}{dt} = f_n(\underline{x}, \underline{u}) = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n + b_{n1}u_1 + b_{n2}u_2 + \dots + b_{nr}u_r \quad (2.110)$$

where

$$\underline{x} \text{ is the } n \times 1 \text{ state vector} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\underline{u} \text{ is the } r \times 1 \text{ input vector} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}$$

and $f_i(\underline{x}, \underline{u})$, $i = 1, 2, 3, \dots, n$ is the state derivative function of the i th state variable.

Equations 2.108 through 2.110 can be written in the compact form

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) = A\underline{x} + B\underline{u} \quad (2.111)$$

where

$$\dot{\underline{x}} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \vdots \\ \frac{dx_n}{dt} \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & \dots & \dots & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{dx_n}{dt} & a_{n1} & a_{n2} & \dots & \dots & a_{nn} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & \dots & \dots & b_{1r} \\ b_{21} & b_{22} & \dots & \dots & \dots & b_{2r} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & \dots & \dots & b_{nr} \end{bmatrix}$$

The $n \times n$ matrix A is called the system matrix, and the $n \times r$ matrix B is the input matrix.

Multivariable, LTI systems involve multiple inputs u_1, u_2, \dots, u_r and outputs y_1, y_2, \dots, y_p . The outputs are linearly related to the states and the inputs according to

$$\underline{y} = C\underline{x} + D\underline{u} \quad (2.112)$$

where

$$\underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}, \quad C = \begin{bmatrix} c_{11} & c_{12} & \dots & \dots & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & \dots & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{p1} & c_{p2} & \dots & \dots & \dots & c_{pn} \end{bmatrix}, \quad D = \begin{bmatrix} d_{11} & d_{12} & \dots & \dots & \dots & d_{1r} \\ d_{21} & d_{22} & \dots & \dots & \dots & d_{2r} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ d_{p1} & d_{p2} & \dots & \dots & \dots & d_{pr} \end{bmatrix}$$

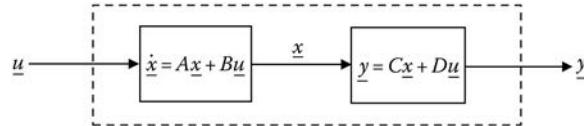


FIGURE 2.27 Dynamic system with input \underline{u} , output \underline{y} , and state \underline{x} .

The $p \times n$ constant matrix C is called the output matrix, and the $p \times r$ matrix D is the direct transmission matrix.

Equations 2.111 and 2.112 taken together are the state equations of the system. Note that the states x_1, x_2, \dots, x_n , are internal to the system as shown in Figure 2.27. Multivariable systems are easier to analyze in terms of state variables compared to the input–output model description of the system, that is, $dy_i/dt = f_i(\underline{y}, \underline{u})$, $i = 1, 2, \dots, n$.

EXAMPLE 2.9

Interacting tanks with inflows into both tanks are shown in Figure 2.28. Choose the states to be the levels $H_1 = H_1(t)$ and $H_2 = H_2(t)$ and the single output as the volume of liquid in both tanks. Write the state equations for the system.

The continuous-time model of the linear tanks consists of the following equations:

$$A_1 \frac{dH_1}{dt} + F_{0,1} = F_1 \quad (2.113)$$

$$F_{0,1} = \frac{1}{R_{12}}(H_1 - H_2) \quad (2.114)$$

$$A_2 \frac{dH_2}{dt} + F_{0,2} = F_{0,1} + F_2 \quad (2.115)$$

$$F_{0,2} = \frac{1}{R_2}H_2 \quad (2.116)$$

Eliminating $F_{0,1}$ and $F_{0,2}$ from Equations 2.113 and 2.115 yields

$$A_1 \frac{dH_1}{dt} + \frac{1}{R_{12}}(H_1 - H_2) = F_1 \quad (2.117)$$

$$A_2 \frac{dH_2}{dt} + \frac{1}{R_2}H_2 = \frac{1}{R_{12}}(H_1 - H_2) + F_2 \quad (2.118)$$

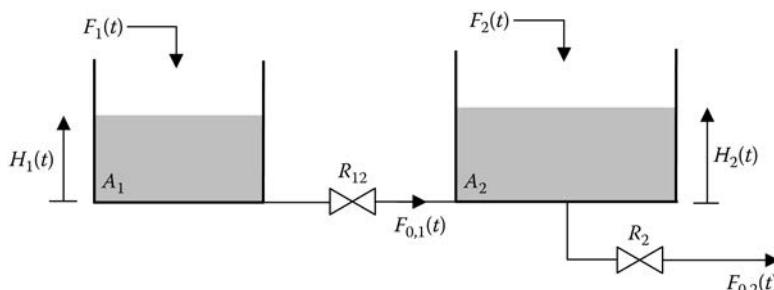


FIGURE 2.28 A system of interacting tanks.

Solving for the state derivatives in Equations 2.117 and 2.118

$$\frac{dH_1}{dt} = -\frac{1}{A_1 R_{12}} H_1 + \frac{1}{A_1 R_{12}} H_2 + \frac{1}{A_1} F_1 \quad (2.119)$$

$$\frac{dH_2}{dt} = \frac{1}{A_2 R_{12}} H_1 - \left[\frac{1}{A_2 R_2} + \frac{1}{A_2 R_{12}} \right] H_2 + \frac{1}{A_2} F_2 \quad (2.120)$$

Writing Equations 2.119 and 2.120 in matrix form gives the first part of the state equations,

$$\begin{bmatrix} \frac{dH_1}{dt} \\ \frac{dH_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_{12}} & \frac{1}{A_1 R_{12}} \\ \frac{1}{A_2 R_{12}} & -\frac{1}{A_2 R_2} - \frac{1}{A_2 R_{12}} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A_1} & 0 \\ 0 & \frac{1}{A_2} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (2.121)$$

The single output V_T , which represents the volume of liquid in both tanks, is

$$V_T = A_1 H_1 + A_2 H_2 = [A_1 \ A_2] \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \quad (2.122)$$

The transmission matrix D is a 1×2 matrix of zeros due to the absence of a direct coupling from either input F_1 or F_2 to the output V_T .

2.6.1 CONVERSION FROM LINEAR STATE VARIABLE FORM TO SINGLE INPUT–SINGLE OUTPUT FORM

In Section 2.3, an example was presented illustrating the conversion of a second-order state variable model into a second-order differential equation by eliminating one of the state variables (see Equations 2.37, 2.38, and 2.42). The procedure involved manipulation and substitution of terms in the time domain, an approach that quickly becomes unwieldy as the number of state variables increases. Simpler methods are described in [Chapter 4](#).

For a linear, third-order system with a single input, the starting point is the state variable model consisting of three coupled first-order differential equations expressing the state derivatives as a linear function of the states and input

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + b_1 u \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + b_2 u \\ \dot{x}_3 &= a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + b_3 u \end{aligned} \quad (2.123)$$

where the output y is x_1 , x_2 , or x_3 .

A third order, input–output differential equation model equivalent to Equation 2.123 is

$$\ddot{y} + \alpha_2 \ddot{y} + \alpha_1 \dot{y} + \alpha_0 y = \beta_2 \ddot{u} + \beta_1 \dot{u} + \beta_0 u \quad (2.124)$$

Expressions for the system coefficients α_2 , α_1 , and α_0 and input coefficients β_2 , β_1 , and β_0 are summarized in Equations 2.125 through 2.127 and [Table 2.1](#).

$$\alpha_2 = -(a_{11} + a_{22} + a_{33}) \quad (2.125)$$

TABLE 2.1**Input Coefficients on Right-Hand Side of Equation 2.125 for $y = x_1, x_2, x_3$**

y	β_2	β_1	β_0
x_1	b_1	$-(a_{22} + a_{33})b_1 + (a_{12}b_2 + a_{13}b_3)$	$(a_{22}a_{33} - a_{23}a_{32})b_1 + (a_{13}a_{32} - a_{12}a_{33})b_2 + (a_{12}a_{23} - a_{13}a_{22})b_3$
x_2	b_2	$a_{21}b_1 - (a_{11} + a_{33})b_2 + a_{23}b_3$	$(a_{23}a_{31} - a_{21}a_{33})b_1 + (a_{11}a_{33} - a_{13}a_{31})b_2 + (a_{13}a_{21} - a_{11}a_{23})b_3$
x_3	b_3	$a_{31}b_1 + a_{32}b_2 - (a_{11} + a_{22})b_3$	$(a_{21}a_{32} - a_{22}a_{31})b_1 + (a_{12}a_{31} - a_{11}a_{32})b_2 + (a_{11}a_{22} - a_{12}a_{21})b_3$

$$\alpha_1 = a_{11}(a_{22} + a_{33}) - a_{12}a_{21} - a_{13}a_{31} + a_{22}a_{33} - a_{23}a_{32} \quad (2.126)$$

$$\alpha_0 = a_{11}(a_{23}a_{32} - a_{22}a_{33}) + a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{22}a_{31} - a_{21}a_{32}) \quad (2.127)$$

2.6.2 GENERAL SOLUTION OF THE STATE EQUATIONS

A solution to the state equation, Equation 2.111 can be found in any one of the texts on linear control theory listed in References. The solution is expressed in terms of an $n \times n$ matrix $\Phi(t)$, called the transition matrix of the system.

$$x(t) = \Phi(t)x(0) + \int_0^t \Phi(t-\tau)Bu(\tau) - d\tau \quad (2.128)$$

The transition matrix depends solely on the system matrix A . One method for finding $\Phi(t)$ uses a definition based on an infinite series,

$$\Phi(t) = I + (tA) + \frac{1}{2!}(tA)^2 + \frac{1}{3!}(tA)^3 + \dots \quad (2.129)$$

As an illustration of how the transition matrix is used to solve the linear state equations, suppose the system matrix for an autonomous system ($u = 0$) is

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

Using the infinite series expansion in Equation 2.129 or some other method (see [Chapter 4](#)) for finding $\Phi(t)$ the result is

$$\Phi(t) = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \frac{1}{2} \quad (2.130)$$

and from Equation 2.128, the state $x(t)$, $t \geq 0$ is

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \quad (2.131)$$

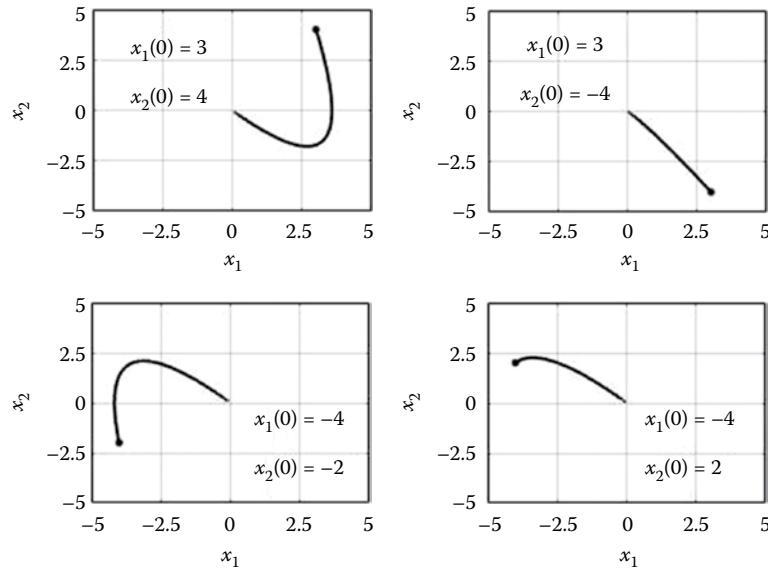


FIGURE 2.29 State trajectory in Equation 2.131 for different initial states.

The state trajectory or state portrait is a plot showing the path of the state vector in state space. In the general case, there is a separate coordinate axis for each of the state variables. The time variable “*t*” does not appear explicitly; however, each point along the state trajectory corresponds to a specific point in time. Figure 2.29 shows four different state trajectories starting from different initial states. Note that the four state trajectories all terminate at the origin, the equilibrium point of the system.

EXERCISES

- 2.17 For the system of interacting tanks in Example 2.9.

- Draw the simulation diagram of the system.
- Choose a new set of state variables as

$$z_1 = H_1 + H_2, z_2 = H_1 - H_2$$

and find the new system and input matrices A and B where

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = A \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + B \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

Hint: Find H_1 and H_2 in terms of z_1 and z_2 .

- Find the new output matrix C where

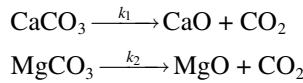
$$V_T = C \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

- 2.18 Write the state equations for the system of three railroad cars in Exercise 2.16. Choose the outputs to be the positions of each car.
- 2.19 An ecosystem consists of three species whose populations are denoted by F , S , and G . The growth rates of each specie are given by

$$\begin{aligned}\text{Growth rate of } F &= \frac{1}{F} \frac{dF}{dt} = a - cS - u_F \\ \text{Growth rate of } S &= \frac{1}{S} \frac{dS}{dt} = -k + \lambda F - m_G - u_S \\ \text{Growth rate of } G &= \frac{1}{G} \frac{dG}{dt} = -e + \sigma S + u_G\end{aligned}$$

Write the system in state variable form $\dot{x} = f(x, u)$ $y = g(x, u)$ with the state $x = [F \ S \ G]^T$, input $u = [u_F \ u_S \ u_G]^T$, and output chosen as $y = F + S + G$.

- 2.20 Limestone is reduced to calcium oxide (CaO), magnesium oxide (MgO), and carbon dioxide (CO_2) by heating it in a reaction vessel maintained at a constant high temperature (McClamroch 1980). The limestone is made up of a fixed fraction β of calcium carbonate(CaCO_3), and the rest is magnesium carbonate (MgCO_3). The process is described by the first-order irreversible chemical reactions



where k_1 and k_2 are the rate constants for the two reactions.

Limestone is added to the reaction vessel at a rate of u mol/h. The mass (in moles) of CaCO_3 , MgCO_3 , CaO , and MgO in the vessel are denoted by x_1 , x_2 , x_3 , and x_4 , respectively (see Figure E2.20).

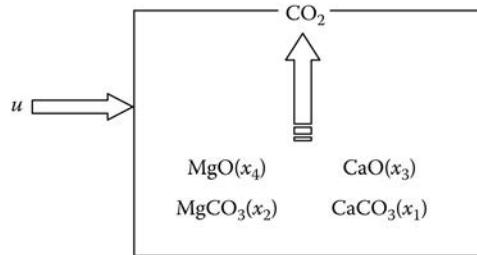


FIGURE E2.20

Since each mole of reactant that decomposes yields one mole of product (plus one mole of carbon dioxide), the state equations are

$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 + \beta u \\ \dot{x}_2 &= -k_2 x_2 + (1 - \beta)u \\ \dot{x}_3 &= k_1 x_1 \\ \dot{x}_4 &= k_2 x_2\end{aligned}$$

- Draw a simulation diagram of the system. What is the order of the system?
- Find the matrices A , B , C , and D in the state equation model if the outputs are $y_1 = x_3$ and $y_2 = x_4$.
- Find the differential equation relating y_1 and u . Comment on the result.

- d. Repeat part (c) for y_2 and u .
- e. The vessel is initially empty and $u(t) = A$, $t \geq 0$. Find analytic expressions for the state variables.

2.21 The populations of three species in a restricted area are governed by the differential equations

$$\begin{aligned}\dot{P}_1(t) &= a_{11}P_1(t) + a_{12}P_2(t) + a_{13}P_3(t) + c_1u(t) \\ \dot{P}_2(t) &= a_{21}P_1(t) + a_{22}P_2(t) + a_{23}P_3(t) + c_2u(t) \\ \dot{P}_3(t) &= a_{31}P_1(t) + a_{32}P_2(t) + a_{33}P_3(t) + c_3u(t) \\ 0 \leq c_1 &\leq 1, \quad 0 \leq c_2 \leq 1, \quad 0 \leq c_3 \leq 1, \quad \text{and} \quad c_1 + c_2 + c_3 = 1\end{aligned}$$

where $u(t)$ is the total immigration rate for all species. The constants c_1 , c_2 , and c_3 represent the fraction of $u(t)$ immigrating to each of the species populations.

- a. Draw a simulation diagram of the system.
- b. Find the third-order differential equation relating $P_1(t)$ and $u(t)$.
- c. Draw a simulation diagram of the system containing three integrators in series where the input to the first integrator is $\ddot{p}_1(t)$.

2.7 NONLINEAR SYSTEMS

Real-world dynamic systems exhibit nonlinear behavior. The continuous-time models that relate inputs and outputs of actual systems are (entirely or partially) composed of nonlinear algebraic and differential equations. We may well choose to employ a linear model as an approximation of a nonlinear system because it is far simpler to work with. A unified approach to solving nonlinear algebraic equations does not exist, to say nothing of nonlinear differential equations.

The principle of superposition states that if a system responds to inputs $u_1(t)$ and $u_2(t)$ with outputs $y_1(t)$ and $y_2(t)$, then the system's response to a linear combination of the inputs $u(t) = c_1u_1(t) + c_2u_2(t)$ is $y(t) = c_1y_1(t) + c_2y_2(t)$. Superposition is a property of linear system models. It is not applicable to models of nonlinear systems.

Unlike linear system models, a nonlinear system model exhibits dynamic response properties whose nature is dependent on the magnitude of its inputs and the initial state. Consider the two simple first-order systems, one linear and the other nonlinear, in Figure 2.30. Both systems are driven by the identical input.

Discrete-time system approximations for both continuous-time systems can be obtained by replacing the first derivative terms with divided differences, that is,

$$\frac{dy}{dt} \approx \frac{y_A[(n+1)T] - y_A[(nT)]}{(n+1)T - nT} = \frac{y_A(n+1) - y_A(n)}{T} \quad (2.132)$$

$$\frac{dz}{dt} \approx \frac{z_A[(n+1)T] - z_A[(nT)]}{(n+1)T - nT} = \frac{z_A(n+1) - z_A(n)}{T} \quad (2.133)$$

resulting in difference equations

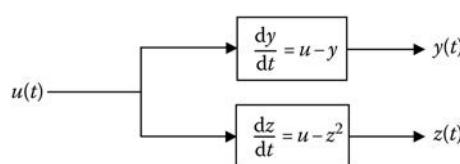


FIGURE 2.30 Linear and nonlinear system subject to identical input.

$$y_A(n+1) = y_A(n) + T[u(n) - y_A(n)] \quad (2.134)$$

$$z_A(n+1) = z_A(n) + T[u(n) - z_A^2(n)] \quad (2.135)$$

Equations 2.134 and 2.135 can be solved recursively for $y_A(n)$ and $z_A(n)$, $n = 1, 2, 3, \dots$ given initial values for $y_A(0)$ and $z_A(0)$. The results (every third point) are plotted in [Figure 2.31](#) when the initial condition is zero for inputs $u(t) = 1$ and $u(t) = 10$.

Approximate responses $y_A(nT)$ for both inputs are typical linear first-order system step responses, namely, they each require roughly four to five time constants ($\tau = 1$ s) to reach steady state. Furthermore, the response $y_A(nT)$ in the lower left corner where $u(t) = 10$ is 10 times the response $y_A(nT)$ in the upper left corner where $u(t) = 1$. For a constant input $u(t) = \bar{u}$, the steady-state value is $y_A(\infty) = \bar{u}$ for the linear system.

In contrast to the linear system, the transient period of the nonlinear system is shorter when the input $u(t) = 10$ compared to when $u(t) = 1$. Furthermore, $z_A(\infty) = \bar{u}^{1/2}$ for the nonlinear system when the input is $u(t) = \bar{u}$, in violation of the principle of superposition.

A linear model approximation of a nonlinear system is often acceptable provided the system variables (inputs, states, outputs) are confined to a restricted operating region. A simple example serves to illustrate the point. Consider a system with input $u = u(t)$ and state $x = x(t)$ described by

$$\frac{dx}{dt} + 0.2x^{1/2} = u \quad (2.136)$$

The state derivative function is a nonlinear function of x , that is,

$$\frac{dx}{dt} = f(x, u) = -0.2x^{1/2} + u \quad (2.137)$$

For arbitrary input $u(t)$, the solution to Equation 2.137 can be approximated in a way similar to what we did in [Chapter 1](#) using difference equations. However, suppose the input $u(t)$ is confined to a

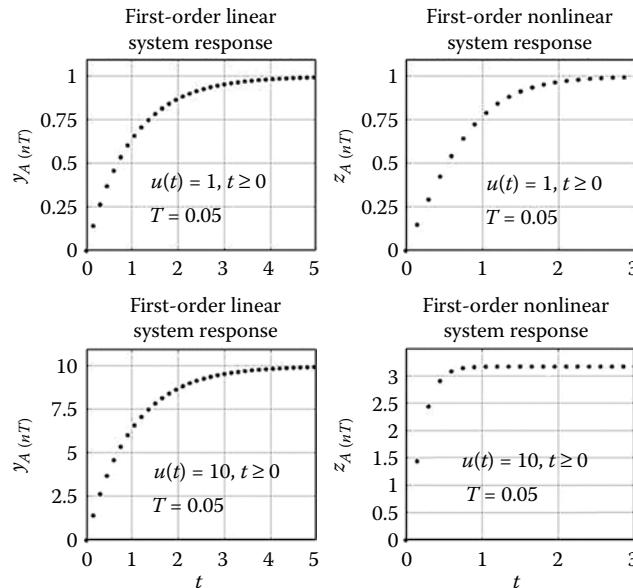


FIGURE 2.31 Approximation of linear and nonlinear system step responses.

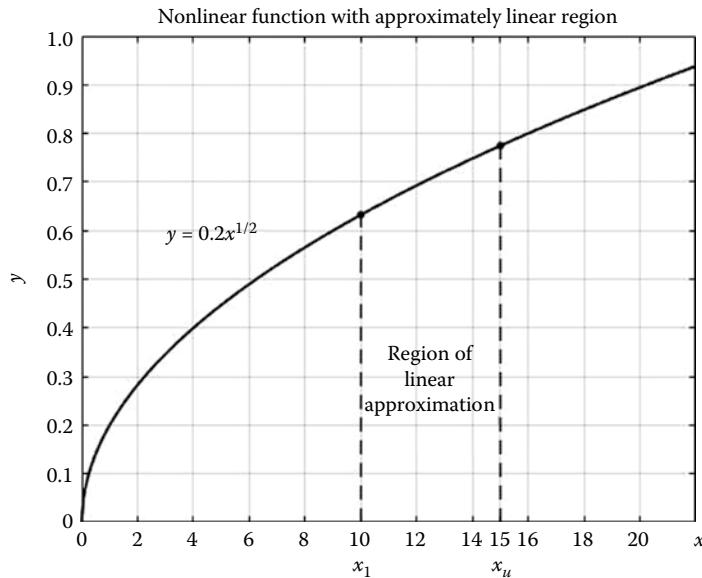


FIGURE 2.32 Linearizing the nonlinear function $0.2x^{1/2}$ in an interval $x_1 \leq x \leq x_u$.

range that results in the state $x(t)$ varying between x_1 and x_u as shown in Figure 2.32. It is reasonable to assume the term $0.2x^{1/2}$ in Equation 2.136 could be replaced by a linear function of x resulting in a simpler model. We will have more to say about linearization of nonlinear system models in Chapter 7.

Another distinguishing property of linear systems is the way they respond to sinusoidal inputs. At steady state, the output of a linear system forced by a sinusoidal input with radian frequency ω_0 is itself a sinusoid at the same frequency. In general, the output is shifted in time (out of phase) with respect to the input, and the amplitude is either attenuated or amplified compared to the amplitude of the input. This property is the foundation of linear AC steady-state analysis and the design of linear control systems by the method of frequency response. In the case of nonlinear systems, the output includes harmonics (sinusoidal terms at frequencies $n\omega_0$, $n = 1, 2, 3, \dots$).

The type of nonlinearity portrayed in Figure 2.32 has been classified as “progressive” (Buckley 1964). The distinguishing characteristics of progressive nonlinearities are their monotonic continuous nature over the range of input and output values of interest. Furthermore, state derivative functions which are progressive nonlinearities can be approximated by linearization methods. “Essential” nonlinearities are those that cannot be represented by a simple continuous analytical function. Phenomena such as friction, dead zone and saturation in valves, and backlash in gears in mechanical systems; hysteresis in electrical components; and analog-to-digital quantization are examples of essential nonlinearities.

2.7.1 FRICTION

The first example illustrates a type of friction called coulomb friction. An object of mass m , resting on a flat surface, is subject to an external horizontal force $f(t)$ and a resisting frictional force f_μ as shown in Figure 2.33. The velocity of the mass obeys the relation in Equation 2.138

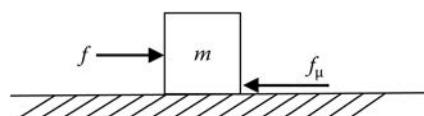


FIGURE 2.33 Nonlinear system example—coulomb friction.

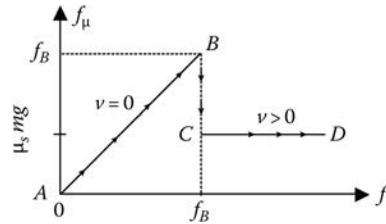


FIGURE 2.34 Friction force f_μ versus increasing f applied to a mass initially at rest.

$$m \frac{dv}{dt} + f_\mu = f \quad (2.138)$$

The friction force f_μ is equal in magnitude to the force f until a breakaway force f_B is applied (see Figure 2.34), and the mass begins to slide along the surface. The breakaway force f_B depends on the coefficient of static friction μ_0 and the object's weight,

$$f_B = \mu_0 mg \quad (2.139)$$

While in motion, the friction force f_μ is a constant dependent on the coefficient of sliding friction μ_s and the weight mg of the object as seen in Equation 2.140. Note that f_μ is also equal to $\mu_s mg$ when $f \leq f_B$ and $v > 0$.

$$f_\mu = \begin{cases} f & \text{when } f \leq f_B(v=0) \\ \mu_s mg & \text{when } f > f_B(v>0) \end{cases} \quad (2.140)$$

EXAMPLE 2.10

The applied force $f(t)$ is shown in Figure 2.35. Find the velocity of the object.

$$f(t) = \begin{cases} 2f_B \left(\frac{t}{t_1} \right) & 0 \leq t < t_1 \\ 2f_B \left[2 - \left(\frac{t}{t_1} \right) \right] & t_1 \leq t < 2t_1 \\ 0 & t \geq 2t_1 \end{cases} \quad (2.141)$$

The difference equation resulting from the substitution of the divided difference $[v_A(n+1) - v_A(n)]/T$ for the first derivative dv/dt in Equation 2.138 is

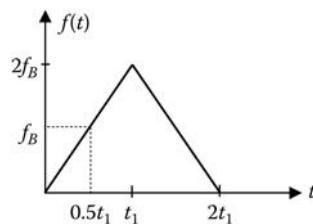


FIGURE 2.35 Applied force $f(t)$ versus t .

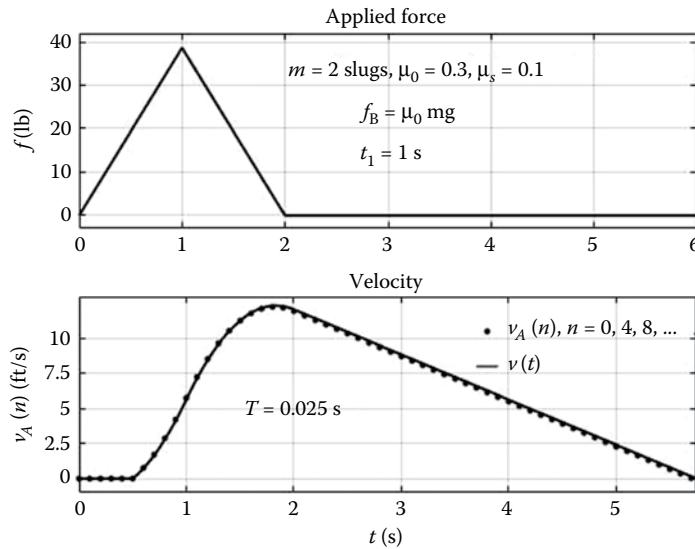


FIGURE 2.36 Approximate solution $v_A(n)$, $n = 0, 4, 8, \dots$ and exact solution $v(t)$, $t \geq 0$.

$$v_A(n+1) = v_A(n) + \frac{T}{m} [f(n) - f_{\mu}(n)] \quad (2.142)$$

A recursive solution for $v_A(n)$, $n = 1, 2, 3, \dots$ given $v_A(0) = v(0) = 0$ is not as straightforward as it was in previous examples owing to the nature of the friction force. The MATLAB® M-file "Ch2_Ex2_10.m" includes the necessary conditional statements to handle the discontinuity in f_{μ} . Results are shown in Figure 2.36.

The analytical solution for the velocity $v(t)$ is plotted along with the approximate solution $v_A(n)$. It can be found by integrating the differential equation (Equation 2.138) over consecutive intervals using the appropriate value for the friction force (f or $\mu_s mg$) and the correct initial velocity for each interval. The details are left for an exercise; the results are as follows.

$$v(t) = \begin{cases} 0, & 0 \leq t \leq 0.5t_1 \\ gt_1 \left[\mu_0 \left(\frac{t}{t_1} \right)^2 - \mu_s \left(\frac{t}{t_1} \right) + \frac{2\mu_s - \mu_0}{4} \right], & 0.5t_1 \leq t < t_1 \\ gt_1 \left[-\mu_0 \left(\frac{t}{t_1} \right)^2 + (4\mu_0 - \mu_s) \left(\frac{t}{t_1} \right) - \frac{9\mu_0 - 2\mu_s}{4} \right], & t_1 \leq t < 2t_1 \\ gt_1 \left[-\mu_s \left(\frac{t}{t_1} \right) + \frac{7\mu_0 + 2\mu_s}{4} \right], & 2t_1 \leq t < T_f \\ 0, & T_f < t \end{cases} \quad (2.143)$$

The time T_f when the velocity returns to zero is obtained from

$$T_f = \frac{t_1}{4\mu_s} (7\mu_0 + 2\mu_s) \quad (2.144)$$

2.7.2 DEAD ZONE AND SATURATION

The next example of mechanical (pneumatic) nonlinearity is a valve that contains two nonlinear elements, dead zone and saturation. First, consider the nonlinear elements individually. An ideal dead zone nonlinearity is shown in [Figure 2.37](#). The dead zone is the region between t_1 and t_2 .

$$f(t) = \begin{cases} f_3 \left(\frac{t - t_2}{t_3 - t_2} \right) & t_2 \leq t \\ 0 & t_1 < t < t_2 \\ f_0 \left(\frac{t - t_1}{t_0 - t_1} \right) & t \leq t_1 \end{cases} \quad (2.145)$$

An ideal saturation nonlinearity is shown in [Figure 2.38](#).

$$f(t) = \begin{cases} f_s \left(\frac{t}{t_s} \right) & |t| \leq t_s \\ \text{sgn}(f_s) & |t| > t_s \end{cases} \quad (2.146)$$

The saturated regions are when $|t| > t_s$, that is, for $t < -t_s$, the value of $-f(t)$ does not change and for $t > t_s$, the value of $f(t)$ does not change.

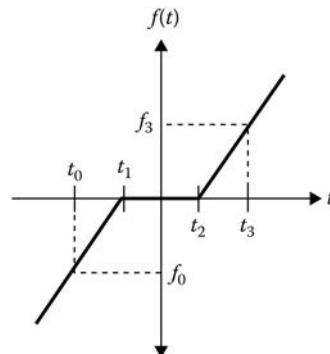


FIGURE 2.37 Dead zone nonlinearity.

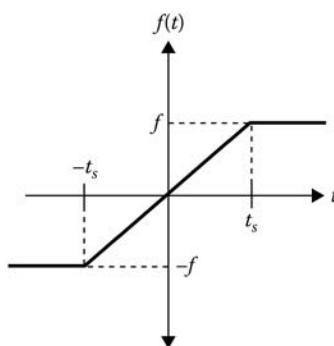


FIGURE 2.38 Saturation nonlinearity.

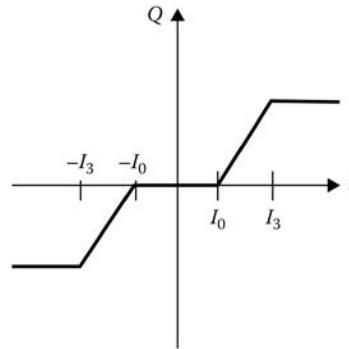


FIGURE 2.39 Valve flow versus current.

Together, these nonlinearities (saturation and dead zone) form an approximation to the pneumatic behavior of a valve shown in [Figure 2.39](#).

$\pm I_o$ is the opening current, that is, the current needed to open the valve. $\pm I_s$ is the saturation current where any additional current (more than I_s or less than $-I_s$) does not open the valve any further. The region between I_o and I_s ($-I_o$ and $-I_s$) is appropriately called the active region. The region between $-I_o$ and I_o is called the dead zone. However, in practice, leakage occurs below the opening current.

2.7.3 BACKLASH

Backlash nonlinearity often occurs in gears due to the spacing between individual teeth. The spacing is needed for the gears to mesh without binding. This spacing (d) is shown in [Figure 2.40](#).

[Figure 2.41](#) shows a plot of the backlash nonlinearity.

Assume the space d exists in the initial condition as in [Figure 2.40](#). As the leading gear moves in one direction, the following gear does not move until contact is made after the leading gear is displaced by d . Then, the following gear tracks the leading gear as indicated by section 1 of [Figure 2.41](#). When the leading gear reverses direction, it must be displaced by a distance $2d$ before contact is reestablished with the following gear, as indicated by section 2 of [Figure 2.41](#). Similar to before, the following gear tracks the leading gear as indicated by section 3 of [Figure 2.41](#). Another reversal of directions leads to section 4 in [Figure 2.41](#).

2.7.4 HYSTERESIS

The graph of f_μ versus f in [Figure 2.34](#) is applicable so long as the applied force f and resulting velocity v are increasing along the path A-B-C-D. Once the block is in motion and the applied force

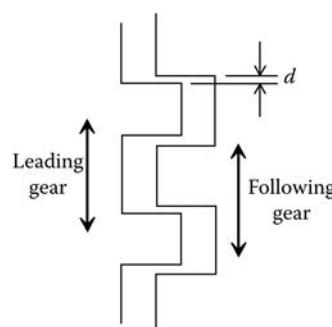


FIGURE 2.40 Backlash in gear teeth.

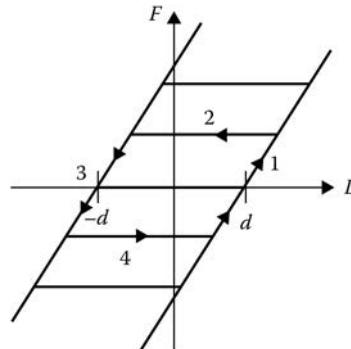


FIGURE 2.41 Backlash nonlinearity.

f diminishes to zero, the return path does not follow D-C-B-A. That is, the sliding block does not abruptly stop when the applied force is reduced to f_B . Rather, the friction force remains at $\mu_s mg$ until the block decelerates to zero velocity. This type of nonlinear phenomenon is referred to as hysteresis.

An example of a real system with hysteresis, present by design, is a thermostatically controlled furnace supplying heat to a building. A simplified diagram of the system is depicted in Figure 2.42. An energy balance on the building interior space relates the accumulation of thermal energy to the heat flow from the furnace and heat loss to the outside.

The equation is

$$C \frac{dT}{dt} = Q - Q_0 \quad (2.147)$$

where C is the thermal capacitance of the air and contents inside the building, all of which are assumed to be at temperature T . The heat loss Q_0 is assumed proportional to the temperature difference $T - T_0$, that is,

$$Q_0 = \frac{T - T_0}{R} \quad (2.148)$$

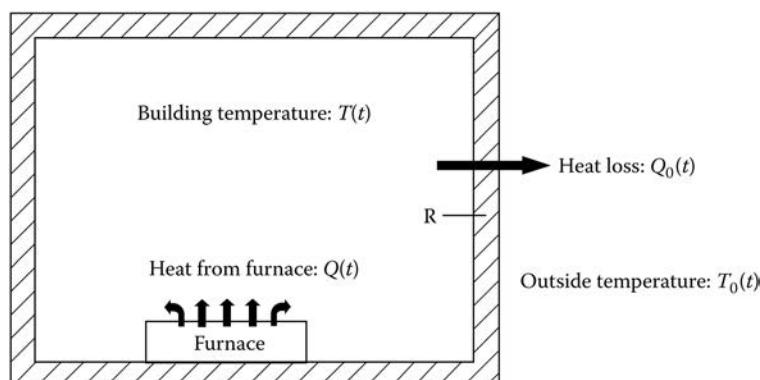


FIGURE 2.42 Temperature regulation in a building.

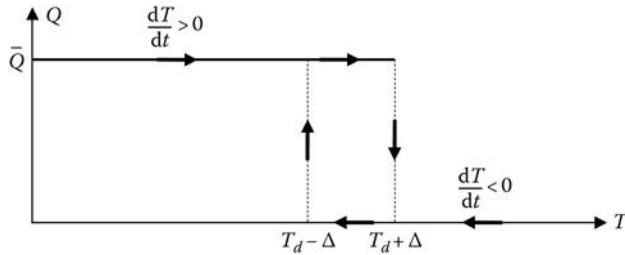


FIGURE 2.43 Hysteresis in furnace output versus building temperature.

with R the overall thermal resistance of the exterior walls and insulation. Combining Equations 2.147 and 2.148 and introducing the thermal system time constant $\tau = RC$ result in the first-order model

$$\tau \frac{dT}{dt} + T = RQ + T_0 \quad (2.149)$$

The furnace operates in one of two modes, on or off, depending on whether the building temperature T is below or above some tolerance Δ about a desired temperature T_d and whether the building temperature is increasing or decreasing. When it is on, a constant amount of heat \bar{Q} is supplied; conversely, no heat is produced when the furnace is off. In mathematical terms,

$$Q = \begin{cases} \bar{Q}, & T \leq T_d - \Delta \text{ or } T_d - \Delta < T < T_d + \Delta \text{ and } dT/dt > 0 \\ 0, & T > T_d + \Delta \text{ or } T_d - \Delta < T < T_d + \Delta \text{ and } dT/dt > 0 \end{cases} \quad (2.150)$$

The hysteresis effect is evident from the graph in Figure 2.43 (McClamroch 1980).

From Equations 2.149 and 2.150, it follows that the state derivative dT/dt depends not only on the input T_0 and the state T but also on its own sign. Furthermore, since the furnace output Q in Figure 2.43 is multi-valued whenever the building temperature T falls within $T_d - \Delta$ to $T_d + \Delta$, the initial state $T(0)$ and the initial state of the furnace (on/off) must be specified to simulate or obtain analytical solutions for $T(t)$, $t \geq 0$.

The example that follows illustrates a method for obtaining an approximate solution and the exact solution for the building temperature $T(t)$ when the outside temperature $T_0(t)$ is constant.

EXAMPLE 2.11

A building's thermostat has been off for a period of time sufficient to allow the inside and outside temperatures to equalize. The thermostat is then set to 75°F. It is programmed to turn off when the interior temperature reaches 78°F and back on when it falls below 72°F. The furnace produces 36,000 Btu/h. Thermal capacitance of the occupied space and interior furnishings is 300 Btu/°F, and the thermal resistance of the walls is 8×10^{-4} °F per Btu/h. The outside temperature is a constant 50°F.

- Find the time constant of the system.
- Show that the furnace is capable of raising the building temperature to 78°F.
- Find the temperature response and the time required for the building temperature to reach 78°F.
- Find the temperature response and the time required for the building temperature to cool down to 72°F.
- Find the temperature response and the time required for the building temperature to go back to 78°F.

- f. Simulate the temperature responses in parts (c), (d), and (e) by solving a difference equation with appropriate step size and compare the approximate and exact solutions.
- a. The time constant, a measure of the speed of the system's dynamics is

$$\tau = RC = 8 \times 10^{-4} \frac{\text{°F}}{\text{Btu/h}} \cdot 300 \frac{\text{Btu}}{\text{°F}} = 0.24 \text{ h}$$

- b. The steady-state temperature differential (inside minus outside) that the furnace is capable of maintaining is obtained from the first-order differential equation model in Equation 2.149 with the derivative set to zero and the furnace on, that is, $Q(t) = \bar{Q}$.

$$T_{ss} = R\bar{Q} + \bar{T}_0 \quad (2.151)$$

$$\Rightarrow T_{ss} - \bar{T}_0 = R\bar{Q} \quad (2.152)$$

where

\bar{T}_0 is the constant outside temperature

T_{ss} is the steady-state inside temperature

In this example,

$$T_{ss} - \bar{T}_0 = R\bar{Q} = 8 \times 10^{-4} \frac{\text{°F}}{\text{Btu/h}} \cdot 36,000 \frac{\text{Btu}}{\text{h}} = 28.8 \text{ °F}$$

Hence, the furnace is capable of raising the inside temperature from 50°F to 78.8°F, which is slightly higher than the 78°F shutoff setting of the thermostat.

- c. From Equation 2.6, the step response of the first-order system is

$$T(t) = T(0)e^{-t/\tau} + (\bar{T}_0 + R\bar{Q})(1 - e^{-t/\tau}) \quad (2.153)$$

which describes the building temperature from time $t = 0$ up to $t = t_1$ where

$$T(t_1) = T_d + \Delta = 75 \text{ °F} + 3 \text{ °F} = 78 \text{ °F} \quad (2.154)$$

Solving for t_1 gives

$$\begin{aligned} t_1 &= \tau \ln \left[\frac{(\bar{T}_0 + R\bar{Q}) - T(0)}{(\bar{T}_0 + R\bar{Q}) - (T_d + \Delta)} \right] \\ &= 0.24 \ln \left[\frac{(50 + 28.8) - 50}{(50 + 28.8) - (75 + 3)} \right] = 0.86 \text{ h} \end{aligned} \quad (2.155)$$

From Equation 2.153 with $T(0) = 50 \text{ °F}$, the temperature response is

$$T(t) = 50e^{-t/0.24} + 78.8(1 - e^{-t/0.24}), \quad 0 \leq t \leq 0.86 \quad (2.156)$$

- d. The furnace shuts off when the temperature reaches $T_d + \Delta = 78 \text{ °F}$ and the subsequent cooling from 78°F to $T_d - \Delta = 72 \text{ °F}$ follows the step response in Equation 2.153 with $\bar{Q} = 0$ and $T(0) = T_d + \Delta = 78 \text{ °F}$. Thus,

$$T(t) = (T_d + \Delta)e^{-(t-t_1)/\tau} + \bar{T}_0[1 - e^{-(t-t_1)/\tau}], \quad t_1 \leq t \leq t_2 \quad (2.157)$$

$$= 78e^{-(t-0.86)/0.24} + 50[1 - e^{-(t-0.86)/0.24}], \quad 0.86 \leq t \leq t_2 \quad (2.158)$$

where t_2 is the time when the building temperature is $T_d - \Delta = 72^\circ\text{F}$. Note the $(t - t_1)$ in the exponent of Equation 2.157 since t_1 is the initial time of the step response. From Equation 2.157 with $t = t_2$, $T(t_2) = T_d - \Delta$, the time t_2 is given by

$$\begin{aligned} t_2 &= t_1 + \tau \ln \left[\frac{(T_d + \Delta) - \bar{T}_0}{(T_d - \Delta) - \bar{T}_0} \right] \\ &= 0.86 + 0.24 \ln \left[\frac{(75 + 3) - 50}{(75 - 3) - 50} \right] = 0.92 \text{ h} \end{aligned} \quad (2.159)$$

- e. The cycle is completed when the building temperature returns to $T_d + \Delta = 78^\circ\text{F}$. Using the same approach as before, the result is

$$\begin{aligned} T(t) &= (T_d - \Delta)e^{-(t-t_2)/\tau} + (\bar{T}_0 + R\bar{Q})[1 - e^{-(t-t_2)/\tau}], \quad t_2 \leq t \leq t_3 \\ &= 72e^{-(t-0.92)/0.24} + 78.8[1 - e^{-(t-0.92)/0.24}], \quad 0.92 \leq t \leq t_3 \end{aligned} \quad (2.160)$$

Setting $T(t_3) = T_d + \Delta$ and solving for t_3 ,

$$\begin{aligned} t_3 &= t_2 + \tau \ln \left[\frac{(\bar{T}_0 + R\bar{Q}) - (T_d - \Delta)}{(\bar{T}_0 + R\bar{Q}) - (T_d + \Delta)} \right] \\ &= 0.92 + 0.24 \ln \left[\frac{(50 + 28.8) - (75 - 3)}{(50 + 28.8) - (75 + 3)} \right] = 1.43 \text{ h} \end{aligned} \quad (2.161)$$

- f. The approximate solution for the building temperature is based on the difference equation obtained by replacing the first derivative dT/dt in Equation 2.149 with the finite difference $[T_A(n+1) - T_A(n)]/T$. The result is

$$T_A(n+1) = \left(1 - \frac{\Delta T}{\tau}\right)T_A(n) + \frac{\Delta T}{\tau}[RQ(n) + \bar{T}_0] \quad (2.162)$$

where $Q(n)$ is based on the logic in Equation 2.150. The MATLAB M-file “Ch2_Ex2_11.m” evaluates the exact and approximate solutions and generates the graph shown in [Figure 2.44](#).

The building temperature experiences periodic fluctuations between $T_d - \Delta = 72^\circ\text{F}$ and $T_d + \Delta = 78^\circ\text{F}$ as long as the outside temperature remains constant. The period is equal to $t_3 - t_1 = 1.43 - 0.86 = 0.57 \text{ h}$.

2.7.5 QUANTIZATION

In digital control, it is often desired to discretize the continuous signal of a sensor for use by a computer or microprocessor. Conversion of this signal is achieved by an analog-to-digital converter (ADC) where the signal is quantized.

The quantization nonlinearity is shown in [Figure 2.45](#). In this example, a voltage range between V_0 and V_1 is designated as state zero, S_0 ; a voltage range between V_1 and V_2 is designated as state one, S_1 ; and so on. Each state is represented by a binary expression according to the number of bits

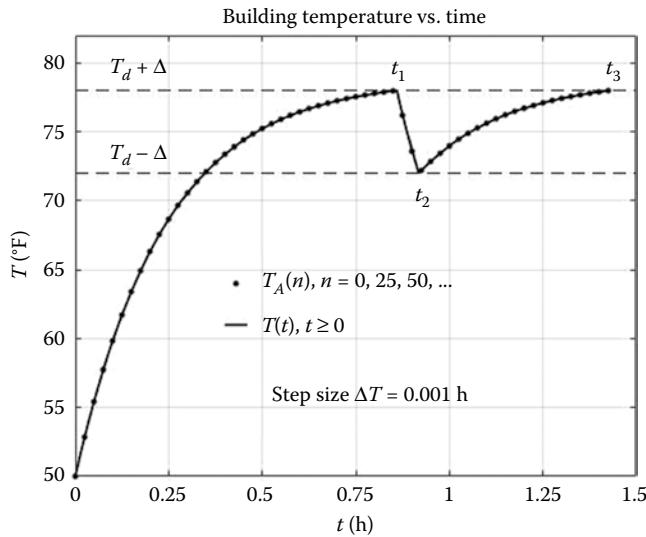


FIGURE 2.44 Exact and approximate solutions for building temperature.

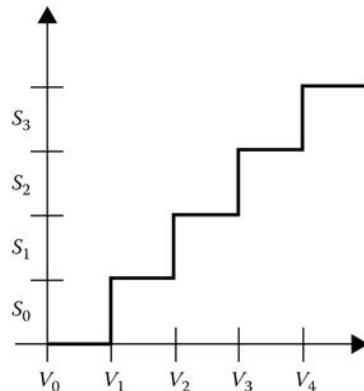


FIGURE 2.45 Quantization.

used by the data type assigned to the state. For example, an 8-bit representation for state zero is 00000000, while state one is represented by 00000001. The more bits that are available for quantization yield a better resolution of the sensor's range, in this case, voltage. There are 2^n states where n is the number of bits. Therefore, an 8-bit ADC has 256 states, 0–255. The resolution is the sensor's range divided by the number of states. For example, a sensor with a voltage range from 0 to 10 V has a resolution of 0.04 V for an 8-bit ADC.

2.7.6 SUSTAINED OSCILLATIONS AND LIMIT CYCLES

Both linear and nonlinear system differential equation models are capable of producing solutions involving sustained oscillations of the state variables. This comes as no surprise for linear systems. Indeed, we have already seen how the natural response of an undamped second-order system continues to oscillate forever (see [Figure 2.4](#)). Examples will be presented in [Chapter 4](#) of forced linear systems with sustained sinusoidal oscillations in the output after the transient response has died out.

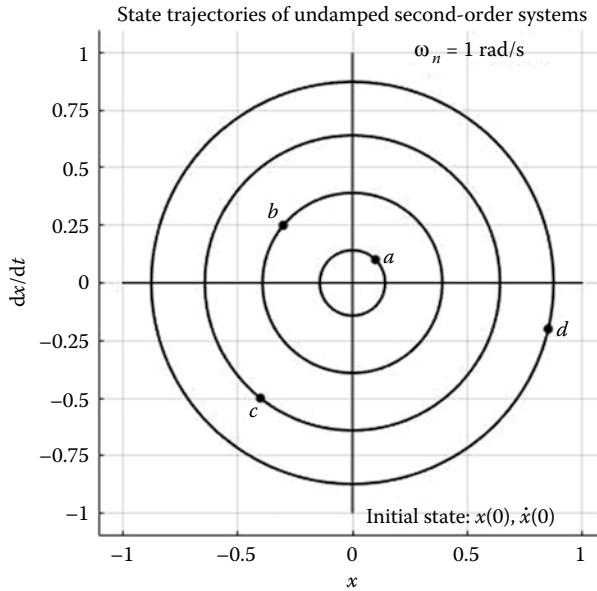


FIGURE 2.46 Closed orbits for the system $\ddot{x} + \omega_n^2 x = 0$ ($\omega_n = 1$ rad/s).

State trajectories of the autonomous system governed by the differential equation

$$\ddot{x} + \omega_n^2 x = 0 \text{ subject to } x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0 \quad (2.163)$$

are closed orbits in the \dot{x} versus x state space. Figure 2.46 shows state trajectories, also known as orbits, for the undamped system in Equation 2.163 with $\omega_n = 1$ rad/s starting from four different initial points in the state space.

The orbits are typically elliptical; however, those in Figure 2.46 are circular because the natural frequency $\omega_n = 1$ rad/s. Sustained oscillations of the state components x and \dot{x} are shown in Figure 2.47.

Nonlinear systems can experience two types of sustained oscillations. The first class is similar to the case of linear systems. In the unforced case, the oscillations are sensitive to the initial conditions. That is, the particular points along the closed path of the state trajectory vary depending on the location of the initial point in state space. The initial point is always on the closed orbit. The amplitude and period of the oscillations depend on the system parameters and initial conditions.

The state trajectories of the nonlinear system described by the coupled first-order differential equations

$$\dot{x}_1 = x_1(a - bx_2) \quad (2.164)$$

$$\dot{x}_2 = x_2(cx_1 - d) \quad (2.165)$$

are concentric closed curves “spun out” in a clockwise rotation from the initial point. The center of rotation is the equilibrium point located at $(d/c, a/b)$. The MATLAB M-file “Ch2_Fig2_46_and_Fig2_47.m” uses a difference quotient with step size $T = 5 \times 10^{-5}$ to approximate the first derivatives in Equations 2.164 and 2.165. The approximate solutions in Figure 2.48 show four orbits starting from different initial states.

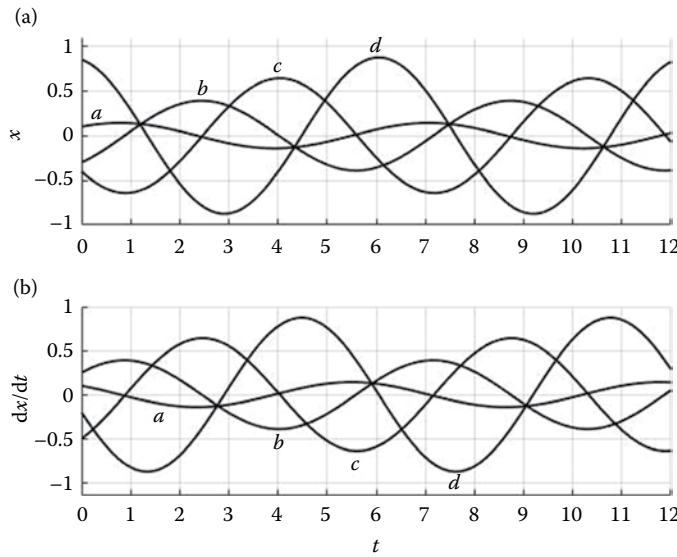


FIGURE 2.47 Sustained oscillations of x (a) and dx/dt (b) for undamped second-order systems.

Time histories of the state variables are shown in Figure 2.49. In contrast to the sinusoidal oscillations of the LTI system governed by Equation 2.163, the oscillations of the nonlinear system in Equations 2.164 and 2.165 are not of a sinusoidal nature.

Another type of sustained oscillation is possible for an unforced nonlinear system. In this case, there is a single closed orbit in the state space independent of the initial conditions. If the initial state is located on this closed path, the state vector remains on it forever, periodically returning to the starting point. When the initial state is inside the closed curve, the state trajectory may be asymptotically attracted to the closed curve or repelled from it towards a stable equilibrium point in its interior. Should the initial state be located outside the closed curve, the state trajectory either converges to it in a finite time period or else spirals outward from it.

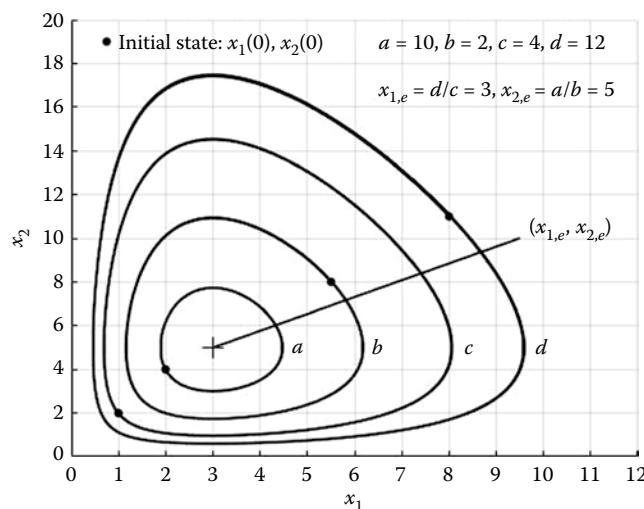


FIGURE 2.48 Closed orbits and sustained oscillations for the nonlinear system.

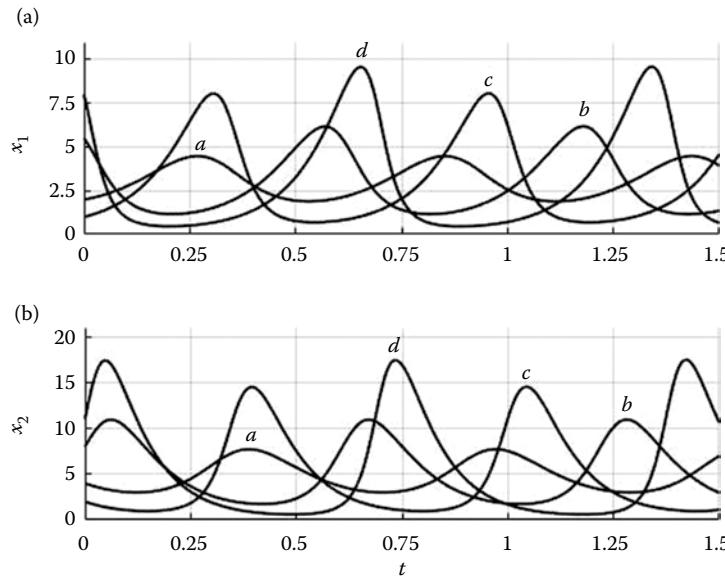


FIGURE 2.49 Sustained oscillations of x_1 (a) and x_2 (b) for nonlinear second-order system.

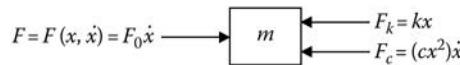


FIGURE 2.50 An autonomous nonlinear system with self-excitation force.

Sustained oscillations of this nature are called limit cycles. If the initial state is not on the limit cycle, the state trajectory is either attracted to or repelled from it. Limit cycles are either stable or unstable depending on which of the two situations applies.

An autonomous mechanical system with a stable limit cycle is given in Tse et al. (1963). Referring to Figure 2.50, the mass m is acted upon by a linear spring force F_k , a nonlinear damping force F_c , and a self-excitation force F , that is, a force with explicit dependence solely on the internal state of the system.

Note that there are no external forces present. The differential equation model is

$$m\ddot{x} = F - F_c - F_k = F_0\dot{x} - (cx^2)\dot{x} - kx \quad (2.166)$$

$$\Rightarrow m\ddot{x} + (cx^2 - F_0)\dot{x} + kx = 0 \quad (2.167)$$

The effective damping force is $(cx^2 - F_0)\dot{x}$. In the neighborhood of the equilibrium point $x = 0$, $\dot{x} = 0$, the term $(cx^2 - F_0) < 0$. The negative damping results in an increase of energy in the system making the equilibrium point inherently unstable. Consequently, the state trajectory will move outwards from the origin in state space.

The reverse is true whenever $(cx^2 - F_0) > 0$. In this case, the damping term is positive and energy is dissipated from the system. The state trajectory spirals inward to points where the total energy in the system is lower. Clearly, a locus of points must exist in state space to function as a transition between the two phenomena. The locus must be a closed curve, namely, the limit cycle.

EXAMPLE 2.12

For the mechanical system described by Equation 2.167,

- Convert the system model to state variable form.
- Numerical values of the system parameters are $m = 1$, $k = 2$, $c = 0.5$, and $F_0 = 3$. Approximate the state derivatives numerically with appropriate step size to determine the state trajectories when the initial state is located at
 - $x(0) = -1, \dot{x}(0) = -5$
 - $x(0) = 2, \dot{x}(0) = 5$
 - $x(0) = -2, \dot{x}(0) = 15$
 - $x(0) = 5, \dot{x}(0) = -20$
- Plot the trajectories in the state space.
- Estimate the period of the limit cycle.

- a. Choosing the state vector as $x_1 = x, x_2 = \dot{x}$ yields the state derivative functions

$$\dot{x}_1 = f_1(x_1, x_2) = x_2 \quad (2.168)$$

$$\dot{x}_2 = f_2(x_1, x_2) = -\frac{1}{m}[kx_1 + (cx_1^2 - F_0)x_2] \quad (2.169)$$

- b. Replacing \dot{x}_1 and \dot{x}_2 by difference quotients leads to the following difference equations for the discrete-time system

$$x_{1,A}(n+1) = x_{1,A}(n) + Tf_1[x_{1,A}(n), x_{2,A}(n)] \quad (2.170)$$

$$= x_{1,A}(n) + Tx_{2,A}(n) \quad (2.171)$$

$$x_{2,A}(n+1) = x_{2,A}(n) + Tf_2[x_{1,A}(n), x_{2,A}(n)] \quad (2.172)$$

$$= x_{2,A}(n) - \frac{T}{m} \left[kx_{1,A}(n) + \left\{ cx_{1,A}^2(n) - F_0 \right\} x_{2,A}(n) \right] \quad (2.173)$$

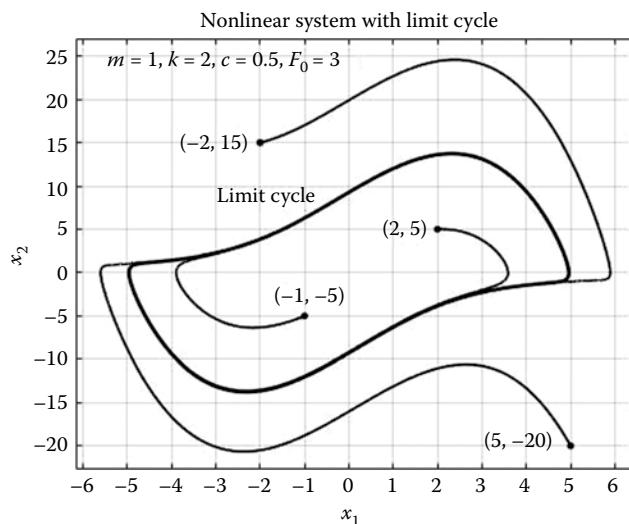


FIGURE 2.51 Approaches to limit cycle from several initial states.

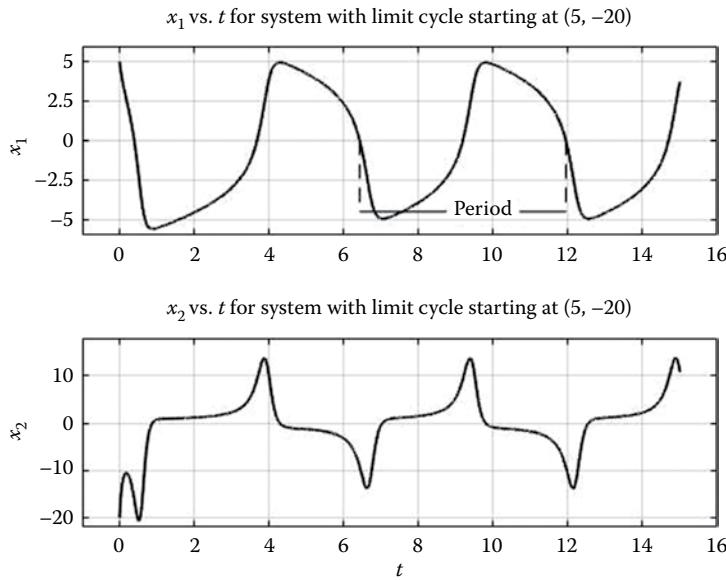


FIGURE 2.52 Time histories of state components (initial state: $x_1(0) = 5$, $x_2(0) = -20$).

The difference equations are solved recursively in “Ch2_Ex2_12.m” for the given initial states. The limit cycle and the four state trajectories are shown in Figure 2.51. As expected, the state trajectories eventually converge to the limit cycle.

- c. Figure 2.52 shows the time responses for the state components starting from the initial state $x_1(0) = 5$, $x_2(0) = -20$. The period of sustained oscillations can be approximated from the graph by estimating the difference in successive zero crossings of either state component once the state “locks into” the limit cycle. By zooming in on Figure 2.52, the period is approximated as $11.94 - 6.43 = 5.51$. Can you determine the approximate time the state enters the limit cycle?

EXERCISES

- 2.22 Examine the effect of changing the initial condition on the unit step response of the nonlinear system

$$\frac{dx}{dt} + x^2 = u, \quad u(t) = 1, \quad \geq 0$$

- Plot $x_A(n)$, $n = 0, 1, 2, 3, \dots$ when $x(0) = -2, -1, 0, 1, 5$ on the same graph. Use $T = 0.05$.
- 2.23 In Example 2.10, suppose instead of a constant friction force applied to the object as it slides, there is a variable friction force given by

$$f_\mu = \alpha v^\beta$$

Find and plot $v_A(n)$, $n = 1, 2, 3, \dots$ in response to the force $f(t)$ in Example 2.10 when

- i. $\alpha = 2, \beta = 0.5$
- ii. $\alpha = 1, \beta = 1$
- iii. $\alpha = 2, \beta = 2$

- 2.24 Nonlinear dynamic system is shown in [Figure E2.24](#). The input $u(t) = \sin 100 \pi t$, $t \geq 0$.
- Is the output $y(t)$ a sinusoidal function of the same frequency as the input like it would be in a linear system? Explain
 - Is the output $y(t)$ a periodic function? If so, what is the frequency?

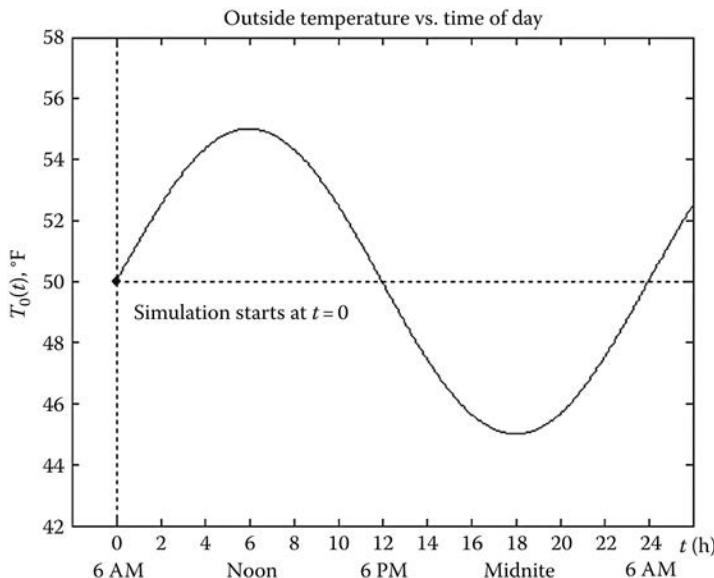
$$u(t) \longrightarrow \boxed{y^{1/2} = u} \longrightarrow y(t)$$

FIGURE E2.24

- 2.25 In Example 2.10, find the displacement of the mass, $x(t)$, $t \geq 0$.
 2.26 In Example 2.10, the applied force is

$$f(t) = \begin{cases} 2f_B \sin 0.25\pi t, & 0 \leq t < 4 \\ 0, & t \geq 4 \end{cases}$$

- Formulate a difference equation for $v_A(n)$ similar to Equation 2.142 and solve recursively.
 - Determine the analytical solution for $v(t)$.
 - Plot the approximate and analytical solutions on the same graph.
- 2.27 In Example 2.11, find the exact and approximate solutions for the building temperature $T(t)$ and furnace output $Q(t)$ if the desired setting T_d and tolerance Δ are
- $T_d = 72^\circ\text{F}$, $\Delta = 3^\circ\text{F}$
 - $T_d = 78^\circ\text{F}$, $\Delta = 1.5^\circ\text{F}$
- 2.28 In Example 2.11, investigate the effect of lowering the desired temperature T_d on the thermostat and its effect on the furnace cycle time and the duty cycle, that is, percentage of time the furnace is on. Plot the results for $T_d = 68^\circ\text{F}, 69^\circ\text{F}, \dots, 75^\circ\text{F}$.
- 2.29 In Example 2.11, suppose the outside temperature $T_0(t)$ varies in a sinusoidal fashion with average value of 50°F and amplitude of 5°F as shown in [Figure E2.29](#). The thermal capacitance of the room is $C = 500 \text{ Btu}/^\circ\text{F}$. The initial room temperature at 6 AM is 50°F .

**FIGURE E2.29**

- The thermostat is set at $T_d = 65^\circ\text{F}$. Simulate the building temperature long enough to show several cycles of $T(t)$ after the initial transient response vanishes.

- b. Find the energy cost per day if the cost of heating is 1.75¢ per 1000 Btu's.
 c. Repeat parts (a) and (b) for $T_d = 70^\circ\text{F}$ and 75°F .

2.30 The coulomb damping force acting on the mass shown in [Figure E2.30](#) is given by $f_\mu = -\text{sgn}(\dot{x}) \mu mg$. The initial condition is $x(0) = x_0 = -1 \text{ m}$, $\dot{x}(0) = 0 \text{ m/s}$. The equation of motion is

$$\ddot{x} + \omega_n^2 x = \frac{1}{m} f_\mu, \quad \omega_n^2 = \frac{k}{m} \quad (g = 9.81 \text{ m/s}^2)$$

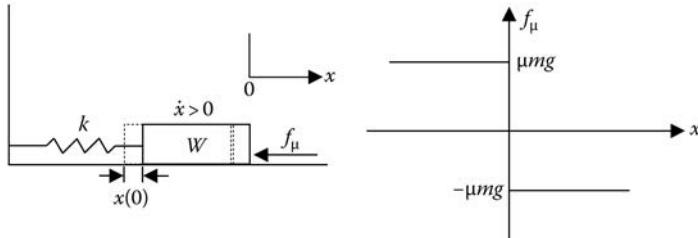


FIGURE E2.30

The system parameters are $m = 6 \text{ kg}$, $k = 300 \text{ N/m}$, $\mu = 0.2$

- a. Define the state as $x_1 = x$, $x_2 = \dot{x}$ and find difference equations for $x_{1,A}(n)$ and $x_{2,A}(n)$.
 b. Solve the difference equation for four cycles of $x_{1,A}(n)$ and $x_{2,A}(n)$ using a step size of $T = 0.001 \text{ s}$. Plot the results.
 c. The exact solution for $x(t)$ over the first cycle ($0 \leq t \leq t_2$) is

$$x(t) = \begin{cases} \left(x_0 + \frac{\mu mg}{k}\right) \cos \omega_n t - \frac{\mu mg}{k}, & 0 \leq t \leq t_1 \\ \left(x_1 + \frac{\mu mg}{k}\right) \cos \omega_n(t - t_1) + \frac{\mu mg}{k}, & t_1 \leq t \leq t_2 \end{cases}$$

where

$$t_1 = \frac{\pi}{\omega_n}$$

$$t_2 = t_1 + \frac{\pi}{\omega_n}$$

$$x_1 = -x_0 - 2 \frac{\mu mg}{k}$$

Plot the exact solution for $x(t)$ over the first cycle and compare it to the approximate solution.

2.31 For the undamped second-order system modeled by

$$\ddot{x} + \omega_n^2 x = 0 \text{ subject to } x(0) = x_0, \dot{x}(0) = \dot{x}_0$$

show the equation of the closed trajectories are ellipses in the $x - \dot{x}$ plane that reduce to the circular orbits in [Figure 2.40](#) when $\omega_n = 1 \text{ rad/s}$.

- 2.32 Generate the state trajectory shown in [Figure 2.45](#) starting at $(-2, 15)$ by finding an approximate solution to the differential equation

$$\frac{dx_2}{dx_1} = -\frac{1}{m} \left[k \frac{x_1}{x_2} + cx_1^2 - F_0 \right]$$

obtained as a result of dividing dx_2/dt in Equation 2.169 by dx_1/dt in Equation 2.168.

- 2.33 Generate 500 state trajectories starting from initial points randomly selected in the region $-10 \leq x(0) \leq 10$, $-10 \leq \dot{x}(0) \leq 10$ for the system governed by

$$m\ddot{x} + (F_0 - cx^2)\dot{x} + kx = 0$$

with the same parameter values as those in Example 2.12. Comment on the existence of a limit cycle and its effect on the trajectories.

- 2.34 Find the period of oscillations for the system modeled by

$$\begin{aligned}\dot{x}_1 &= x_1(10 - 2x_2) \\ \dot{x}_2 &= x_2(4x_1 - 12)\end{aligned}$$

when the initial state is (i) $x_1(0) = 10$, $x_2(0) = 15$ and (ii) $x_1(0) = 4$, $x_2(0) = 6$.

2.8 CASE STUDY: SUBMARINE DEPTH CONTROL SYSTEM

Automatic depth control of a submarine is the focus of this section. [Figure 2.53](#) illustrates a representative situation, where the actual depth of the submarine, denoted $c(t)$, is measured by a sensor and compared with the desired depth $r(t)$.

A simplified block diagram of the depth control system is shown in [Figure 2.54](#). The error signal $e(t)$ is the difference between the commanded depth $r(t)$ and the actual depth $c(t)$. It is fed back to the

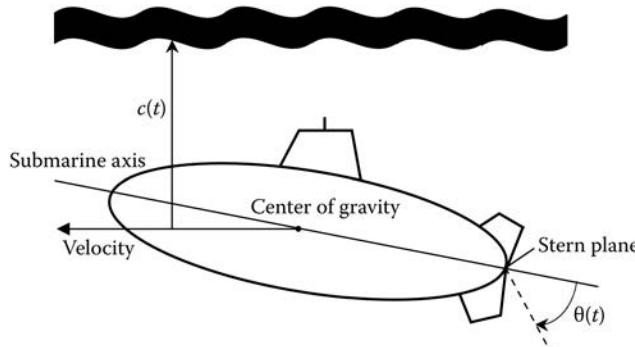


FIGURE 2.53 Depth control of a submarine.

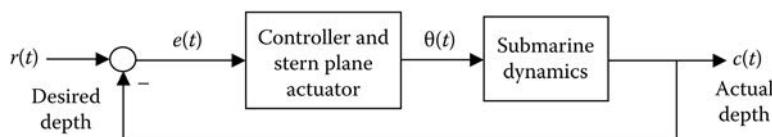


FIGURE 2.54 Block diagram of a submarine depth control system.

controller that sends a signal to the stern plane actuator motor that controls the stern plane actuator angle $\theta(t)$. The submarine depth responds to changes in the stern plane angle.

The controller and stern plane actuator combination are modeled by

$$\theta = K_C e + K_I \int e \, dt \quad (2.174)$$

and the submarine dynamics are approximated by the simple first-order model

$$\tau \frac{dv}{dt} + v = K_{\dot{\theta}} \frac{d\theta}{dt} + K_{\theta}\theta \quad (2.175)$$

where $v = v(t)$ is the depth rate of the submarine. Integrating the depth rate yields the depth of the submarine

$$c = \int v \, dt \quad (2.176)$$

The error signal is output from the summer as

$$e = r - c \quad (2.177)$$

Equations 2.174 through 2.177 constitute the mathematical model of the simplified submarine depth control system. The goal is to choose the parameters K_C and K_I , so that the submarine responds to step changes in commanded depth in an acceptable manner.

A simulation diagram of the control system is a useful first step in helping choose a set of state variables. Employing the technique discussed in Section 2.4 for drawing a simulation diagram with input derivative terms present, the diagram is shown in [Figure 2.55](#).

From the simulation diagram, the state equations are

$$\dot{x}_1 = v = K_{\theta}x_2 + K_{\dot{\theta}}\dot{x}_2 \quad (2.178)$$

$$= K_{\theta}x_2 + K_{\dot{\theta}}\left[\frac{1}{\tau}(\theta - x_2)\right] \quad (2.179)$$

The stern plane angle θ is expressible in terms of the states x_1 , x_2 , and x_3 and input r by

$$\theta = K_C e + K_I x_3 = K_C(r - x_1) + K_I x_3 \quad (2.180)$$

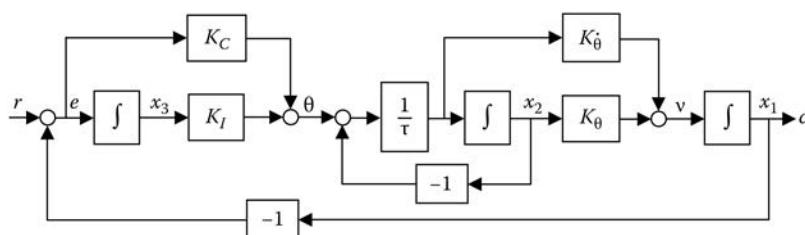


FIGURE 2.55 Simulation diagram of a submarine depth control system.

Combining Equations 2.179 and 2.180 gives

$$\dot{x}_1 = K_\theta x_2 + K_{\dot{\theta}} \left[\frac{1}{\tau} \{ K_C(r - x_1) + K_I x_3 - x_2 \} \right] \quad (2.181)$$

$$\Rightarrow \dot{x}_1 = \left(\frac{-K_{\dot{\theta}} K_C}{\tau} \right) x_1 + \left(K_\theta - \frac{K_{\dot{\theta}}}{\tau} \right) x_2 + \left(\frac{K_{\dot{\theta}} K_I}{\tau} \right) x_3 + \left(\frac{K_{\dot{\theta}} K_C}{\tau} \right) r \quad (2.182)$$

$$\dot{x}_2 = \left[\frac{1}{\tau} (\theta - x_2) \right] \quad (2.183)$$

$$= \left[\frac{1}{\tau} \{ K_C(r - x_1) + K_I x_3 - x_2 \} \right] \quad (2.184)$$

$$\Rightarrow \dot{x}_2 = \left(\frac{K_C}{\tau} \right) x_1 - \left(\frac{1}{\tau} \right) x_2 + \left(\frac{K_I}{\tau} \right) x_3 + \left(\frac{K_C}{\tau} \right) r \quad (2.185)$$

$$\dot{x}_3 = r - x_1 \quad (2.186)$$

Equations 2.182, 2.185, and 2.186 represent the dynamic portion of the state variable model, that is, $\dot{\underline{x}} = A\underline{x} + Br$. Choosing the outputs as $y_1 = \theta$, $y_2 = v$, and $y_3 = c$ determines the matrices C and D in the output equation $\underline{y} + C\underline{x} + Dr$.

$$y_1 = \theta = K_C(r - x_1) + K_I x_3 \quad (2.187)$$

$$= -K_C x_1 + K_I x_3 + K_C r \quad (2.188)$$

$$y_2 = v = \dot{x}_1 = \left(\frac{-K_{\dot{\theta}} K_C}{\tau} \right) x_1 + \left(K_\theta - \frac{K_{\dot{\theta}}}{\tau} \right) x_2 + \left(\frac{K_{\dot{\theta}} K_I}{\tau} \right) x_3 + \left(\frac{K_{\dot{\theta}} K_C}{\tau} \right) r \quad (2.189)$$

$$y_3 = c = x_1 \quad (2.190)$$

In summary, the state equations are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \frac{-K_{\dot{\theta}} K_C}{\tau} & K_\theta - \frac{K_{\dot{\theta}}}{\tau} & \frac{K_{\dot{\theta}} K_I}{\tau} \\ -K_C & -1 & K_I \\ \frac{\tau}{\tau} & \frac{\tau}{\tau} & \frac{\tau}{\tau} \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{K_{\dot{\theta}} K_C}{\tau} \\ \frac{K_C}{\tau} \\ 1 \end{bmatrix} r \quad (2.191)$$

$$\begin{bmatrix} \theta \\ v \\ c \end{bmatrix} = \begin{bmatrix} -K_C & 0 & K_I \\ \frac{-K_{\dot{\theta}} K_C}{\tau} & K_\theta - \frac{K_{\dot{\theta}}}{\tau} & \frac{K_{\dot{\theta}} K_I}{\tau} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} K_C \\ \frac{K_{\dot{\theta}} K_C}{\tau} \\ 0 \end{bmatrix} r \quad (2.192)$$

The exact solution for the outputs θ , v , and c in response to a given depth command r can be approximated by solving a system of difference equations obtained using the same approach we employed on previous occasions, that is, the first derivatives \dot{x}_1 , \dot{x}_2 , \dot{x}_3 in Equation 2.191 are replaced by first-order difference quotients, and the resulting difference equations are solved recursively for $x_{1,A}(n)$, $x_{2,A}(n)$, $x_{3,A}(n)$. The result is

$$\begin{bmatrix} \frac{1}{T} \{x_{1,A}(n+1) - x_{1,A}(n)\} \\ \frac{1}{T} \{x_{2,A}(n+1) - x_{2,A}(n)\} \\ \frac{1}{T} \{x_{3,A}(n+1) - x_{3,A}(n)\} \end{bmatrix} = \begin{bmatrix} -K_\theta K_C & K_\theta - \frac{K_\dot{\theta}}{\tau} & \frac{K_\dot{\theta} K_I}{\tau} \\ \frac{-K_C}{\tau} & -1 & \frac{K_I}{\tau} \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1,A}(n) \\ x_{2,A}(n) \\ x_{3,A}(n) \end{bmatrix} + \begin{bmatrix} \frac{-K_\dot{\theta} K_C}{\tau} \\ \frac{K_C}{\tau} \\ 1 \end{bmatrix} r(n) \quad (2.193)$$

The difference equations are updated according to

$$\begin{aligned} x_{1,A}(n+1) &= x_{1,A}(n) - \left(\frac{K_\dot{\theta} K_C T}{\tau} \right) x_{1,A}(n) + \left(K_\dot{\theta} - \frac{K_\dot{\theta}}{\tau} \right) T x_{2,A}(n) \\ &\quad + \left(\frac{K_\dot{\theta} K_I T}{\tau} \right) x_{3,A}(n) + \left(\frac{K_\dot{\theta} K_C T}{\tau} \right) r(n) \end{aligned} \quad (2.194)$$

$$x_{2,A}(n+1) = x_{2,A}(n) - \left(\frac{K_C T}{\tau} \right) x_{1,A}(n) - \left(\frac{T}{\tau} \right) x_{2,A}(n) + \left(\frac{K_I T}{\tau} \right) x_{3,A}(n) + \left(\frac{K_C T}{\tau} \right) r(n) \quad (2.195)$$

$$x_{3,A}(n+1) = x_{3,A}(n) - T x_{1,A}(n) + T r(n) \quad (2.196)$$

From Equation 2.192, the discrete-time outputs are

$$\begin{bmatrix} \theta_A(n) \\ v_A(n) \\ c_A(n) \end{bmatrix} = \begin{bmatrix} -K_C & 0 & K_1 \\ \frac{-K_\dot{\theta} K_C}{\tau} & K_\theta - \frac{K_\dot{\theta}}{\tau} & \frac{K_\dot{\theta} K_I}{\tau} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{1,A}(n) \\ x_{2,A}(n) \\ x_{3,A}(n) \end{bmatrix} + \begin{bmatrix} K_C \\ \frac{K_\dot{\theta} K_C}{\tau} \\ 0 \end{bmatrix} r(n) \quad (2.197)$$

Equations 2.194 through 2.197 are solved recursively in the M-file “*Ch2_CaseStudy.m*” for the case where $r(t) = 100$, $t \geq 0$. The baseline parameter values are

Sub dynamics: $\tau = 10$ s, $K_\dot{\theta} = 20$ ft/s per deg/s, $K_\theta = 10$ ft/s per deg

Controller gains: $K_C = 0.6$ deg/ft, $K_I = 0.1$ deg/ft s

Step size: $T = 0.0025$ s

Graphs of the discrete-time outputs $\theta_A(n)$, $v_A(n)$, $c_A(n)$ are shown in Figure 2.56. For clarity, every 100th value of discrete-time output is plotted.

The discontinuity in stern plane angle θ at $t = 0$ is a consequence of lumping the controller and stern plane actuator dynamics into a single equation as we did in Equation 2.174. The first term $K_C e$ is responsible for the direct (strictly algebraic) connection from the error e to the stern plane angle θ and ultimately from r to θ in Figure 2.55. The discontinuity is calculated from

$$\theta(0) = K_c e(0) = K_C [r(0) - c(0)] = 0.6 \text{ deg/ft} \times (100 \text{ ft} - 0) = 60 \text{ deg} \quad (2.198)$$

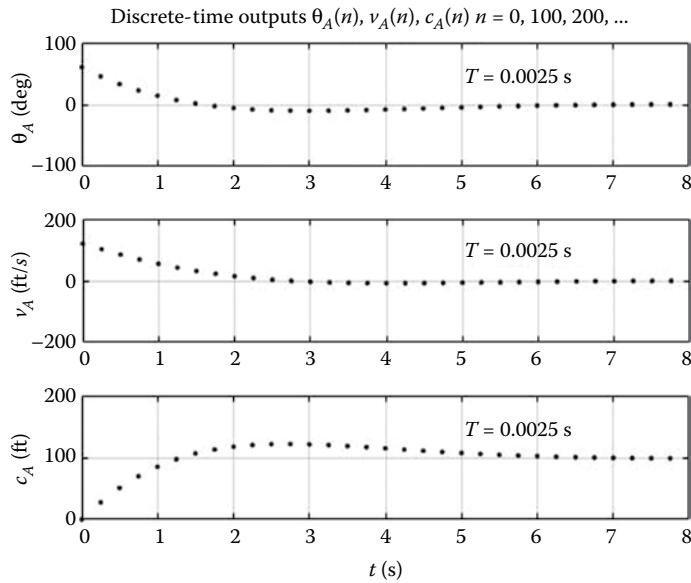


FIGURE 2.56 Discrete-time approximation of subdepth control system step response.

There is a direct connection from θ to v and, therefore, a direct path from r to v explaining the initial jump in depth rate as well. Figure 2.55 shows the term involving $K_{\dot{\theta}}$ in the sub dynamics is responsible for this. Exercise 2.36 presents an alternate representation of the stern plane actuator that eliminates the discontinuity in both θ and v .

EXERCISES

2.35 Suppose the model of the controller and stern plane actuator in Equation 2.174 is replaced by the following equation:

$$\theta = K_C e + K_I \int e(t) dt + K_D \frac{d}{dt} e(t)$$

The differential equation relating the control system output $c(t)$ and input $r(t)$ is

$$\begin{aligned} a_3 \ddot{c} + a_2 \dot{c} + a_1 c + a_0 r &= b_3 r + b_2 \ddot{r} + b_1 \dot{r} + b_0 \ddot{r} \\ a_3 &= \tau + K_D K_{\dot{\theta}} b_3 = K_D K_{\dot{\theta}} \\ a_2 &= 1 + K_C K_{\dot{\theta}} + K_D K_{\dot{\theta}} \quad b_2 = K_C K_{\dot{\theta}} + K_D K_q \\ a_1 &= K_C K_{\theta} + K_I K_{\dot{\theta}} \quad b_1 = K_C K_{\theta} + K_I K_{\dot{q}} \\ a_0 &= K_I K_{\theta} \quad b_0 = K_I K_{\theta} \end{aligned}$$

- Draw a simulation diagram of the system with three integrators in series.
- Choose the state variables as $x_1 = z$, $x_2 = \dot{z}$, $x_3 = \ddot{z}$ where z , \dot{z} , \ddot{z} are the outputs of the integrators. Define the output as $y = c$. Find the matrices A , B , C , and D in the state equations.
- Find the difference equations for the discrete-time states $x_{1,A}(n+1)$, $x_{2,A}(n+1)$, $x_{3,A}(n+1)$ and discrete-time output $c_A(n)$ similar to Equations 2.194 through 2.197.

- d. Choose the same values for K_C and K_I used to generate [Figure 2.56](#) along with $K_D = 0$. Solve the difference equations recursively to obtain the sub response $y(n)$ for the same input $r(t)$ and compare your result with the graph for $c_A(n)$ in [Figure 2.56](#).
- e. Experiment with new values for K_C , K_p , and K_D . Plot the results for $c_A(n)$.
- 2.36 The controller and stern plane actuator are modeled separately as shown [Figure E2.36](#):

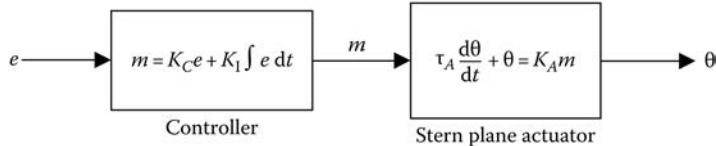


FIGURE E2.36

- a. Redraw the simulation diagram of the subdepth control system. Comment on whether m , θ , or v is discontinuous at $t = 0$ when the commanded depth $r(t)$ is a step input.
- b. Determine the state variables and find the new matrices A , B , C , and D in the state equations assuming the output vector $y = [m \ \theta \ v]^T$.

3 Elementary Numerical Integration

3.1 INTRODUCTION

Dynamic systems with continuous-time models in the form of differential equations possess memory. For systems with memory, knowledge of the system's inputs at a given point in time is insufficient to determine the state of the system at the same time. For example, a circuit with capacitors and inductors possesses memory. The instantaneous energy stored in the circuit is a function of the current state (capacitor voltages and inductor currents) which depends on the history of its sources (inputs) from the time when the complete state was last known.

An n th-order dynamic system with state variables x_1, x_2, \dots, x_n and inputs u_1, u_2, \dots, u_m is modeled by expressions for the state derivatives, that is

$$\dot{\underline{x}}(t) = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \vdots \\ \frac{dx_n}{dt} \end{bmatrix} = \underline{f}(\underline{x}, \underline{u}) \quad (3.1)$$

where

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad \underline{f}(\underline{x}, \underline{u}) = \begin{bmatrix} f_1(\underline{x}, \underline{u}) \\ f_2(\underline{x}, \underline{u}) \\ \vdots \\ f_n(\underline{x}, \underline{u}) \end{bmatrix} \quad (3.2)$$

In a formal sense, n distinct integrations are required to obtain the state \underline{x} , namely

$$x_1(t) = x_1(t_0) + \int_{t_0}^t f_1(\underline{x}, \underline{u}) dt \quad (3.3)$$

$$x_2(t) = x_2(t_0) + \int_{t_0}^t f_2(\underline{x}, \underline{u}) dt \quad (3.4)$$

$$\vdots$$

$$x_n(t) = x_n(t_0) + \int_{t_0}^t f_n(\underline{x}, \underline{u}) dt \quad (3.5)$$

For time-varying systems, a number of the system parameters are explicit functions of time. For example, the amount of fuel in a rocket or aircraft diminishes with time thereby affecting its dynamic properties. The state derivative vector is generally denoted by $f(t, \underline{x}, \underline{u})$, and Equations 3.3 through 3.5 are expressed as

$$x_1(t) = x_1(t_0) + \int_{t_0}^t f_1(t', \underline{x}, \underline{u}) dt' \quad (3.6)$$

$$x_2(t) = x_2(t_0) + \int_{t_0}^t f_2(t', \underline{x}, \underline{u}) dt' \\ \vdots \quad (3.7)$$

$$x_n(t) = x_n(t_0) + \int_{t_0}^t f_n(t', \underline{x}, \underline{u}) dt' \quad (3.8)$$

Equations 3.3 through 3.8 remind us that if we know the complete state at some initial time t_0 , then at some future time t , the state can be determined provided we know the inputs from t_0 to t . The n integrals to be evaluated in Equations 3.3 through 3.8 constitute the process of advancing or updating the state through time. This chapter looks at various alternatives for approximating these integrals.

3.2 DISCRETE-TIME SYSTEM APPROXIMATION OF A CONTINUOUS FIRST-ORDER SYSTEM

Simulation of dynamic systems modeled by first-order differential equations requires finding approximate solutions to the differential equations. A first-order, “continuous-time” (hereafter referred to as simply “continuous”) system is shown in [Figure 3.1](#), where

$u(t), t \geq 0$ is the input

$x(t), t \geq 0$ is the output (also referred to as the state)

$x(0)$ is the initial condition, $x(t)$ at $t = 0$

$f[x(t), u(t)]$ is a mathematical function encapsulating the system dynamics

Several examples of first-order, continuous-time systems are:

$$1. \text{ Integrator} \quad f[x(t), u(t)] = u(t) \quad (3.9)$$

$$2. \text{ Linear system} \quad f[x(t), u(t)] = ax(t) + bu(t) \quad (3.10)$$

$$3. \text{ Nonlinear system} \quad f[x(t), u(t)] = ax^2(t) + bu(t) \quad (3.11)$$

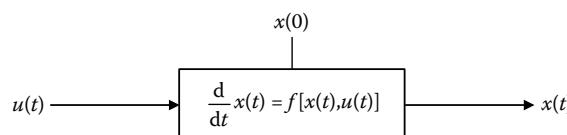


FIGURE 3.1 A first-order continuous system.

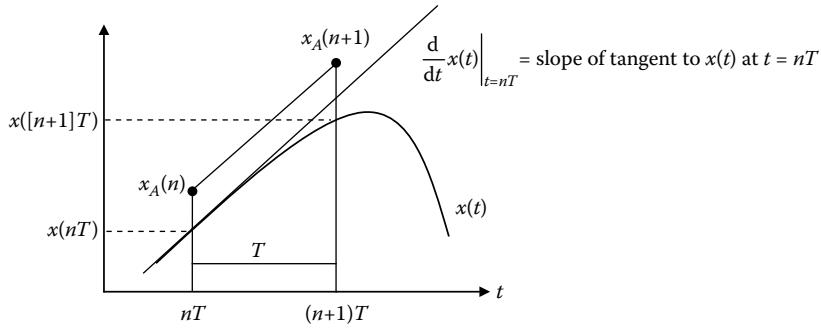


FIGURE 3.2 Diagram for obtaining difference equation to solve for $x_A(n)$, $n = 0, 1, 2, \dots$

Given the system model,

$$\frac{d}{dt} x(t) = f[x(t), u(t)] \quad (3.12)$$

an approximate solution for the state response $x(t)$, $t \geq 0$, in Equation (3.12) can be obtained at discrete points in time $t_n = nT$, $n = 0, 1, 2, \dots$. The approximate solution is $x_A(n)$, $n = 0, 1, 2, \dots$ where $x_A(n) \approx x(t_n) = x(nT)$, $n = 0, 1, 2, \dots$

Figure 3.2 illustrates the difference between $x(t)$, $t \geq 0$ and $x_A(n)$, $n = 0, 1, 2, \dots$. It is useful for deriving a difference equation which can be solved to generate $x_A(n)$, $n = 0, 1, 2, \dots$.

$$\text{At } t = nT, \quad \left. \frac{d}{dt} x(t) \right|_{t=nT} = f[x(t), u(t)] \Big|_{t=nT} = f[x(nT), u(nT)] \quad (3.13)$$

Approximating $\left. \frac{d}{dt} x(t) \right|_{t=nT}$ by the slope of the line connecting $x_A(n)$ and $x_A(n + 1)$,

$$\frac{x_A(n+1) - x_A(n)}{T} \approx f[x(nT), u(nT)] \quad (3.14)$$

Replacing $x(nT)$ with $x_A(n)$, and writing $u(nT)$ as $u(n)$ for short,

$$\frac{x_A(n+1) - x_A(n)}{T} = f[x_A(n), u(n)] \quad (3.15)$$

Note the use of the equality in Equation (3.15) enabling $x_A(n + 1)$ to be solved for as follows:

$$x_A(n+1) = x_A(n) + Tf[x_A(n), u(n)], \quad n = 0, 1, 2, \dots \quad (3.16)$$

Equation (3.16) is a difference equation. Given the initial condition $x_A(0)$, it is easily solved in a recursive manner.

EXAMPLE 3.1

To illustrate the use of Equation (3.16), consider the first-order system

$$\frac{d}{dt} x(t) + 2x(t) = u(t) \quad (3.17)$$

$$\frac{dx}{dt} = f(x, u) = -2x + u \quad (3.18)$$

Suppose the input $u(t) = 3t$, $t \geq 0$ and initial condition $x(0) = 1$.

Using Equation (3.16), the difference equation for obtaining $x_A(n)$, $n = 0, 1, 2, \dots$ is obtained as follows:

$$x_A(n+1) = x_A(n) + T[-2x_A(n) + u(n)], \quad n = 0, 1, 2, \dots \quad (3.19)$$

$$= (1 - 2T)x_A(n) + Tu(n), \quad n = 0, 1, 2, \dots \quad (3.20)$$

Letting $\alpha = (1 - 2T)$,

$$x_A(n+1) = \alpha x_A(n) + Tu(n), \quad n = 0, 1, 2, \dots \quad (3.21)$$

where

$$u(n) = u(nT) = u(t)|_{t=nT} = 3t|_{t=nT} = 3nT, \quad n = 0, 1, 2, \dots \quad (3.22)$$

$$x_A(n+1) = \alpha x_A(n) + T(3nT), \quad n = 0, 1, 2, \dots \quad (3.23)$$

$$= \alpha x_A(n) + 3nT^2, \quad n = 0, 1, 2, \dots \quad (3.24)$$

Starting with $n = 0$,

$$n = 0: \quad x_A(1) = \alpha x_A(0) + 3(0)T^2 \quad (3.25)$$

Choosing $x_A(0) = x(0)$ gives

$$n = 0: \quad x_A(1) = \alpha x(0) \quad (3.26)$$

$$n = 1: \quad x_A(2) = \alpha x_A(1) + 3(1)T^2 \quad (3.27)$$

$$= \alpha[\alpha x(0)] + 3T^2 \quad (3.28)$$

$$= \alpha^2 x(0) + 3T^2 \quad (3.29)$$

$$n = 2: \quad x_A(3) = \alpha x_A(2) + 3(2)T^2 \quad (3.30)$$

$$= \alpha[\alpha^2 x(0) + 3T^2] + 6T^2 \quad (3.31)$$

$$= \alpha^3 x(0) + 3\alpha T^2 + 6T^2 \quad (3.32)$$

$$= \alpha^3 x(0) + 3(\alpha + 2)T^2 \quad (3.33)$$

$$n = 3: \quad x_A(4) = \alpha x_A(3) + 3(3)T^2 \quad (3.34)$$

$$= \alpha [\alpha^3 x(0) + 3(\alpha + 2)T^2] + 9T^2 \quad (3.35)$$

$$= \alpha^4 x(0) + 3\alpha(\alpha + 2)T^2 + 9T^2 \quad (3.36)$$

$$= \alpha^4 x(0) + 3T^2 [\alpha(\alpha + 2) + 3] \quad (3.37)$$

The smaller the time step T , the closer $x_A(n)$, $n = 0, 1, 2, \dots$ will be to the exact solution $x(t)$, $t \geq 0$ at $t = 0, T, 2T, \dots$. Finding an approximate solution for the interval $0 \leq t \leq t_{final}$ requires t_{final}/T iterations in the recursive solution of the difference equation. There is a trade-off between accuracy of the approximate solution $x_A(n)$, $n = 0, 1, 2, \dots$ and the amount of computations required to generate it.

Equation 3.17 with the specified input $u(t) = 3t$ and initial condition $x(0) = 1$ is easily solved by analytical methods. The exact solution is given by

$$x(t) = 1.75e^{-2t} + 1.5t - 0.75, \quad t \geq 0 \quad (3.38)$$

Matlab program “Ch3_Ex3_1.m” solves Equation 3.21 recursively in addition to computing points on the exact solution for purposes of comparison. [Figure 3.3](#) shows the results for $x_A(n)$ and $x(t)$ when $T = 0.01, 0.025, 0.05$, and 0.1 .

Note, the approximate solution is known only at a discrete set of points and plotted accordingly. For purposes of clarity, a subset of the known discrete values are plotted with the exception for the case when $T = 0.1$. As expected, the accuracy of the approximate solution can be improved by decreasing the time step T at the expense of requiring additional calculations.

The approximate solutions are seen to converge to the exact solution as the time step decreases. Further it appears that setting the time step below $T = 0.025$ may not be necessary unless extreme accuracy is required.

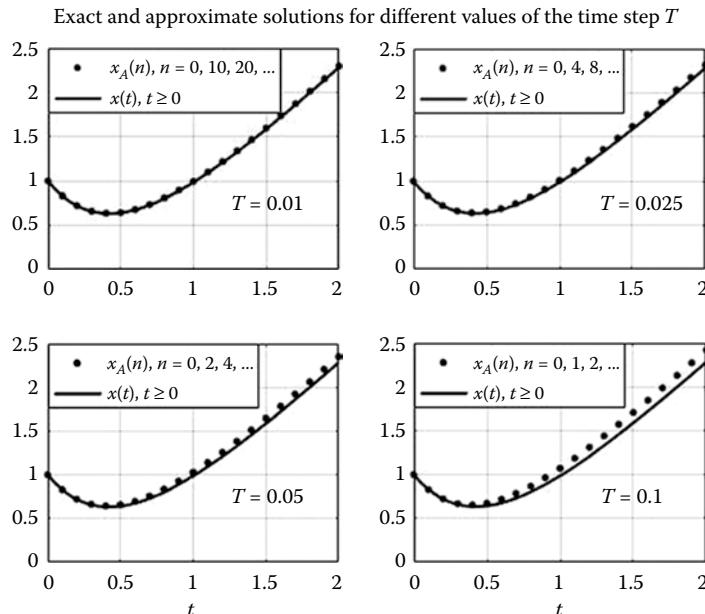


FIGURE 3.3 Exact and approximate solutions to Equation 3.17.

TABLE 3.1
Comparison of Exact and Approximate
($T = 0.01$) Solutions at Different Times

n	$t_n = nT$	$x_A(n)$	$x(t_n)$	$\frac{100}{x(t_n)} \times \left[\frac{x_A(n) - x(t_n)}{x(t_n)} \right]$
0	0	1	1	0
20	0.2	0.7233	0.7231	0.03
50	0.5	0.6468	0.6438	0.47
100	1.0	0.9951	0.9868	0.84
150	1.5	1.5988	1.5871	0.74
200	2.0	2.2955	2.2821	0.59

TABLE 3.2
Comparison of Exact and Approximate
($T = 0.1$) Solutions at Different Times

n	$t_n = nT$	$X_a(n)$	$X(t_n)$	$\frac{100}{X(t_n)} \times \left[\frac{X_A(n) - X(t_n)}{X(t_n)} \right]$
0	0	1	1	0
2	0.2	0.7240	0.7231	0.13
5	0.5	0.6743	0.6438	4.74
10	1.0	1.0718	0.9868	8.61
15	1.5	1.7063	1.5871	7.51
20	2.0	2.4148	2.2821	5.98

Table 3.1 compares the approximate solution for $T = 0.01$ to the exact solution at several discrete points in time. The last column contains the per cent error in the approximate solution,

$$\% \text{ Error} = 100 \times \left[\frac{x_A(n) - x(t_n)}{x(t_n)} \right] \quad (3.39)$$

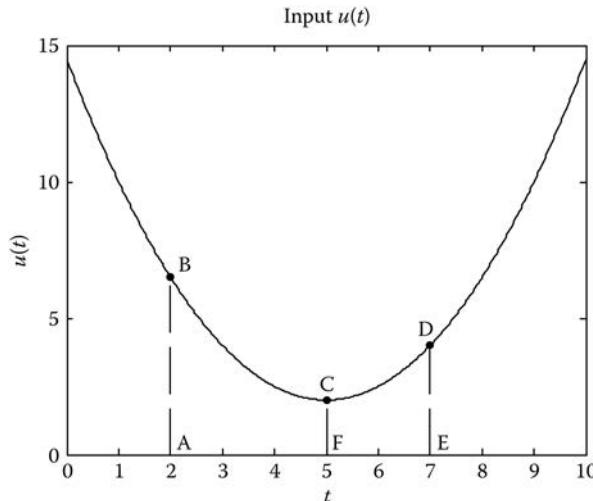
Table 3.2 compares the approximate solution for $T = 0.1$ to the exact solution at the same discrete points in time given in Table 1.1.

Note the percent error is roughly 10 times greater when the time step $T = 0.1$ compared to when $T = 0.01$.

EXERCISES

- 3.1 Consider the first-order system $\frac{dy}{dt} + a_0 y(t) = u(t)$
- Find the response of the system to a step input $u(t) = 1, t \geq 0$.
 - Find the response of the system to a ramp input $u(t) = t, t \geq 0$.

- c. In the limit as a_0 approaches zero, the first-order system reduces to a pure integrator. Show that the step and ramp responses in parts (a) and (b) approach $\int_0^t 1 \cdot dt$ and $\int_0^t t dt$, respectively.
- 3.2 The signal $u(t) = c_0 + c_1(t - t_0)^2$, $t \geq 0$ in [Figure E3.2](#) is input to a system governed by $dy/dt = u(t)$. i.e. a continuous integrator.

**FIGURE E3.2**

The change in output $y(t)$ from $t = t_0 - \Delta_1$ to $t = t_0 + \Delta_2$ is of interest. Using the following values: $c_0 = 2$, $c_1 = 1/2$, $t_0 = 5$, $\Delta_1 = 3$, $\Delta_2 = 2$ approximate the difference $y(t_0 + \Delta_2) - y(t_0 - \Delta_1)$ by

- Replacing $u(t)$ with a piecewise linear function $u_l(t)$ through pts B and C, and C and D and then integrating $u_l(t)$ between appropriate limits.
- As the areas of trapezoids ABCF and CDEF.
- Compare your answers in parts (a) and (b) to the true value

$$y(t_0 + \Delta_2) - y(t_0 - \Delta_1) = \int_{t_0 - \Delta_1}^{t_0 + \Delta_2} u(t) dt$$

- 3.3 A tank with cross-sectional area A_1 and resistance R_1 empties into a second tank with cross-sectional area A_2 . The first tank has no inflow and is initially filled to a height $h_1(0)$. The second tank is initially empty and has no outflow. The flow between the tanks is denoted by $f_1(t)$, and the tank levels are $h_1(t)$ and $h_2(t)$.
- Find the first-order differential equations for $f_1(t)$ and $h_2(t)$.
 - Show that the second tank is an integrator.
 - Find expressions for the transient responses of $f_1(t)$ and $h_2(t)$.
 - For system parameter values $A_1 = 100 \text{ ft}^2$, $R_1 = 0.25 \text{ ft per ft}^3/\text{min}$, $A_2 = 50 \text{ ft}^2$, and $h_1(0) = 20 \text{ ft}$, the responses $f_1(t)$ and $h_2(t)$ are plotted in [Figure E3.3](#). Estimate the level in tank 2 after 50 min by approximating the area under $f_1(t)$, $0 \leq t \leq 50$ and dividing by A_2 . Approximate the area using simple geometric shapes like rectangles and trapezoids.
 - Compare your answer from part (d) with the true value $h_2(50)$.

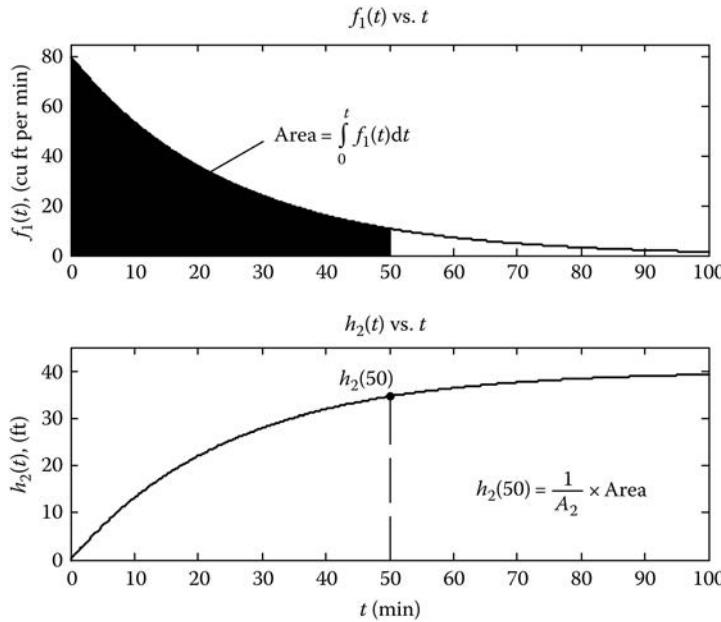


FIGURE E3.3

3.3 EULER INTEGRATION

An alternate derivation of Equation 3.16 for obtaining the approximate solution $x_A(n)$, $n = 0, 1, 2, \dots$ to Equation 3.12 will now be presented. The integral solutions in Equations 3.3–3.5 reduce to a single equation when the state vector $[x_1(t), x_2(t), \dots, x_n(t)]$ reduces to the scalar $x(t)$ and the initial time $t_0 = 0$, namely

$$x(t) = x(0) + \int_0^t f(x, u) dt \quad (3.40)$$

For $t = (n + 1)T$,

$$x[(n + 1)T] = x(0) + \int_0^{(n+1)T} f(x, u) dt \quad (3.41)$$

Thinking of the integral in Equation 3.41 as an area bounded by the function $f[x(t), u(t)]$ and the t -axis, Equation 3.41 can be written as

$$x[(n + 1)T] = x(0) + \int_0^{nT} f(x, u) dt + \int_{nT}^{(n+1)T} f(x, u) dt \quad (3.42)$$

$$= x(nT) + \int_{nT}^{(n+1)T} f(x, u) dt \quad (3.43)$$

The integral in Equation 3.43 is equal to the area under the derivative function $f[x(t), u(t)]$ between $t = nT$ and $t = (n + 1)T$. Various approximations to this area result in different numerical integrators.

3.3.1 EXPLICIT EULER INTEGRATION

The simplest approach is based on the assumption

$$f[x(t), u(t)] \approx f[x(nT), u(nT)] \quad \text{for } nT \leq t \leq (n+1)T \quad (3.44)$$

This assumption results in the true area $\int_{nT}^{(n+1)T} f(x, u) dt$ being approximated by the area of the rectangle shown in [Figure 3.4](#).

Replacing $\int_{nT}^{(n+1)T} f(x, u) dt$ in Equation 3.43 by the rectangular area $f[x(nT), u(nT)] T$ results in

$$x[(n+1)T] \approx x(nT) + T f[x(nT), u(nT)] \quad (3.45)$$

Denoting the approximation to $x(nT)$ by $x_A(nT)$ and dropping the “ T ” for simplicity, results in the difference equation

$$x_A(n+1) = x_A(n) + T f[x_A(n), u(n)] \quad (3.46)$$

Equation 3.46 is the difference equation of a discrete-time system. Recursive solution of this equation produces the discrete output $x_A(n)$ as the approximation to $x(nT)$ at discrete points in time. That is,

$$x_A(n) \approx x(nT), \quad n = 0, 1, 2, \dots \quad (3.47)$$

The first-order continuous system $\frac{d}{dt}x(t) = f[x(t), u(t)]$ and the discrete system approximation are illustrated in [Figure 3.5](#).

In the case of a simple integrator,

$$f[x(t), u(t)] = u(t) \quad (3.48)$$

and Equation 3.46 reduces to

$$x_A(n+1) = x_A(n) + Tu(n) \quad (3.49)$$

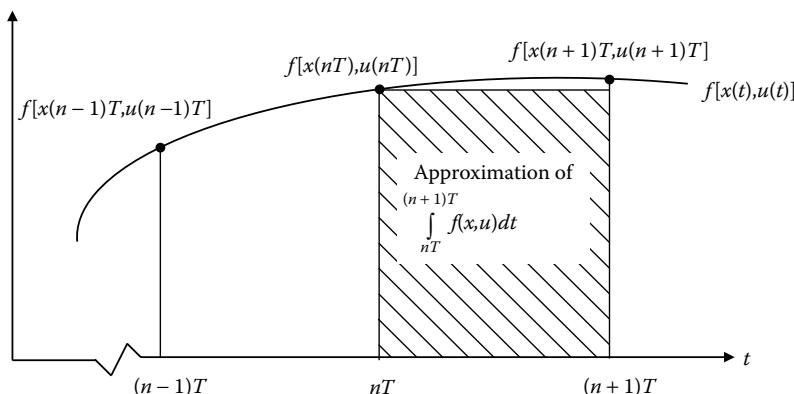


FIGURE 3.4 Graph showing an approximation of the integral in Equation 3.43.

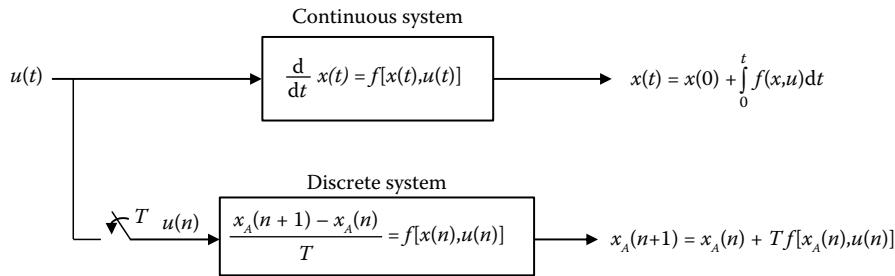


FIGURE 3.5 A first-order continuous system and a discrete system approximation.

Equation 3.49 is the difference equation for a numerical integrator known as an Euler integrator. It is also referred to as rectangular integration because of the rectangular approximation to the true area under the derivative function in Equation 3.48.

3.3.2 IMPLICIT EULER INTEGRATION

Another way of approximating the area under the derivative function $f[x(t), u(t)]$ is to assume

$$f[x(t), u(t)] \approx f[x(n+1)T, u(n+1)T] \quad \text{for } nT \leq t \leq (n+1)T \quad (3.50)$$

Using the same reasoning that resulted in Equation 3.46, the difference equation becomes

$$x_A(n+1) = x_A(n) + T f[x_A(n+1), u(n+1)] \quad (3.51)$$

For a simple integrator, Equation 3.51 reduces to

$$x_A(n+1) = x_A(n) + T u(n+1) \quad (3.52)$$

While both Equations 3.46 and 3.51 are difference equations based on Euler (or rectangular) integration, there is a fundamental difference between them. Equation 3.46 is explicit in nature, which means a recursive solution, like the one in Example 3.1, is straightforward. Equation 3.51 is an implicit equation as a result of $x_A(n+1)$ appearing on both sides of the equation. The solution to Equation 3.51 can be challenging depending on the nature of the function $f[x(t), u(t)]$. In all but the simplest cases, some form of iterative solution to a nonlinear algebraic equation is required to update the state from $x_A(n)$ to $x_A(n+1)$.

Equation 3.46 is the difference equation for explicit Euler integration, also known as forward rectangular integration. Equation 3.51 is the difference equation for implicit Euler integration, also known as backward rectangular integration. Both types are illustrated in Example 3.2.

EXAMPLE 3.2

The input to a continuous integrator is $u(t) = \sin 2\pi t$, $t \geq 0$. Compare the output of the continuous integrator to the outputs from explicit and implicit Euler numerical integrators at $t = 0.4$. Choose the integration step size $T = 0.005, 0.01, 0.025, 0.05$.

The output of the continuous integrator with input $u(t) = A \sin \omega t$ is

$$x(t) = \int_0^t u(t) dt = \int_0^t A \sin \omega t dt = A \left[-\frac{1}{\omega} \cos \omega t \right]_0^t = \frac{A}{\omega} (1 - \cos \omega t) \quad (3.53)$$

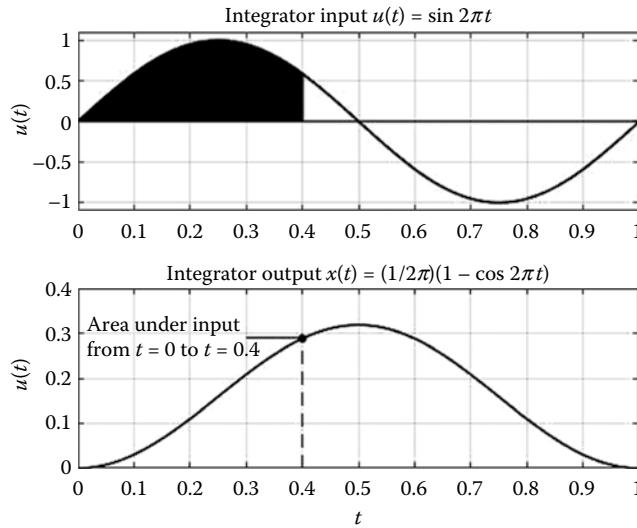


FIGURE 3.6 Continuous integrator with input $u(t) = \sin 2\pi t$ and output $x(t) = (1 - \cos 2\pi t)/2\pi$.

For $A = 1$, $\omega = 2\pi$ rad/s,

$$x(t) = \frac{1}{2\pi}(1 - \cos 2\pi t) \quad (3.54)$$

The continuous input $u(t)$ and the integrator output are graphed in Figure 3.6

$$x(0.4) = \left. \frac{1}{2\pi}(1 - \cos 2\pi t) \right|_{t=0.4} = 0.2879 \quad (3.55)$$

Equations 3.49 and 3.52 are solved recursively in “Ch3_Ex3_2.m” producing the values shown in Table 3.3.

According to Equation 3.49, the Euler integrator simply adds a rectangular area $Tu(n)$ to the current state $x_A(n)$ to produce the updated state $x_A(n + 1)$. A general formula for $x_A(n + 1)$ is easily obtained by observing

$$x_A(1) = x_A(0) + Tu(0) \quad (3.56)$$

TABLE 3.3
Comparison of Explicit and Implicit Euler Integrators

T	$n_{final} = \frac{0.4}{T}$	Explicit Euler		% Error	
		$x_A(n_{final})$	$100 \times \left[\frac{x_A(n_{final}) - x(0.4)}{x(0.4)} \right]$	$x_A(n_{final})$	$100 \times \left[\frac{x_A(n_{final}) - x(0.4)}{x(0.4)} \right]$
0.005	80	0.2864	-0.5186	0.2894	0.5022
0.01	40	0.2849	-1.0537	0.2908	0.9879
0.025	16	0.2800	-2.7576	0.2947	2.3462
0.05	8	0.2708	-5.9276	0.3002	4.2800

$$x_A(2) = x_A(1) + Tu(1) \quad (3.57)$$

$$= [x_A(0) + Tu(0)] + Tu(1) \quad (3.58)$$

$$= x_A(0) + T[u(0) + u(1)] \quad (3.59)$$

Replacing $x_A(0)$ with $x(0)$, the discrete output of an explicit Euler integrator is given by

$$x_A(n+1) = x(0) + T[u(0) + u(1) + u(2) + \dots + u(n-1) + u(n)] \quad (3.60)$$

$$= x(0) + T \sum_{k=0}^n u(k), \quad n = 0, 1, 2, \dots \quad (3.61)$$

By the same reasoning, the discrete output of an implicit Euler integrator is given by

$$x_A(n+1) = x(0) + T \sum_{k=0}^n u(k+1), \quad n = 0, 1, 2, \dots \quad (3.62)$$

EXERCISES

3.4 In Example 3.2

- a. Explain why the implicit Euler integrator produces higher estimates of the continuous-time integrator output than the explicit Euler integrator. Is this true in general?
- b. Find $x_A(5)$ for both numerical integrators and compare the results to $x(0.25)$. Explain why both integrators incur the maximum error $|x(nT) - x_A(n)|$ for $n = 5$.
- c. Repeat Examples 3.2 and 3.3 with a step size $T = 0.01$. Enter the numerical results for $n = 0, 5, 10, \dots, 50$ in a table rounded to 4 places after the decimal point.

- 3.5 The RC circuit shown in [Figure E3.5](#) is a first order low pass filter. The differential equation relating the output voltage $v_0(t)$ and input voltage $v_i(t)$ is

$$RC \frac{dv_0}{dt} + v_0 = v_i$$

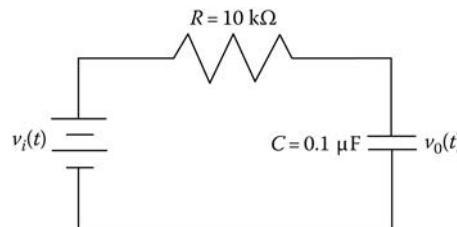
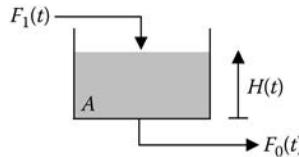


FIGURE E3.5

The capacitor is initially uncharged. A discrete-time integrator is used to approximate the continuous output $v_0(t)$ when the input $v_i(t)$ is an AC signal $\sin \omega t$.

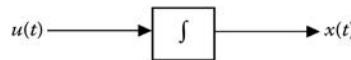
- a. Find the difference equation used to obtain $v_{0,A}(n)$ if forward Euler integration is used with a step size of T .

- b. For $v_i(t) = \sin \omega t$, find and plot $v_{0,A}(n)$ corresponding to $0 \leq n \leq 4\pi/\omega T$ when
- $\omega = 100 \text{ rad/s}, T = RC/10$
 - $\omega = 1000 \text{ rad/s}, T = RC/100$
- 3.6 The flow out of the tank shown in Figure E3.6 is given by $F_0 = cH^{1/2}$. The cross-sectional area of the tank $A = 50 \text{ ft}^2$ and the constant $c = 2 \text{ ft}^3/\text{min per ft}^{1/2}$. The tank is 25 ft in height and the initial level in the tank $H(0) = 16 \text{ ft}$.

**FIGURE E3.6**

- a. The flow into the tank is $F_1(t) = \bar{F}_1 = 10 \text{ ft}^3/\text{min}, t \geq 0$. Find the steady-state height of liquid in the tank, $H(\infty)$.
- b. Use forward Euler integration with a suitable step size and compare $\lim_{n \rightarrow \infty} H_A(n)$ to the result from part (a).
- c. The flow into the tank is $F_1(t) = 4 + (t/10), t \geq 0$. Use forward Euler integration with a step size T and find the difference equation for updating the state $H_A(n)$. Leave your answer in terms of c, A , and T .
- d. For the input flowrate in part (c), using forward Euler integration with $T = 0.1 \text{ min}$, find n_f , where $n_f T$ is the time required to fill the tank, i.e. $H_A(n_f - 1) < 25$ and $H_A(n_f) \geq 25$. Plot the results.
- 3.7 The input to the integrator shown in Figure E3.7 is the continuous-time signal $u(t) = 1/(t+1), t \geq 0$
- Find the difference equation for computing the state $x_A(n)$ recursively when implicit Euler integration with a step size T is used.
 - Find $x_A(1), x_A(2)$ and $x_A(3)$ if $T = 0.1$.
 - Compare your answer for $x_A(3)$ to the exact value $x(3T)$.

Note:
$$\int_0^t \frac{1}{t+1} dt = \ln(1+t)$$

**FIGURE E3.7**

- 3.8 Show that an approximate solution of the first-order continuous-time model

$$\frac{dx}{dt} = f(x, u)$$

based on replacing the derivative dx/dt with the finite difference $[x(n+1) - x(n)]/T$ is equivalent to using forward (explicit) Euler integration.

- 3.9 Consider the case of a liquid discharged from a tank at a rate proportional to the square root of the level in the tank. The continuous model is

$$A \frac{dH}{dt} + \alpha H^{1/2} = F_1$$

where $H = H(t)$ is the continuous tank level, $F_1 = F_1(t)$ is the flow in and α is a constant dependent on the physical characteristics of the tank.

- Use implicit Euler integration to find a difference equation involving the time signals $H_A(n)$ and $F_1(n)$ where $H_A(n) \approx H(nT)$ and $F_1(n) = F_1(nT)$. Write the equation in implicit form with $H_A(n+1)$ on both sides.
- Show that the implicit equation can be solved explicitly for $H_A(n+1)$ in terms of $H_A(n)$ and $F_1(n+1)$ by making the substitution $x = [H_A(n+1)]^{1/2}$ and solving the resulting quadratic equation in x .

3.4 TRAPEZOIDAL INTEGRATION

Of the numerical integrators, the Euler integrators are the simplest to implement. However, for a given integration step size they are also the least accurate. This is not necessarily a reason to choose another integrator since any desired level of accuracy is achievable with Euler integrators (in principle) simply by reducing the step size and performing additional calculations. Indeed, the simplicity of Euler integration is responsible for its widespread use in far ranging applications.

There may be circumstances which dictate the integration step size in a simulation study and thus compel the developer to consider other methods for approximating the dynamics of a continuous system. Accordingly, we shall investigate other formulas and algorithms for numerical integration.

Starting with Equation 3.43, repeated below

$$x[(n+1)T] = x(nT) + \int_{nT}^{(n+1)T} f(x, u) dt \quad (3.63)$$

Recall that approximating the derivative function $f(x, u)$, by a constant over the interval $nT \leq t \leq (n+1)T$ (see Figure 3.4) resulted in

$$x_A(n+1) = x_A(n) + T f[x_A(n), u(n)] \quad (3.64)$$

or

$$x_A(n+1) = x_A(n) + T f[x_A(n+1), u(n+1)] \quad (3.65)$$

Using a linear approximation to the derivative function $f(x, u)$ over the interval $nT \leq t \leq (n+1)T$ instead leads to a more accurate discrete system representation than either Equation 3.64 or Equation 3.65. Figure 3.7 illustrates the situation.

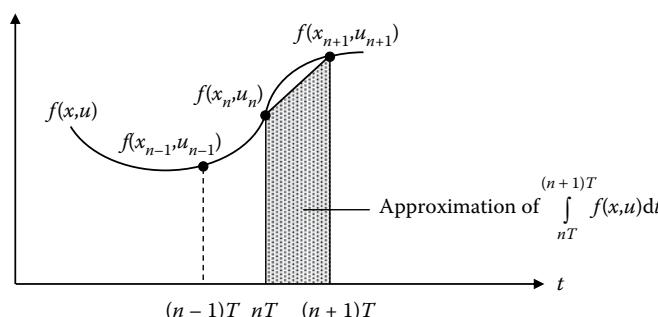


FIGURE 3.7 Trapezoidal approximation of area under $f(x, u)$, $nT \leq t \leq (n+1)T$.

Note $f(x_{n-1}, u_{n-1})$, $f(x_n, u_n)$ and $f(x_{n+1}, u_{n+1})$ are short for $f[x(n-1)T, u(n-1)T]$, $f[x(n)T, u(n)T]$ and $f[x(n+1)T, u(n+1)T]$, respectively. The equation of the line connecting the points $[nT, f(x_n, u_n)]$ and $[(n+1)T, f(x_{n+1}, u_{n+1})]$ is easily found. Integrating the linear equation over the interval $nT \leq t \leq (n+1)T$ results in the approximation to the integral in Equation 3.63 and the required difference equation.

A far simpler approach is based on recognizing the approximating area is a trapezoid leading directly to the difference equation

$$x_A(n+1) = x_A(n) + \frac{T}{2} \{f[x_A(n), u(n)] + f[x_A(n+1), u(n+1)]\} \quad (3.66)$$

In the case of a continuous integrator, $f(x, u) = u$, and Equation 3.66 reduces to

$$x_A(n+1) = x_A(n) + \frac{T}{2} [u(n) + u(n+1)] \quad (3.67)$$

Equation 3.67 is the difference equation of a trapezoidal integrator, so named because the area approximating $\int_{nT}^{(n+1)T} f[x(t), u(t)] dt = \int_{nT}^{(n+1)T} u(t) dt$ is a trapezoid.

EXAMPLE 3.3

The input to a continuous integrator is $u(t) = e^{-2t}$, $t \geq 0$. Find the difference equation for approximating the continuous output $x(t)$ at $t = 0.1, 0.2, \dots, 1.0$ using

- a. explicit Euler integration
- b. implicit Euler integration
- c. trapezoidal integration
- d. Choose the step size $T = 0.1$ and compute $x_A(n)$, $n = 0, 1, 2, \dots, 10$ to approximate $x(t)$ at $t = 0.1, 0.2, \dots, 1.0$.
- e. Compare the results to the continuous response $x(t)$ at $t = 0.1, 0.2, \dots, 1.0$.

The continuous and discrete integrators are shown in [Figure 3.8](#).

a. $u(n) = e^{-2nT}, \quad n = 0, 1, 2, \dots \quad (3.68)$

$$x_A(n+1) = x_A(n) + Tu(n) \quad (3.69)$$

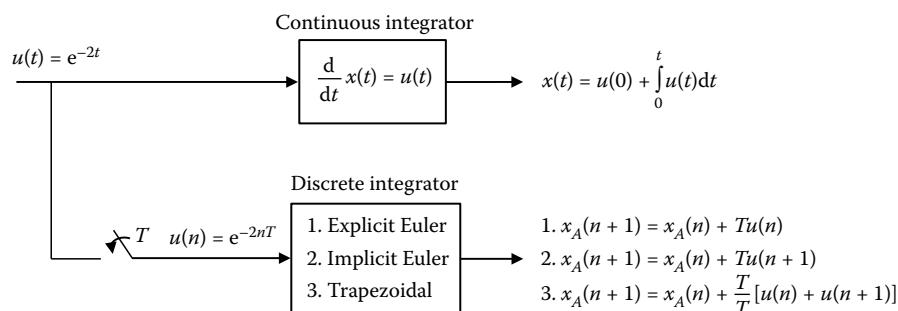


FIGURE 3.8 Continuous and discrete integrators for Example 3.3.

$$= x_A(n) + T e^{-2nT}, \quad n = 0, 1, 2, \dots \quad (3.70)$$

$$n = 0: \quad x_A(1) = x_A(0) + 0.1e^{-2(0)(0.1)} = 0 + 0.1e^0 = 0.1$$

$$n = 1: \quad x_A(2) = x_A(1) + 0.1e^{-2(1)(0.1)} = 0.1 + 0.1e^{-0.2} = 0.1819$$

$$n = 2: \quad x_A(3) = x_A(2) + 0.1e^{-2(2)(0.1)} = 0.1819 + 0.1e^{-0.4} = 0.2489$$

The remaining values are computed in “Ch3_Ex3_3.m” and shown in [Table 3.4](#).

b. $x_A(n+1) = x_A(n) + Tu(n+1) \quad (3.71)$

$$= x_A(n) + T e^{-2(n+1)T}, \quad n = 0, 1, 2, \dots \quad (3.72)$$

$$n = 0: \quad x_A(1) = x_A(0) + 0.1e^{-2(1)(0.1)} = 0 + 0.1e^{-0.2} = 0.0819$$

$$n = 1: \quad x_A(2) = x_A(1) + 0.1e^{-2(2)(0.1)} = 0.0819 + 0.1e^{-0.4} = 0.1489$$

$$n = 2: \quad x_A(3) = x_A(2) + 0.1e^{-2(3)(0.1)} = 0.1489 + 0.1e^{-0.6} = 0.2038$$

The remaining values are computed in “Ch3_Ex3_3.m” and shown in [Table 3.4](#).

c. $x_A(n+1) = x_A(n) + \frac{T}{2}[u(n)+u(n+1)] \quad (3.73)$

$$= x_A(n) + \frac{T}{2}[e^{-2nT} + e^{-2(n+1)T}], \quad n = 0, 1, 2, \dots \quad (3.74)$$

$$n = 0: \quad x_A(1) = x_A(0) + \frac{0.1}{2}[e^{-2(0)(0.1)} + e^{-2(1)(0.1)}] = 0 + 0.05[1 + e^{-0.2}] = 0.0909$$

$$n = 1: \quad x_A(2) = x_A(1) + \frac{0.1}{2}[e^{-2(1)(0.1)} + e^{-2(2)(0.1)}] = 0.0909 + 0.05[e^{-0.2} + e^{-0.4}] = 0.1654$$

$$n = 2: \quad x_A(3) = x_A(2) + \frac{0.1}{2}[e^{-2(2)(0.1)} + e^{-2(3)(0.1)}] = 0.1654 + 0.05[e^{-0.4} + e^{-0.6}] = 0.2263$$

The remaining values are computed in “Ch3_Ex3_3.m” and shown in [Table 3.4](#).

d. The output of the continuous integrator is

$$x(t) = \int_0^t e^{-2t'} dt' = \left[\frac{e^{-2t'}}{-2} \right]_0^t = \frac{1}{2}(1 - e^{-2t}) \quad (3.75)$$

[Table 3.4](#) compares the continuous and discrete outputs at $t_n = nT = 0, 0.1, 0.2, \dots, 1$.

For the same step size, the trapezoidal integrator is superior to the Euler integrators. An advantage of trapezoidal integration compared to Euler is the increased step size that can be used while maintaining comparable accuracy.

The following example illustrates the use of trapezoidal integration for a first-order system modeled by a differential equation with time varying parameters.

TABLE 3.4**Comparison of Output from Continuous and 3 Discrete Integrators for $u(t) = e^{-2t}$**

n	$t_n = nT$	Explicit Euler $x_A(n)$	Implicit Euler $x_A(n)$	Trapezoidal $x_A(n)$	Continuous $x(t_n)$
0	0.0	0.0	0.0	0.0	0.0
1	0.1	0.1000	0.0819	0.0909	0.0906
2	0.2	0.1819	0.1489	0.1654	0.1649
3	0.3	0.2489	0.2038	0.2263	0.2256
4	0.4	0.3038	0.2487	0.2763	0.2753
5	0.5	0.3487	0.2855	0.3171	0.3161
6	0.6	0.3855	0.3156	0.3506	0.3494
7	0.7	0.4156	0.3403	0.3780	0.3767
8	0.8	0.4403	0.3605	0.4004	0.3991
9	0.9	0.4605	0.3770	0.4187	0.4174
10	1.0	0.4770	0.3905	0.4338	0.4323

EXAMPLE 3.4

A nonlinear, time-varying dynamic system is modeled by the differential equation

$$t^2 \frac{dx}{dt} + x \frac{dx}{dt} + 2tx = u(t) \quad (3.76)$$

- Find the difference equation of the discrete system based on trapezoidal integration for approximating the response of the continuous system.
- Solve the difference equation for $x_A(n)$, $n = 0, 1, 2, \dots$ when the continuous input $u(t) = -3t^2/2$. The initial condition is $x(0) = 1$ and the step size $T = 0.01$.
- Plot the discrete response $x_A(n)$, $n = 0, 1, 2, \dots, 100$ and the continuous response

$$x(t) = -t^2 + (t^4 - t^3 + 1)^{1/2}, \quad 0 \leq t \leq 1 \quad (3.77)$$

on the same graph.

- Solving for the state derivative,

$$\frac{dx}{dt} = f(t, x, u) = \frac{1}{t^2 + x(t)} [u(t) - 2tx(t)] \quad (3.78)$$

The difference equation based on trapezoidal integration is

$$x_A(n+1) = x_A(n) + \frac{T}{2} [f\{nT, x_A(n), u(n)\} + f\{(n+1)T, x_A(n+1), u(n+1)\}] \quad (3.79)$$

$$\begin{aligned} &= x_A(n) + \frac{T}{2} \left\{ \frac{1}{(nT)^2 + x_A(n)} [u(n) - 2(nT)x_A(n)] \right. \\ &\quad \left. + \frac{1}{[(n+1)T]^2 + x_A(n+1)} [u(n+1) - 2(n+1)T]x_A(n+1) \right\} \end{aligned} \quad (3.80)$$

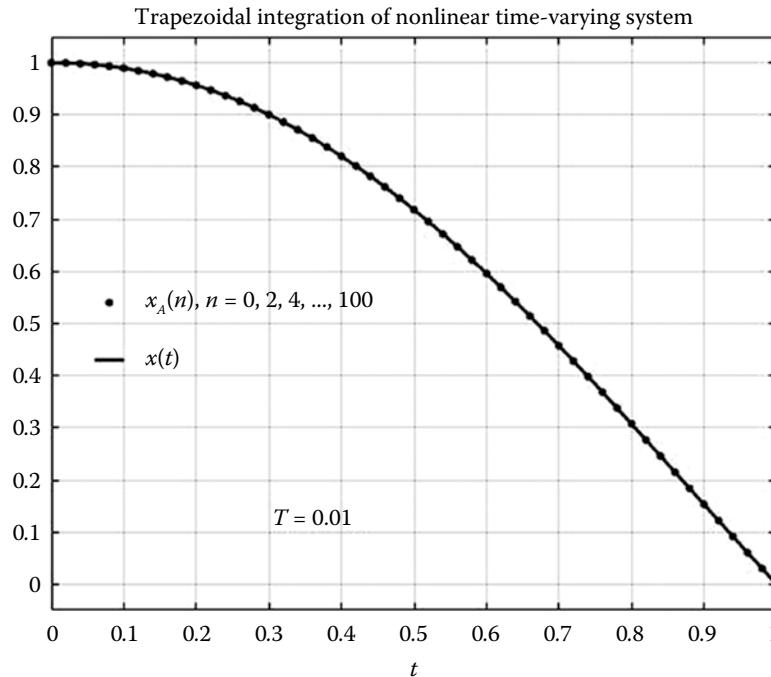


FIGURE 3.9 Graph of discrete (trapezoidal, $T = 0.01$) and continuous responses.

- b. Equation 3.80 is an implicit equation for $x_A(n + 1)$ which generally means some type of iterative, numerical root solving algorithm is required to find $x_A(n + 1)$ at each time step. This can increase the computational requirements dramatically, not to mention the additional programming required to implement the algorithm. In this example however, Equation 3.80 can be manipulated to produce a quadratic function of the form

$$a[x_A(n + 1)]^2 + bx_A(n + 1) + c = 0 \quad (3.81)$$

where a, b, c are expressible in terms of $u(n), x_A(n)$ and $u(n + 1)$, all of which can be calculated at time $t_n = nT$. "Ch3_Ex3_4.m" includes the statements to determine a, b, c and solve Equation 3.81 at each time step for the positive root.

- c. The discrete and continuous responses are shown in [Figure 3.9](#).

The discrete and continuous responses are indistinguishable from each other at times $t_n = nT$, $n = 0, 1, 2, \dots, 100$. The discrete signal $x_A(n)$ is defined solely at the discrete times $0, T, 2T, \dots$, which explains why discrete signals should always be plotted as discrete data points.

Consider finding a discrete approximation of the linear first-order system

$$\frac{d}{dt} x(t) + a_0 x(t) = b_0 u(t) \quad (3.82)$$

using trapezoidal integration.

Solving for the derivative function $f[x(t), u(t)]$,

$$f[x(t), u(t)] = \frac{dx}{dt} = b_0 u(t) - a_0 x(t) \quad (3.83)$$

$$x_A(n+1) = x_A(n) + \frac{T}{2} \left\{ f[x_A(n), u(n)] + f[x_A(n+1), u(n+1)] \right\} \quad (3.84)$$

$$= x_A(n) + \frac{T}{2} \left[\{b_0 u(n) - a_0 x_A(n)\} + \{b_0 u(n+1) - a_0 x_A(n+1)\} \right] \quad (3.85)$$

Equation 3.85 is implicit as a result of $x_A(n+1)$ appearing on both sides of the equation. However, owing to the linear nature of Equation 3.85, it is easily solved for $x_A(n+1)$,

$$x_A(n+1) = \frac{\left(1 - \frac{a_0 T}{2}\right)}{\left(1 + \frac{a_0 T}{2}\right)} x_A(n) + \frac{b_0 T}{2} u(n) + u(n+1) \quad (3.86)$$

EXAMPLE 3.5

The velocity $v = v(t)$ of an object sinking in a body of water is described by

$$\frac{dv}{dt} + \frac{cg}{W} v = \frac{g}{W} (W - F_B) \quad (3.87)$$

where W is the weight of the object; c is the drag coefficient; F_B is the buoyant force; g is the gravitational constant (32.2 ft/s^2).

The buoyant force is a constant which equals the weight of the volume of water displaced by the object. The object is a drum full of hazardous materials [Braun] weighing 350 lb and its volume is such that the buoyant force is 275 lb. The drag coefficient c was determined experimentally to be 0.8 lb/(ft/s) . The drum is released at the surface with zero velocity.

- Find a difference equation based on trapezoidal integration to approximate the dynamics of the sinking drum.
- Find the approximate velocity, $v_A(n)$, $n = 0, 10, 20, \dots, 150$. Choose a step size of $T = 0.5 \text{ s}$.
- Find the continuous velocity $v(t)$. Use it to find
 - the terminal velocity $v(\infty) = \lim_{t \rightarrow \infty} v(t)$.
 - $v(nT)$, $n = 0, 10, 20, \dots, 150$.
- Graph the approximate and continuous velocity over a period of time sufficient for the drum to reach its terminal velocity.
- If the drum impacts the ocean floor, 1 mile below the surface, at greater than 60 mph, it will break apart. Comment on the possibility of this happening.

- Equation 3.87 can be expressed in the form

$$\frac{dv}{dt} = f(v, u) = b_0 u - a_0 v \quad (3.88)$$

where $a_0 = \frac{cg}{W} = \frac{0.8(32.2)}{350} = 0.0736$, $b_0 = \frac{g}{W} (W - F_B) = \frac{32.2}{350} (350 - 275) = 6.9$ and the input u treated as a unit step function $u(t) = 1$, $t \geq 0$.

Evaluating the coefficient terms in Equation 3.86,

$$1 - \frac{a_0 T}{2} = 1 - \frac{0.0736(0.5)}{2} = 0.9816, \quad 1 + \frac{a_0 T}{2} = 1 + \frac{0.0736(0.5)}{2} = 1.0184$$

$$\frac{b_0 T}{2} = \frac{6.9(0.5)}{2} = 1.725$$

From Equation 3.86, the difference equation for approximating the dynamics of the sinking drum using trapezoidal integration is

$$v_A(n+1) = \frac{0.9816}{1.0184} v_A(n) + \frac{1.725}{1.0184} [1+1] \quad (3.89)$$

$$= 0.9639 v_A(n) + 3.3877, \quad n = 0, 1, 2, 3, \dots \quad (3.90)$$

b. [Table 3.5](#) shows the results for $v_A(n)$ at discrete times $n = 0, 10, 20, \dots, 150$.

The numerical values in [Table 3.5](#) were computed in the Matlab file “Ch3_Ex3_5.m”.

c. The continuous response is the analytical solution to Equation 3.87, namely

$$v(t) = \frac{W - F_B}{c} \left[1 - e^{-(cg/W)t} \right] \quad (3.91)$$

From Equation 3.91, the terminal velocity is

$$v(\infty) = \lim_{t \rightarrow \infty} v(t) = \frac{W - F_B}{c} = \frac{350 - 275}{0.8} = 93.75 \text{ ft/s} \quad (3.92)$$

The analytical solution $v(t)$ is evaluated at $t = 0, 5, 10, \dots, 75$ s and the values entered in [Table 3.5](#).

- d. Graphs of $v(t)$ and the approximate solution (every fifth point) are shown in [Figure 3.10](#).
- e. Since the terminal velocity of the drum exceeds 88 ft/s (60 mph), the possibility exists of it breaking when it reaches the ocean floor. It remains to be determined what the velocity of the drum is at the 1 mile depth of the ocean floor.

From [Table 3.5](#), it's apparent that trapezoidal integration with a step size of $T = 0.5$ s results in a very accurate approximation of the continuous response. However, in most simulation studies

TABLE 3.5
Discrete Response from Trapezoidal Integration
($T = 0.5$ s) and Continuous Response

n	$t_n = nT$	$v_A(n)$	$v(t_n)$
0	0	0.0	0.0
10	5	28.8667	28.8640
20	10	48.8450	48.8413
30	15	62.6718	62.6679
40	20	72.2411	72.2376
50	25	78.8640	78.8609
60	30	83.4475	83.4450
70	35	86.6198	86.6177
80	40	88.8153	88.8136
90	45	90.3347	90.3334
100	50	91.3863	91.3853
110	55	92.1141	92.1134
120	60	92.6178	92.6173
130	65	92.9664	92.9660
140	70	93.2077	93.2074
150	75	93.3747	93.3745

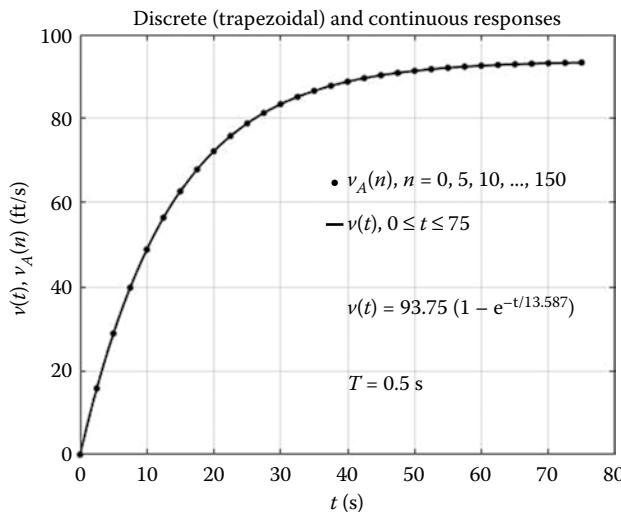


FIGURE 3.10 Discrete (trapezoidal, $T = 0.5$ s) response $v_A(n)$ and continuous response $v(t)$.

an exact solution of the governing differential equations is not available. In that case, what can we do to assure accurate simulation results?

An iterative method to determine an acceptable integration step size requires the simulation be executed with different values of T . For example, the step size can be continually reduced (say by one half, or a factor of 10) until changes in the output are deemed insignificant. The next to last step size is used in subsequent investigations. The method is not fool-proof and should be repeated if the simulation conditions change as a result of significant changes in the system inputs or initial conditions. We will have more to say about how to select the integration step size in [Chapters 6 and 8](#) when we investigate the subject of truncation errors and dynamic errors.

EXERCISES

3.10 Referring to [Figure 3.7](#),

- Find the equation of the linear approximation $f_1(t)$ through the end points $[nT, f(x_n, u_n)]$ and $[(n+1)T, f(x_{n+1}, u_{n+1})]$.
- Verify Equation 3.66 by integrating $f_1(t)$ from nT to $(n+1)T$.

- 3.11 The first-order system $dx/dt = \lambda x$ with initial condition $x(0) = x_0$ is to be simulated using trapezoidal integration with step size T . The truncation error after n steps is $\varepsilon_n = x_A(n) - x(nT)$, where $x(t)$, $t \geq 0$ is the exact solution and $x_A(n)$, $n = 0, 1, 2, \dots$ is the approximate (simulated) solution, i.e. $x_A(n) \approx x(nT)$, $n = 0, 1, 2, 3, \dots$. Suppose the truncation error after the first step is equal to a fraction of the initial condition, i.e.

$$\varepsilon_1 = x_A(1) - x(T) = \alpha x_0, \quad (0 < \alpha \ll 1)$$

λT satisfies the condition

$$e^{\lambda T} = \frac{a\lambda T + b}{\lambda T + c}$$

Express the constants a, b, c in terms of α and x_0 .

- 3.12 The population of a city $P(t)$ is modeled by the differential equation $dP/dt = kP$.
- Find the equation for updating $P_A(n)$, the approximate population at the end of year nT , using trapezoidal integration with step size T . Leave your answer in terms of k and T .

- b. Suppose $k = 0.01$ people/year per person, the initial population is 1 million people and the step size $T = 1$ yr. Find $P_A(1)$ and $P_A(2)$ to the nearest person.
- c. Find the general solution for $P_A(n)$ and use it to find $P_A(100)$.
- d. Compare the result from part (c) to the exact value $P(100)$.
- 3.13 The mass m in Figure E3.13 is subjected to a time varying damping force $f_d(t)$. The differential equation describing the motion is $m \frac{d}{dt} v(t) = f_d(t)$ where $v(t)$ is the velocity of the mass and $f_d(t) = -\frac{t}{1+t} v(t)$.
- Use trapezoidal integration with suitable step size T to approximate the velocity $v(t)$, $t \geq 0$. Note, $m = 1$ slug and the initial velocity $v(0) = 10$ ft/s.
 - Compare the simulated response $v_A(n)$, $n = 0, 1, 2, \dots$ in part (a) to the exact solution $v(t) = 10(1+t)e^{-t}$, $t \geq 0$.

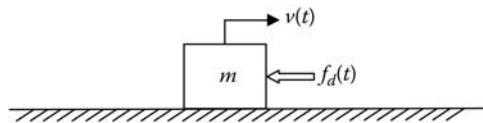


FIGURE E3.13

- 3.14 Find the largest step size T in Example 3.5 for which $|v(nT) - v_A(n)| < 0.05$ for the entire transient response.
- Start with $T = 0.025$ s and keep incrementing by 0.025 s until the condition is no longer satisfied.
- 3.15 Rework Example 3.5 using forward Euler integration. Choose the integration step size as $T = 0.5$ s, the same value used for trapezoidal integration. Prepare a similar table of results for the approximate and exact solutions.
- 3.16 The position of the sinking drum in Example 3.5 is related to its velocity by

$$y(t) = y(0) + \int_0^t v(t') dt'$$

Using trapezoidal integration and a step size $T = 2$ s, find the approximate solution $v_A(n)$ for 100 s and feed this discrete signal to another trapezoidal integrator to generate $y_A(n)$, the approximation to the actual position of the drum.

3.5 DISCRETE APPROXIMATION OF NONLINEAR FIRST-ORDER SYSTEMS

We now turn our attention to nonlinear first-order systems, that is, systems in which the state derivative $f(x,u)$ is a nonlinear function of the state x . The implicit numerical integrators produce implicit difference equations for updating the state.

Consider the first-order system governed by

$$\frac{dx}{dt} + N(x) = Ku \quad (3.93)$$

where $N(x)$ is a nonlinear function of the state x . The derivative function is

$$f(x,u) = \frac{dx}{dt} = Ku - N(x) \quad (3.94)$$

and the equation for updating the state using implicit Euler integration is

$$x_A(n+1) = x_A(n) + Tf[x_A(n+1), u(n+1)] \quad (3.95)$$

$$= x_A(n) + T \{ Ku(n+1) - N[x_A(n+1)] \} \quad (3.96)$$

Rearranging Equation 3.96 gives

$$x_A(n+1) + TN[x_A(n+1)] = x_A(n) + KT u(n+1) \quad (3.97)$$

a nonlinear equation that may prove difficult or impossible to solve for $x_A(n+1)$. To complicate matters further, multiple solutions may exist. The situation is illustrated in the following example.

EXAMPLE 3.6

The continuous model for the sinking drum in Example 3.5 governed its motion $v(t)$ as a function of time t . A relationship between its velocity $v = v(t)$ and depth $y = y(t)$ is obtained by solving the differential equation [Braun]

$$\frac{W}{g} v \frac{dv}{dy} + cv = W - F_B \quad (3.98)$$

- a. Find the difference equation to approximate the velocity of the drum as a function of depth using an implicit Euler integrator. Choose the integration step $T = 1$ ft.
- b. Find the approximate velocity $v_A(n)$ at depths of 0, 1000, 2000, 3000, 4000, 5000, and 6000 ft.
- c. Compare the results from part (b) to the true velocities $v(nT)$ at depths of 0, 1000, 2000, 3000, 4000, 5000, and 6000 ft.
- a. Dividing both sides of Equation 3.98 by Wv/g and introducing the input u gives

$$\frac{dv}{dy} + \frac{g}{W}(F_B - W) \frac{1}{v} = -\frac{gc}{W} u \quad (3.99)$$

where $u = u(y) = 1$, $y \geq 0$. Comparing Equations 3.93 and 3.99, it follows the nonlinear function $N(v)$ is

$$N(v) = \frac{g}{W}(F_B - W) \frac{1}{v} \quad (3.100)$$

and the constant K is expressible as

$$K = -\frac{gc}{W} \quad (3.101)$$

According to Equation 3.97, the implicit equation for $v_A(n+1)$ is

$$v_A(n+1) + T \frac{g}{W}(F_B - W) \frac{1}{v_A(n+1)} = v_A(n) - \frac{gc}{W} T(1) \quad (3.102)$$

Substituting the values $g = 32.2$, $c = 0.8$, $W = 350$, $F_B = 275$ and $T = 1$ ft yields

$$v_A(n+1) - 6.9 \frac{1}{v_A(n+1)} = v_A(n) - 0.0736 \quad (3.103)$$

b. Multiplying Equation 3.103 by $v_A(n+1)$ and collecting terms gives

$$v_A^2(n+1) + [0.0736 - v_A(n)]v_A(n+1) - 6.9 = 0 \quad (3.104)$$

which can be solved using the quadratic formula. The result is

$$v_A(n+1) = \frac{[v_A(n) - 0.0736] \pm \sqrt{[v_A(n) - 0.0736]^2 + 27.6}}{2} \quad (3.105)$$

Hence, in this case we are still able to update the new state $v_A(n+1)$ explicitly in terms of the previous state $v_A(n)$. The first two iterations are illustrated below.

$$\begin{aligned} n = 0: \quad v_A(1) &= \frac{[v_A(0) - 0.0736] \pm \sqrt{[v_A(0) - 0.0736]^2 + 27.6}}{2} \\ &= \frac{0 - 0.0736 + \sqrt{[0 - 0.0736]^2 + 27.6}}{2} \\ &= 2.5902 \\ \\ n = 1: \quad v_A(2) &= \frac{[v_A(1) - 0.0736] \pm \sqrt{[v_A(1) - 0.0736]^2 + 27.6}}{2} \\ &= \frac{2.5902 - 0.0736 + \sqrt{[2.5902 - 0.0736]^2 + 27.6}}{2} \\ &= 4.1709 \end{aligned}$$

Note, since the velocity is increasing, the negative root of Equation 3.105 was discarded. The M-file "Chap3_Ex3_6.m" generates the values of $v_A(n)$, $n = 1$ to 6000. The approximate velocities at depths $y_n = nT$, $n = 0, 1000, 2000, 3000, 4000, 5000, 6000$ are listed in [Table 3.6](#).

c. An exact solution to Equation 3.98, $v = v(y)$ is not possible. However, it is possible to obtain an exact solution for depth y as a function of the velocity v , namely

$$y = -\frac{W}{g} \left[\frac{v}{c} + \frac{W - F_B}{c^2} \ln \left(\frac{W - F_B - cv}{W - F_B} \right) \right] \quad (3.106)$$

We are interested in the depths corresponding to velocities up to the terminal velocity of 93.75 ft/s. Equation 3.106 can be evaluated for $0 \leq v \leq 93.75$ ft/s and the results plotted with depth y along the abscissa and velocity v along the ordinate axis as in [Figure 3.11](#).

From observation of [Figure 3.11](#), the true velocities at the required depths, 0, 1000, 2000, 3000, 4000, 5000, and 6000 ft agree with the approximate values in [Table 3.6](#).

The question in part (e) of Example 3.5 can now be answered. From [Figure 3.11](#), the velocity of the drum at a depth of 1 mile (5280 ft) does exceed 60 mph (88 ft/s).

TABLE 3.6
Implicit Euler Integration ($T = 1$ ft) of
Continuous Model in Equation 3.98

n	$y_n = nT$, (ft)	$v_A(n)$, (ft/s)
0	0	0
1000	1000	74.3629
2000	2000	85.9310
3000	3000	90.3467
4000	4000	92.2281
5000	5000	93.0618
6000	6000	93.4373

In the majority of cases, difference equations resulting from the use of implicit numerical integrators can only be solved by iterative schemes for finding the roots of nonlinear algebraic equations, as illustrated in the following example.

Consider an object falling in a viscous medium where the drag force is a nonlinear function of velocity as shown in Figure 3.12.

The continuous model describing the object's velocity $v(t)$ is given in Equation 3.107.

$$m \frac{dv}{dt} = W - f_D \quad (3.107)$$

Solving for the derivative function $f(v, W) = dv/dt$,

$$f(v, W) = \frac{1}{m}(W - f_D) \quad (3.108)$$

$$= \frac{1}{m}(W - cv^p) \quad (3.109)$$

Note the input is the constant weight W .

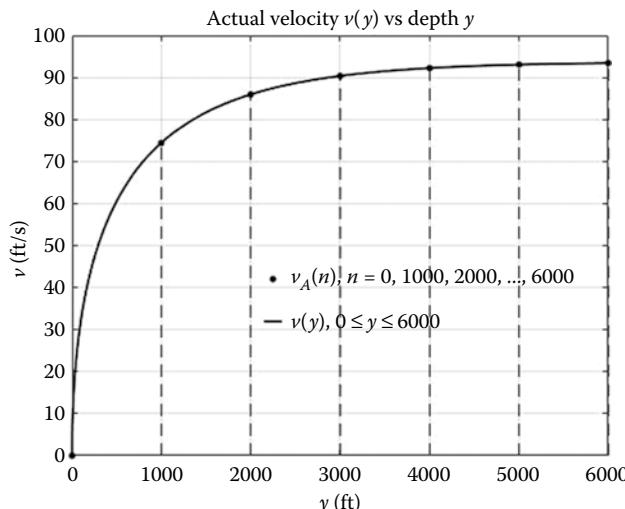


FIGURE 3.11 Plot of Equation 3.106 with v and y axes reversed.

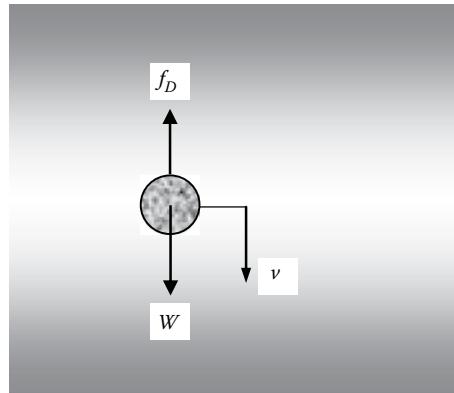


FIGURE 3.12 Object falling in a viscous medium with nonlinear drag force $f_D = cv^p$.

The difference equation based on implicit Euler integration is obtained as follows.

$$v_A(n+1) = v_A(n) + T f[v_A(n+1), W] \quad (3.110)$$

$$= v_A(n) + \frac{T}{m} \left\{ W - c [v_A(n+1)]^p \right\} \quad (3.111)$$

$$v_A(n+1) + \frac{cT}{m} [v_A(n+1)]^p = v_A(n) + \frac{WT}{m} \quad (3.112)$$

Unless p is numerically equal to 1 or 2, a root solving algorithm is required to solve Equation 3.112 for $v_A(n+1)$ once $v_A(n)$ has been determined. This process can dramatically increase the amount of computational overhead in comparison to what would be required for an explicit numerical integrator.

EXERCISES

- 3.17 In Example 3.6, find the largest step size T for which

$$\text{Max} |v(nT) - v_A(nT)| \leq 0.1$$

Start with $T = 0.025$ s and keep incrementing T by 0.025 s until the condition is no longer satisfied. Graph $\text{Max} |v(nT) - v_A(nT)|$ versus T and also prepare a plot of the exact and approximate velocities versus time.

- 3.18 Suppose $\alpha = 0.5$ and $p = 1.2$ in the example of the object falling in a viscous medium. The object is initially at rest.

- Find the approximate velocity of the object after 5 s. Use an explicit Euler integrator with an appropriate step size.

- Repeat part (a) using an implicit Euler integrator.

Hint: Use a root solving routine like the Single Point Iteration or Bisection Method to solve the implicit equation.

3.6 DISCRETE STATE EQUATIONS

Given the linear state equations

$$\underline{\dot{x}} = \underline{f}(\underline{x}, \underline{u}) = A\underline{x} + B\underline{u} \quad (3.113)$$

$$\underline{y} = \underline{g}(\underline{x}, \underline{u}) = C\underline{x} + D\underline{u} \quad (3.114)$$

A discrete model approximation of Equation 3.113 can be obtained in a straightforward manner. The approximation to the continuous state $\underline{x}(t)$ is $\underline{x}_A(nT)$ or simply $\underline{x}_A(n)$ for short. Difference equations for the discrete state $\underline{x}_A(n)$, using one of the previously discussed numerical integrators, are obtained in exactly the same way as before. For example, using explicit Euler integration,

$$\underline{x}_A(n+1) = \underline{x}_A(n) + T f[\underline{x}_A(n), \underline{u}(n)] \quad (3.115)$$

$$= \underline{x}_A(n) + T [A\underline{x}_A(n) + B\underline{u}(n)] \quad (3.116)$$

$$= (I + TA)\underline{x}_A(n) + TB\underline{u}(n) \quad (3.117)$$

The discrete output is determined from

$$\underline{y}_A(n) = C\underline{x}_A(n) + D\underline{u}(n) \quad (3.118)$$

An example involving the discrete state equations follows.

EXAMPLE 3.7

A circuit used in control systems is the RC lead-lag network shown in Figure 3.13. The differential equation relating the output $v_o(t)$ and input $v_i(t)$ is

$$R_1 C_1 R_2 C_2 \ddot{v}_o + (R_1 C_1 + R_1 C_2 + R_2 C_2) \dot{v}_o + v_o = R_1 C_1 R_2 C_2 \ddot{v}_i + (R_1 C_1 + R_2 C_2) \dot{v}_i + v_i \quad (3.119)$$

- Represent the circuit in state variable form.
- Find the discrete state equations for approximating the circuit dynamics based on the use of explicit Euler integration.
- The capacitor voltages are initially zero and the input is a step $v_i(t) = 1V$, $t \geq 0$. Approximate the step response using explicit Euler integration with step size $T = 0.001$ s. The circuit parameter values are $R_1 = 10,000 \Omega$, $R_2 = 5000 \Omega$, $C_1 = 7.5 \times 10^{-6} F$, $C_2 = 2.5 \times 10^{-6} F$.
- The circuit shown in Figure 3.13 is governed by the state equations

$$\frac{dv_{C_1}}{dt} = -\frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) v_{C_1} - \frac{1}{R_2 C_1} v_{C_2} + \frac{1}{R_2 C_1} v_i \quad (3.120)$$

$$\frac{dv_{C_2}}{dt} = -\frac{1}{R_2 C_2} v_{C_1} - \frac{1}{R_2 C_2} v_{C_2} + \frac{1}{R_2 C_2} v_i \quad (3.121)$$

Find the matrices A , B , C , and D in the state variable model with the states equal to the capacitor voltages.

- Repeat parts (b) and (c) using Equations 3.120 and 3.121. Compare the results in parts (c) and (e).

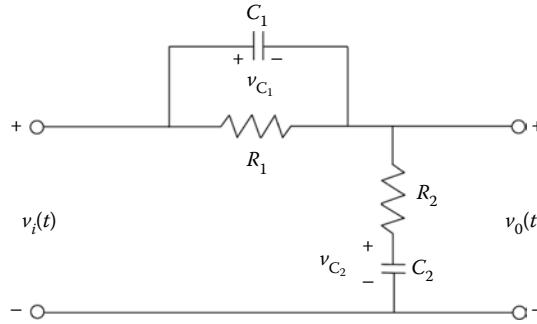


FIGURE 3.13 A lead-lag network.

- a. Dividing through by the lead coefficient term $R_1R_2C_1C_2$ and introducing new constants a_1, a_2, b_0, b_1, b_2 gives

$$\ddot{v}_o + a_1\dot{v}_o + a_0v_o = b_2\ddot{v}_i + b_1\dot{v}_i + b_0v_i \quad (3.122)$$

where

$$a_0 = \frac{1}{R_1C_1R_2C_2}, \quad a_1 = \frac{R_1C_1 + R_1C_2 + R_2C_2}{R_1C_1R_2C_2} \quad (3.123)$$

$$b_0 = \frac{1}{R_1C_1R_2C_2}, \quad b_1 = \frac{R_1C_1 + R_2C_2}{R_1C_1R_2C_2}, \quad b_2 = 1 \quad (3.124)$$

Constructing the simulation diagram for the system starts with the following two equations which are equivalent to Equation 2.73 (see [Chapter 2, Section 4](#)).

$$\ddot{z} + a_1\dot{z} + a_0z = v_i \quad (3.125)$$

$$v_o = b_0z + b_1\dot{z} + b_2\ddot{z} \quad (3.126)$$

Solving for \ddot{z} in Equation 3.125 and substituting the result in Equation 3.126 yields

$$v_o = b_0z + b_1\dot{z} + b_2(v_i - a_0z - a_1\dot{z}) \quad (3.127)$$

$$= (b_0 - a_0b_2)z + (b_1 - a_1b_2)\dot{z} + b_2v_i \quad (3.128)$$

The simulation diagram follows directly from Equations 3.125 and 3.128. It is presented in [Figure 3.14](#).

Choosing the outputs of the integrators in [Figure 3.14](#) as the states results in

$$\dot{x}_1 = x_2 \quad (3.129)$$

$$\dot{x}_2 = -a_0x_1 - a_1x_2 + v_i \quad (3.130)$$

$$v_o = (b_0 - a_0b_2)x_1 + (b_1 - a_1b_2)x_2 + b_2v_i \quad (3.131)$$

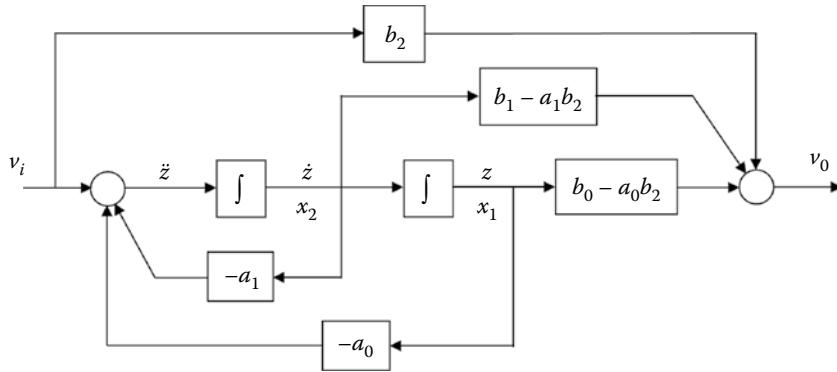


FIGURE 3.14 Simulation diagram for RC lead-lag network based on Equation 3.119.

From Equations 3.129–3.131, the matrices A , B , C , and D in the linear state equations $\dot{\underline{x}} = A\underline{x} + B\underline{u}$, $\underline{y} = C\underline{x} + D\underline{u}$ are

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [b_0 - a_0 b_2 \quad b_1 - a_1 b_2], \quad D = [b_2] \quad (3.132)$$

In term of the electrical parameters

$$A = \begin{bmatrix} 0 & 1 \\ \frac{-1}{R_1 C_1 R_2 C_2} & -\left(\frac{R_1 C_1 + R_1 C_2 + R_2 C_2}{R_1 C_1 R_2 C_2}\right) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & -1 \\ R_2 C_1 \end{bmatrix}, \quad D = [1] \quad (3.133)$$

b. From Equations 3.117 and 3.118, the discrete state equations are

$$\underline{x}_A(n+1) = \begin{bmatrix} 1 & T \\ \frac{-T}{R_1 C_1 R_2 C_2} & 1 - T \left(\frac{R_1 C_1 + R_1 C_2 + R_2 C_2}{R_1 C_1 R_2 C_2} \right) \end{bmatrix} \underline{x}_A(n) + \begin{bmatrix} 0 \\ T \end{bmatrix} v_i(n) \quad (3.134)$$

$$y_{A,1}(n) = v_o(n) = \begin{bmatrix} 0 & -1 \\ R_2 C_1 \end{bmatrix} \underline{x}_A(n) + v_i(n) \quad (3.135)$$

c. Equation 3.134 is solved recursively in “Ch3_Ex3_7.m” for the state $\underline{x}_A(n)$, which is used in Equation 3.135 to find the discrete step response $v_o(n)$, $n = 0, 1, 2, \dots$. The first 25 discrete points and every 10th point after that until steady-state are plotted in the top window in [Figure 3.15](#).

d. Writing Equations 3.111 and 3.112 in state variable form

$$\begin{bmatrix} \dot{v}_{C_1} \\ \dot{v}_{C_2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{C_1} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) & -\frac{1}{R_2 C_1} \\ -\frac{1}{R_2 C_2} & -\frac{1}{R_2 C_2} \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_2 C_1} \\ \frac{1}{R_2 C_2} \end{bmatrix} v_i \quad (3.136)$$

From the circuit, the output equation is

$$v_o = v_i - v_{C_1} \quad (3.137)$$

$$= [-1 \quad 0] \begin{bmatrix} v_{C_1} \\ v_{C_2} \end{bmatrix} + [1] v_i \quad (3.138)$$

The matrices A , B , C , and D follow directly from Equations 3.136 and 3.138.

- e. The new state equations are discretized based on the use of explicit Euler integration and solved recursively in "Ch3_Ex7_3.m". The result is shown in the bottom window of [Figure 3.15](#). The two step responses are identical.

The second choice of the state variables, namely the capacitor voltages is more intuitive than the state definition based on the simulation diagram in [Figure 3.14](#). The output vector could be modified to include additional outputs $y_2 = v_{C_1}$ and $y_3 = v_{C_2}$ making $\underline{y} = [v_o \ v_{C_1} \ v_{C_2}]^T$ to allow visualizing the capacitor voltages.

A recursive solution to Equation 3.134 requires the initial discrete state vector $\underline{x}_A(0) = [x_{1,A}(0) \ x_{2,A}(0)]^T = [x_1(0) \ x_2(0)]^T$. Since the states are not physical quantities, their initial values must be calculated from knowledge of the initial capacitor voltages $v_{C_1}(0)$ and $v_{C_2}(0)$. In this example, $x_{1,A}(0) = x_1(0) = 0$ and $x_{2,A}(0) = x_2(0) = 0$ because $v_{C_1}(0)$ and $v_{C_2}(0)$ are both zero.

If either of the two implicit numerical integrators were used instead of the explicit Euler integrator, Equation 3.115 is replaced with one of the following two equations:

$$\text{Implicit Euler: } \underline{x}_A(n+1) = \underline{x}_A(n) + T \underline{f}[\underline{x}_A(n), \underline{u}(n+1)] \quad (3.139)$$

$$\text{Trapezoidal: } \underline{x}_A(n+1) = \underline{x}_A(n) + \frac{T}{2} \left\{ \underline{f}[\underline{x}_A(n), \underline{u}(n)] + \underline{f}[\underline{x}_A(n+1), \underline{u}(n+1)] \right\} \quad (3.140)$$

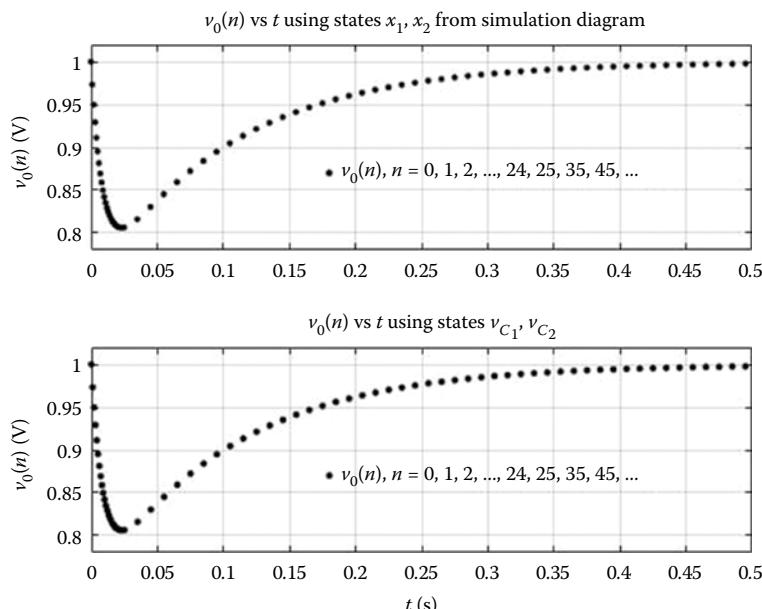


FIGURE 3.15 Discrete step response of circuit using different state definitions.

If the continuous system is linear,

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}) = A\underline{x} + B\underline{u} \quad (3.141)$$

$$\underline{y} = \underline{g}(\underline{x}, \underline{u}) = C\underline{x} + D\underline{u} \quad (3.142)$$

Equations 3.139 and 3.140 can be solved explicitly for $\underline{x}_A(n+1)$ in terms of $\underline{x}_A(n)$, $\underline{u}(n)$ and $\underline{u}(n+1)$. For implicit Euler integration,

$$\underline{x}_A(n+1) = \underline{x}_A(n) + T[A\underline{x}_A(n+1) + B\underline{u}(n+1)] \quad (3.143)$$

Solving for $\underline{x}_A(n+1)$ gives

$$\underline{x}_A(n+1) = (I - TA)^{-1}[\underline{x}_A(n) + TB\underline{u}(n+1)] \quad (3.144)$$

Using trapezoidal integration to update the state,

$$\underline{x}_A(n+1) = \underline{x}_A(n) + \frac{T}{2}[A\underline{x}_A(n) + B\underline{u}(n) + A\underline{x}_A(n+1) + B\underline{u}(n+1)] \quad (3.145)$$

Solving Equation 3.145 for $\underline{x}_A(n+1)$ gives

$$\underline{x}_A(n+1) = \left(I - \frac{1}{2}TA\right)^{-1}\left(I + \frac{1}{2}TA\right)\underline{x}_A(n) + \frac{1}{2}\left(I - \frac{1}{2}TA\right)^{-1}TB[\underline{u}(n) + \underline{u}(n+1)] \quad (3.146)$$

In Equations 3.144 and 3.146, the state is updated recursively without the need to solve an implicit equation for $\underline{x}_A(n+1)$; however, the computations are more extensive than with explicit Euler integration because of the requirement to invert the matrix $(I - TA)$ in Equation 3.144 and $\left(I - \frac{1}{2}TA\right)$ in Equation 3.146.

The difference equation for approximating the state response of the linear system using explicit Euler integration assumed the form

$$\underline{x}_A(n+1) = G\underline{x}_A(n) + H\underline{u}(n) \quad (3.147)$$

where

$$G = (I + TA), \quad H = TB \quad (3.148)$$

For a stable discrete system, the steady-state response to a constant input $\underline{u}(n) = \underline{u}^0$, $n = 0, 1, 2, \dots$ is found by setting $\lim_{n \rightarrow \infty} \underline{x}_A(n+1) = \lim_{n \rightarrow \infty} \underline{x}_A(n) = \underline{x}_A(\infty)$. Doing this in Equation 3.147,

$$\underline{x}_A(\infty) = G\underline{x}_A(\infty) + H\underline{u}^0 \quad (3.149)$$

$$(I - G)\underline{x}_A(\infty) = H\underline{u}^0 \quad (3.150)$$

$$\underline{x}_A(\infty) = (I - G)^{-1}H\underline{u}^0 \quad (3.151)$$

Substituting for G and H from Equation 3.148,

$$\underline{x}_A(\infty) = [I - (I + TA)]TB\underline{u}^0 \quad (3.152)$$

$$= -A^{-1}B\underline{u}^0 \quad (3.153)$$

A similar approach applies to the implicit integrators with difference equations given in Equations 3.144 and 3.146. Starting with

$$\underline{x}_A(n+1) = G\underline{x}_A(n) + H_0\underline{u}(n) + H_1\underline{u}(n+1) \quad (3.154)$$

$$\text{Implicit Euler: } G = (I - TA)^{-1}, \quad H_0 = 0, \quad H_1 = (I - TA)^{-1}TB \quad (3.155)$$

$$\text{Trapezoidal: } G = \left(I - \frac{1}{2}TA\right)^{-1}\left(I + \frac{1}{2}TA\right), \quad H_0 = H_1 = \frac{1}{2}\left(I - \frac{1}{2}TA\right)^{-1}TB \quad (3.156)$$

The steady-state responses are obtained from

$$\underline{x}_A(\infty) = G\underline{x}_A(\infty) + H_0\underline{u}^0 + H_1\underline{u}^0 \quad (3.157)$$

$$\underline{x}_A(\infty) - G\underline{x}_A(\infty) = (H_0 + H_1)\underline{u}^0 \quad (3.158)$$

$$\underline{x}_A(\infty) = (I - G)^{-1}(H_0 + H_1)\underline{u}^0 \quad (3.159)$$

Using G , H_0 and H_1 in Equation 3.155 for implicit Euler integration,

$$\underline{x}_A(\infty) = [I - (I - TA)^{-1}]^{-1}(I - TA)^{-1}TB\underline{u}^0 \quad (3.160)$$

Using $A^{-1}B^{-1} = (BA)^{-1}$ from matrix algebra, Equation 3.160 becomes

$$\underline{x}_A(\infty) = [(I - TA) - I]^{-1}TB\underline{u}^0 \quad (3.161)$$

$$= (-TA)^{-1}TB\underline{u}^0 \quad (3.162)$$

$$= -A^{-1}B\underline{u}^0 \quad (3.163)$$

which is the identical result for explicit Euler integration in Equation 3.153. It should come as no surprise that the same result occurs when combining Equations 3.156 and 3.159 for trapezoidal integration.

Indeed, the difference equation resulting from the use of any numerical integrator will produce the same value for $\underline{x}_A(\infty)$, provided a finite $\underline{x}_A(\infty)$ exists. Note the predicted steady-state result for $\underline{x}_A(\infty)$ is independent of the step size T . Of course the accuracy of the dynamic (transient) component of the discrete response is strongly dependent on the step size T .

It's instructive to investigate the continuous system $\dot{\underline{x}} = A\underline{x} + B\underline{u}$ at steady-state in response to a constant input $\underline{u} = \underline{u}^0$, $t \geq 0$. The steady-state $\underline{x}(\infty) = \lim_{t \rightarrow \infty} \underline{x}(t)$ is obtained by setting the derivative $\dot{\underline{x}}$ equal to the zero vector $\underline{0}$. That is,

$$\underline{0} = A\underline{x}(\infty) + B\underline{u}^0 \quad (3.164)$$

$$\underline{x}(\infty) = -A^{-1}B\underline{u}^0 \quad (3.165)$$

leading to the conclusion that $\underline{x}(\infty) = \lim_{t \rightarrow \infty} \underline{x}(t) = \underline{x}_A(\infty) = \lim_{n \rightarrow \infty} \underline{x}_A(n)$.

EXAMPLE 3.8

Consider the system of two interacting tanks presented in [Chapter 2](#), Section 2.6. (See [Figure 2.28](#))
The state equations are

$$\begin{bmatrix} \frac{dH_1}{dt} \\ \frac{dH_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_{12}} & \frac{1}{A_1 R_{12}} \\ \frac{1}{A_2 R_{12}} & -\frac{(R_1 + R_2)}{A_1 R_{12} R_2} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A_1} & 0 \\ 0 & \frac{1}{A_2} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad (3.166)$$

The numerical values of the physical parameters are:

$$R_{12} = 2 \text{ ft per cu ft/min}, R_2 = 0.5 \text{ ft per cu ft/min}, A_1 = 15 \text{ ft}^2, A_2 = 10 \text{ ft}^2,$$

For input flows of $F_1(t) = 5 \text{ ft}^3/\text{min}$ and $F_2(t) = 2 \text{ ft}^3/\text{min}$, the discrete responses for both tank levels, based on explicit Euler and implicit Euler integration with step size $T = 0.25 \text{ min}$, are shown in [Figures 3.16](#) and [3.17](#), respectively.

Methods for finding the continuous responses for $H_1(t)$ and $H_2(t)$ are deferred until [Chapter 4](#).
The continuous tank level responses are:

$$H_1(t) = 13.5 + 0.0703e^{-0.2574t} - 13.5703e^{-0.0259t}, \quad t \geq 0 \quad (3.167)$$

$$H_2(t) = 3.5 - 0.4723e^{-0.2574t} - 3.0277e^{-0.0259t}, \quad t \geq 0 \quad (3.168)$$

The discrete responses are in close agreement with the continuous responses.
The predicted steady-state levels are

$$\underline{x}_A(\infty) = -A^{-1}B\underline{u}^0 = -\begin{bmatrix} -0.0333 & 0.0333 \\ 0.05 & -0.25 \end{bmatrix}^{-1} \begin{bmatrix} 0.0667 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 13.5 \\ 3.5 \end{bmatrix}$$

in agreement with the values obtained by letting $t \rightarrow \infty$ in Equations 3.167 and 3.168.

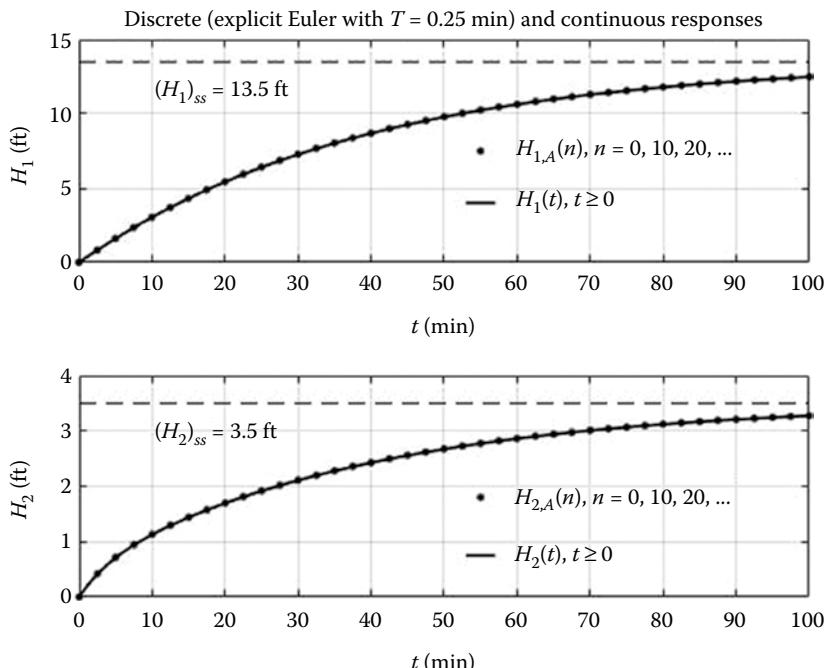


FIGURE 3.16 Discrete (explicit Euler, $T = 0.25 \text{ min}$) and continuous tank level responses.

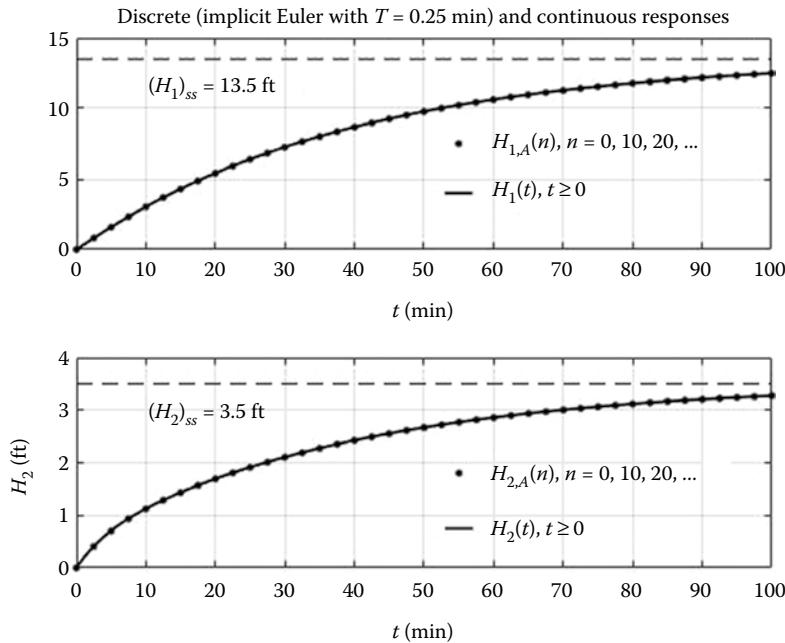


FIGURE 3.17 Discrete (implicit Euler, $T = 0.25$ min) and continuous tank level responses.

Figure 3.18 demonstrates the effect of step size T on accuracy of the discrete response. The step size is increased to $T = 7.5$ min producing a noticeable difference between the discrete and continuous responses when using explicit Euler integration. In fact, $H_{2,A}(n)$ exhibits an oscillatory response. However, despite the oscillatory response, both $H_{1,A}(n)$ and $H_{2,A}(n)$ are stable and approach the correct steady-state values.

EXERCISES

- 3.19 Verify the solution for $\underline{x}_A(n+1)$ in Equation 3.146 which gives the updated state in the approximate solution of $\dot{\underline{x}} = A\underline{x} + B\underline{u}$ by trapezoidal integration.
- 3.20 Find the discrete state equations for the circuit in Example 3.7 using
 - a. Implicit Euler integration
 - b. Trapezoidal integration
- 3.21 In the lead-lag circuit of Example 3.7, the outputs are $y_1 = v_0$, $y_2 = v_{C_1}$, $y_3 = v_{C_2}$.
 - a. Find the difference equations based on trapezoidal integration with step size T for approximating the continuous system outputs to input $v_i(t)$
 - b. The capacitor voltages are both initially zero and the input is a step voltage of 12 volts applied at $t = 0$. Solve the difference equations recursively and plot the discrete-time outputs in the output vector $\underline{y}_A(n) = [y_{1,A}(n) \ y_{2,A}(n) \ y_{3,A}(n)]^T$.
 - c. The initial capacitor voltages are $v_{C_1}(0) = 1$ V, $v_{C_2}(0) = 0$ V and the input $v_i(t) = 0$ V, $t \geq 0$. Solve the difference equations recursively and plot the discrete outputs in the output vector $\underline{y}_A(n) = [y_{1,A}(n) \ y_{2,A}(n) \ y_{3,A}(n)]^T$.
- 3.22 For the circuit in Example 3.7 described by Equations 3.120 and 3.121.
 - a. Use the technique presented in Chapter 2, Section 3 for converting two first-order differential equations into a single second order differential equation to eliminate $v_{C_2}(t)$ from the two equations and obtain

$$\ddot{v}_{C_1} + \alpha_1 \dot{v}_{C_1} + \alpha_0 v_{C_1} = \beta_2 \ddot{v}_i + \beta_1 \dot{v}_i + \beta_0 v_i$$

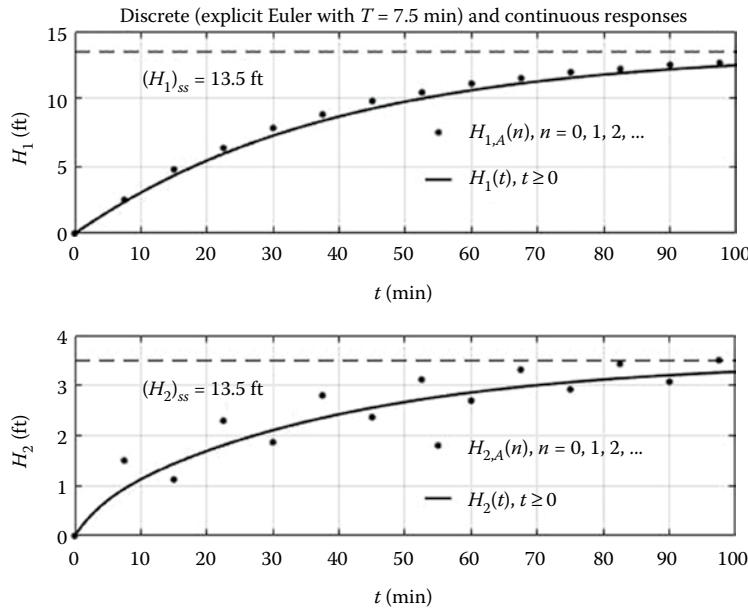


FIGURE 3.18 Discrete (explicit Euler, $T = 7.5$ min) and continuous tank level responses.

- Express the coefficients $\alpha_1, \alpha_0, \beta_2, \beta_1, \beta_0$ in terms of the electrical parameters R_1, R_2, C_1, C_2 .
- The circuit output is $v_0(t)$. Find the matrices A, B, C, D in the continuous state equation model. Express your answers in terms of the circuit parameters R_1, R_2, C_1 , and C_2 .
 - Find the matrices G and H in the discrete state equations resulting from the use of explicit Euler integration to approximate the continuous response of the circuit.
 - The input $v_i(t) = 1$ V, $t \geq 0$. Find and plot the discrete response $v_0(n)$, $n = 0, 1, 2, \dots$ based on explicit Euler integration with step size $T = 0.001$ s and compare your answer to the results shown in [Figure 3.15](#).
- 3.23 The dynamic interaction of rabbit and fox populations in a forest is under investigation. The predator-prey ecosystem is illustrated in [Figure E3.23](#).

$R(t)$ = Population of rabbits after “ t ” wks

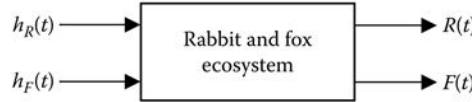


FIGURE E3.23

$F(t)$ = Population of foxes after “ t ” wks

$h_R(t)$ = Rate of rabbit hunting, (rabbits/wk)

$h_F(t)$ = Rate of fox hunting, (fox/wk)

The mathematical model consists of the following coupled differential equations:

$$\frac{dR}{dt} = aR - bF - h_R$$

$$\frac{dF}{dt} = -cF + dR - h_F$$

a, b = constant parameters defining the growth rate of rabbits

c, d = constant parameters defining the growth rate of foxes

- a. Find the equilibrium point (R_e, F_e) when $h_R(t) = \bar{h}_R$, $t \geq 0$ and $h_F(t) = \bar{h}_F$, $t \geq 0$. Express your answers for R_e and F_e in terms of the system parameters a, b, c, d and constant hunting rates \bar{h}_R, \bar{h}_F .
- b. Baseline values of the system parameters are given below.

$$a = 0.04 \frac{\text{rabbits/wk}}{\text{rabbit}}, b = 0.2 \frac{\text{rabbits/wk}}{\text{fox}}, c = 0.1 \frac{\text{foxes/wk}}{\text{fox}}, d = 0.0075 \frac{\text{foxes/wk}}{\text{rabbit}}$$

Foxes are endangered and hunting foxes is forbidden. Rabbits are hunted at a constant rate and after a long period of time the fox population stabilizes at 750. Find the constant rate of rabbit hunting. Find the rabbit population at the same time.

- c. Let the state \underline{x} be defined as $\underline{x}(t) = \begin{bmatrix} R(t) \\ F(t) \end{bmatrix}$ and the input vector \underline{u} be defined as $\underline{u}(t) = \begin{bmatrix} h_R(t) \\ h_F(t) \end{bmatrix}$. Find the matrices A and B in the state equation $\dot{\underline{x}} = A\underline{x} + B\underline{u}$.
- d. Suppose neither rabbits or foxes are hunted. Using explicit Euler integration with step size $T = 1$ week, find and plot the discrete responses $R(n)$ and $F(n)$ until steady-state is reached. The initial populations of rabbits and foxes are $R(0) = 10,000$ and $F(0) = 1000$.

- 3.24 A mass is suspended from a stationary support by a spring as shown in [Figure E3.24](#). The mass is displaced from its equilibrium position 1 ft and released with zero velocity. The continuous model of the system is $m\ddot{x} + kx = 0$.

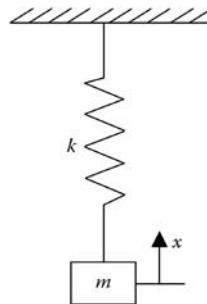


FIGURE E3.24

- a. Find the matrix A in the state equations $\dot{\underline{x}} = A\underline{x}$ for the continuous model.
- b. Find the matrix G in the discrete state equations $\underline{x}_A(n+1) = G\underline{x}_A(n)$ resulting from the use of explicit Euler integration to approximate the response of the continuous system.
- c. The system parameters are $k = 4$ lb/ft and $m = 1$ slug. Fill in [Table E3.24](#).

TABLE E3.24

n	$x_A(n)$		n	$x_A(n)$	
	$(T = 0.05 \text{ s})$	$x(nT)$		$(T = 0.01 \text{ s})$	$x(nT)$
0		0			0
1					5
2					10
3					15
4					20
5					25
6					30
7					35
8					40
9					45
10					50

3.7 IMPROVEMENTS TO EULER INTEGRATION

Euler integration is popular in large measure due to its simplicity. A graphical interpretation of either explicit or implicit Euler integration is straightforward. A discussion of error characteristics for Euler integrators is deferred until a later chapter. However, it's apparent that serious errors can propagate as the discrete-time variable “ n ” increases with Euler integration as a result of the underlying assumption that the state derivative remains constant for an entire integration step. For systems in which one or more of the state variables experience frequent fluctuations (relative to the integration step size) this assumption is unjustified.

The inherent weakness of Euler integration can be overcome in ways other than simply reducing the integration step size, which may not always be practical. In addition to trapezoidal integration, another method for obtaining more accurate state updates than Euler integration is illustrated in [Figure 3.19](#).

3.7.1 IMPROVED EULER INTEGRATION

With explicit Euler integration, advancing the state $x_A(n)$ is equivalent to projecting line segment L_1 , whose slope is $f[x_A(n), u(n)]$ until it reaches the end of the interval at $(n + 1)T$. The updated state is shown as $\hat{x}_A(n + 1)$. From there, another forward Euler integration step would proceed along the line segment L_2 , whose slope is $f[\hat{x}_A(n + 1), u(n + 1)]$.

Recognizing that L_1 may not be the most judicious direction to move along for obtaining $x_A(n + 1)$, the approximation of the continuous state $x[(n + 1)T]$, the question to be asked is “Is there a better choice for determining the path from $x_A(n)$ to $x_A(n + 1)$?” The line segment L starting from $x_A(n)$ with slope equal to the average of the slopes of L_1 and L_2 appears to be a more prudent choice.

Since Euler integration is predicated on the assumption that the derivative function $f(x, u)$ is constant, it makes sense to base the constant on evaluations of $f(x, u)$ at more than one point. In summary, a new method for computing $x_A(n + 1)$ consists of the following:

1. Prediction of the new state using forward Euler integration, that is, moving from $x_A(n)$ to $\hat{x}_A(n + 1)$ along the line segment with slope is L_1 .

$$\hat{x}_A(n + 1) = x_A(n) + Tf[x_A(n), u(n)] \quad (3.169)$$

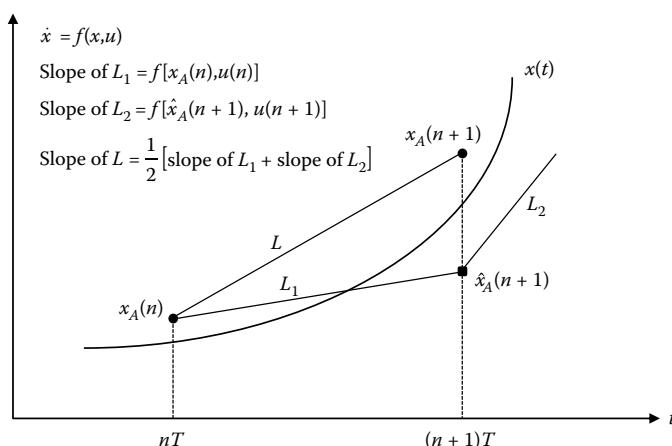


FIGURE 3.19 Illustration of improved Euler method.

2. Computing the derivative function $f[\hat{x}_A(n+1), u(n+1)]$ at $\hat{x}_A(n+1)$, that is, the slope of line segment L_2 .
3. Improving the predicted value $\hat{x}_A(n+1)$, that is, moving from $x_A(n)$ along a line segment whose slope is the average of the slopes of line segments L_1 and L_2 to the new updated state $x_A(n+1)$.

$$x_A(n+1) = x_A(n) + \frac{T}{2} \{ f[x_A(n), u(n)] + f[\hat{x}_A(n+1), u(n+1)] \} \quad (3.170)$$

The numerical integrator based on Equations 3.169 and 3.170 is called improved Euler integration, also known as Heun's Method.

When the state is a vector and the system model is linear, that is,

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}) = A\underline{x} + B\underline{u} \quad (3.171)$$

the predicted state using forward Euler integration was shown to be

$$\hat{x}_A(n+1) = (I + TA)\underline{x}_A(n) + TB\underline{u}(n) \quad (3.172)$$

The improved state estimate is computed from

$$\underline{x}_A(n+1) = \underline{x}_A(n) + \frac{T}{2} \{ f[\underline{x}_A(n), \underline{u}(n)] + f[\hat{x}_A(n+1), \underline{u}(n+1)] \} \quad (3.173)$$

Substituting Equation 3.172 into Equation 3.173 results in

$$\underline{x}_A(n+1) = \left[I + TA + \frac{1}{2}(TA)^2 \right] \underline{x}_A(n) + \frac{1}{2} T(I + TA)B\underline{u}(n) + \frac{1}{2} TB\underline{u}(n+1) \quad (3.174)$$

Note the additional term $\frac{1}{2}(TA)^2$ in brackets in Equation 3.174 compared with explicit Euler integration.

The following example demonstrates the improved accuracy with improved Euler integration compared to ordinary Euler integration (explicit or implicit).

EXAMPLE 3.9

Consider the autonomous, undamped second-order system

$$\ddot{x} + \omega^2 x = 0 \quad (3.175)$$

Choosing state variables $x_1(t) = x(t)$ and $x_2(t) = \dot{x}(t)$ leads to the state equations

$$\dot{x}_1 = f_1(x_1, x_2) = x_2 \quad (3.176)$$

$$\dot{x}_2 = f_2(x_1, x_2) = -\omega^2 x_1 \quad (3.177)$$

The initial conditions are $x_1(0) = x(0) = x_0$, $x_2(0) = \dot{x}(0) = \dot{x}_0$.

- Find the system matrix A .
- Find the general solution of the discrete state equations using explicit and improved Euler integrators.
- Find the transient response using explicit and Improved Euler integrators when $\omega = 1 \text{ rad/s}$, $x_0 = 1 \text{ ft}$, $\dot{x}_0 = 0 \text{ ft/s}$, and $T = 0.25 \text{ s}$. Plot the results.
- Find the exact solution for the transient response of the continuous system and compare it to the approximate solutions in part (c).

- From Equations 3.176 and 3.177, the system matrix is

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \quad (3.178)$$

- It is left as an exercise problem to show the general solutions for the discrete states for each integrator are:

$$\text{Explicit Euler: } \underline{x}_A(n) = (I + TA)^n \underline{x}(0) \quad (3.179)$$

$$\text{Improved Euler: } \underline{x}_A(n) = \left[I + TA + \frac{1}{2}(TA)^2 \right]^n \underline{x}(0) \quad (3.180)$$

Substituting Equation 3.178 into Equations 3.179 and 3.180 gives

$$\text{Explicit Euler: } \underline{x}_A(n) = \begin{bmatrix} 1 & T \\ -\omega^2 T & 1 \end{bmatrix}^n \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} \quad (3.181)$$

$$\text{Improved Euler: } \underline{x}_A(n) = \begin{bmatrix} 1 - \frac{1}{2}(\omega T)^2 & T \\ -\omega^2 T & 1 - \frac{1}{2}(\omega T)^2 \end{bmatrix}^n \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} \quad (3.182)$$

- The transient responses of the discrete states $x_{1,A}(n)$ and $x_{2,A}(n)$ when $\omega = 1 \text{ rad/s}$, $x_0 = 1 \text{ ft}$, $\dot{x}_0 = 0 \text{ ft/s}$, and $T = 0.25 \text{ s}$ are plotted in Figures 3.20 and 3.21 for the explicit and Improved Euler integrators.
- The exact solution for the continuous states of the undamped second-order system is

$$x_1(t) = x_0 \cos \omega t \quad (3.183)$$

$$x_2(t) = -\omega x_0 \sin \omega t \quad (3.184)$$

and plotted in Figures 3.20 and 3.21.

Note the considerable improvement in accuracy obtained with the Improved Euler integrator. The discrete state $\underline{x}_A(n) = [x_{1,A}(n) \ x_{2,A}(n)]^T$ based on explicit Euler integration is a poor approximation to the continuous state, to say the least. This is not surprising in light of the fact that the state derivatives \dot{x}_1 and \dot{x}_2 vary significantly over the interval T , in violation of the basic assumption underlying explicit Euler integration.

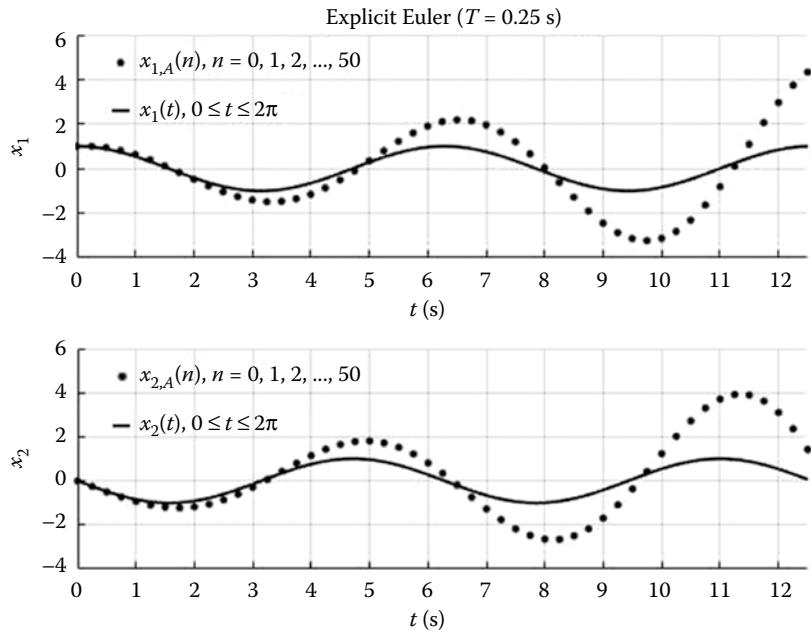


FIGURE 3.20 Continuous and discrete (explicit Euler, $T = 0.25$ s) responses.

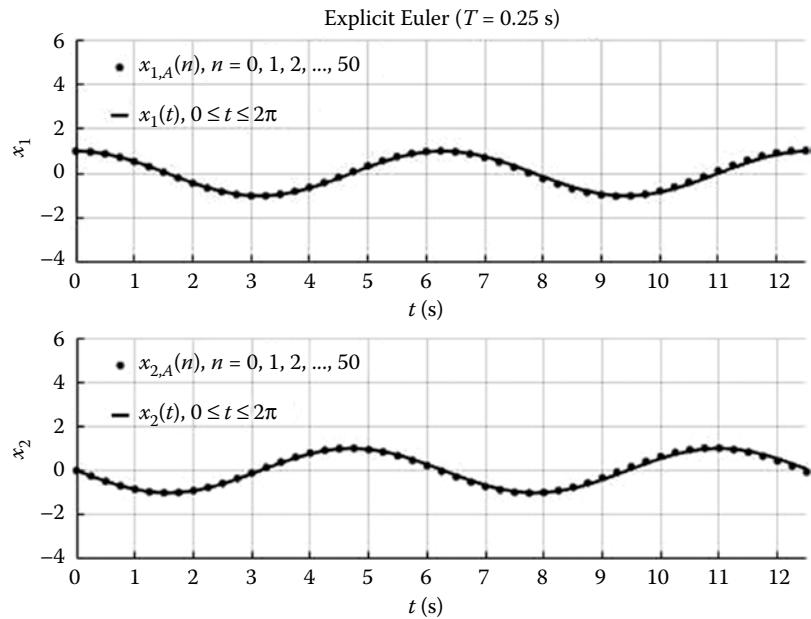


FIGURE 3.21 Continuous and discrete (implicit Euler, $T = 0.25$ s) responses.

In Chapter 8, we will learn that explicit Euler integration of an undamped second-order system is never stable and should not be used. However, lightly damped second-order systems, which have high natural frequencies, require smaller integration steps for accurate results. Dynamic accuracy is discussed in great detail in Chapter 8. It will be shown that the controlling parameter for dynamic accuracy is ωT , the product of natural frequency and integration time step.

In general, explicit (also known as forward) Euler integration does not result in the “best” direction for advancing the state from $x_A(n)$ to $x_A(n + 1)$. As the name suggests, improved Euler integration represents an improvement although it comes with a penalty of requiring twice as many state derivative function evaluations compared with explicit Euler integration for the identical step size.

3.7.2 MODIFIED EULER INTEGRATION

Another method for finding a better direction (compared with explicit Euler integration) to proceed from the current state is portrayed in Figure 3.22. It is called the midpoint or modified Euler method because the line segment L , which determines the new approximate state is based on a state derivative calculation at the midpoint of the interval.

Starting from the current discrete state $\underline{x}_A(n)$, a forward Euler step is taken along line segment L_1 ending up at the point $[(n + 1/2)T, \underline{x}_A(n + 1/2)]$. A new direction is calculated, namely, $f[\underline{x}_A(n + 1/2), \underline{u}(n + 1/2)]$ which represents the slope of line L_2 . Finally, the updated state $\underline{x}_A(n + 1)$ is obtained by starting from the current state $\underline{x}_A(n)$ and moving in the direction of line segment L , which is parallel to line segment L_2 , until the end of the interval.

A discrete state equation can be obtained for the approximate solution of $\dot{\underline{x}} = f(\underline{x}, \underline{u}) = A\underline{x} + B\underline{u}$, based on the use of modified Euler integration, in the same way it was obtained with Improved Euler integration. First the state $\underline{x}_A(n + 1/2)$ is calculated from

$$\underline{x}_A\left(n + \frac{1}{2}\right) = \underline{x}_A(n) + \frac{T}{2}f[\underline{x}_A(n), \underline{u}(n)] \quad (3.185)$$

The updated state $\underline{x}_A(n + 1)$ is based on the derivative function $f(\underline{x}, \underline{u})$ evaluated at the point $[(n + 1/2)T, \underline{x}_A(n + 1/2)]$ according to

$$\underline{x}_A(n + 1) = \underline{x}_A(n) + Tf\left[\underline{x}_A\left(n + \frac{1}{2}\right), \underline{u}\left(n + \frac{1}{2}\right)\right] \quad (3.186)$$

From Equations 3.185, 3.186, and $\dot{\underline{x}} = A\underline{x} + B\underline{u}$, the equation for updating the state is

$$\underline{x}_A(n + 1) = \left[I + (TA) + \frac{1}{2}(TA)^2 \right] \underline{x}_A(n) + \frac{1}{2}T^2 AB\underline{u}(n) + TB\underline{u}\left(n + \frac{1}{2}\right) \quad (3.187)$$

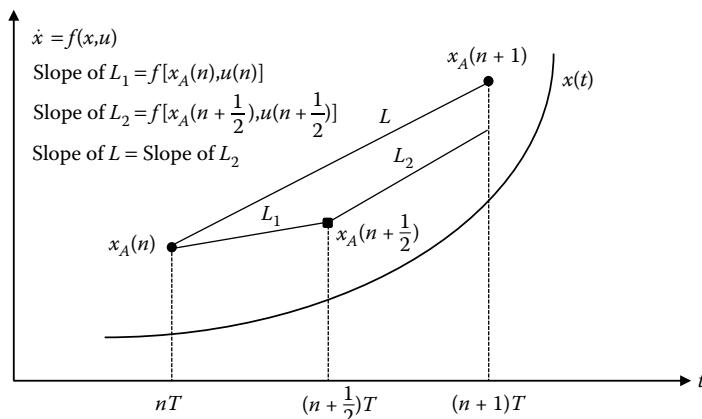


FIGURE 3.22 Illustration of the modified Euler method.

Note the bracketed term multiplying $\underline{x}_A(n)$ in Equation 3.187 is identical to the bracketed term multiplying $\underline{x}_A(n)$ in Equation 3.174 in the improved Euler method.

The modified Euler integrator requires input sampling at twice the normal frequency of $1/T$ due to the presence of the term $\underline{u}(n+1/2)$ in Equation 3.187.

3.7.3 DISCRETE-TIME SYSTEM MATRICES

For autonomous systems governed by $\dot{\underline{x}} = A\underline{x}$, the discrete state $\underline{x}_A(n)$ is updated according to

$$\underline{x}_A(n+1) = G\underline{x}_A(n) \quad (3.188)$$

where the discrete-time system matrix G for each of the five numerical integrators is

$$\text{Explicit Euler: } G = I + TA \quad (3.189)$$

$$\text{Implicit Euler: } G = (I - TA)^{-1} \quad (3.190)$$

$$\text{Trapezoidal: } G = \left[\left(I - \frac{1}{2}TA \right)^{-1} \left(I + \frac{1}{2}TA \right) \right] \quad (3.191)$$

$$\text{Improved Euler: } G = I + (TA) + \frac{1}{2}(TA)^2 \quad (3.192)$$

$$\text{Modified Euler: } G = I + (TA) + \frac{1}{2}(TA)^2 \quad (3.193)$$

It follows directly from Equation 3.188 that the general solution for $\underline{x}_A(n)$ is given by

$$\underline{x}_A(n) = G^n \underline{x}_A(0), \quad n = 1, 2, 3, \dots \quad (3.194)$$

The discrete-time state transition matrix $\Phi(n)$ is defined as

$$\Phi(n) = G^n \quad (3.195)$$

From Equations 3.194 and 3.195,

$$\underline{x}_A(n) = \Phi(n) \underline{x}_A(0) \quad (3.196)$$

From Equation 2.128 in [Chapter 2](#), Section 2.6, the solution to $\dot{\underline{x}} = A\underline{x}$ is

$$\underline{x}(t) = \Phi(t) \underline{x}(0) \quad (3.197)$$

where $\Phi(t)$ is expressible as an infinite series (see Equation 2.129).

$$\Phi(t) = I + (tA) + \frac{1}{2!}(tA)^2 + \frac{1}{3!}(tA)^3 + \frac{1}{4!}(tA)^4 \dots \quad (3.198)$$

The difference between the continuous and discrete responses at $t = t_n = nT$ is

$$\underline{x}(t_n) - \underline{x}_A(n) = \Phi(t_n)\underline{x}(0) - \Phi(n)\underline{x}_A(0) \quad (3.199)$$

Substituting $\Phi(n) = G^n$ from Equation 3.195 along with the fact that $\underline{x}_A(0) = \underline{x}(0)$,

$$\underline{x}(t_n) - \underline{x}_A(n) = [\Phi(t_n) - G^n]\underline{x}(0) \quad (3.200)$$

$$= \left[\left\{ I + (t_n A) + \frac{1}{2!} (t_n A)^2 + \frac{1}{3!} (t_n A)^3 + \frac{1}{4!} (t_n A)^4 + \dots \right\} - G^n \right] \underline{x}(0) \quad (3.201)$$

$$= \left[\left\{ I + (nT)A + \frac{(nT)^2}{2!} A^2 + \frac{(nT)^3}{3!} A^3 + \frac{(nT)^4}{4!} A^4 + \dots \right\} - G^n \right] \underline{x}(0) \quad (3.202)$$

Confining our attention to the explicit numerical integrators, Equation 3.202 becomes

Explicit Euler:

$$\underline{x}(t_n) - \underline{x}_A(n) = \left[\left\{ I + (nT)A + \frac{(nT)^2}{2!} A^2 + \frac{(nT)^3}{3!} A^3 + \frac{(nT)^4}{4!} A^4 + \dots \right\} - (I + TA)^n \right] \underline{x}(0) \quad (3.203)$$

$$\begin{aligned} \text{Improved or modified Euler: } & \underline{x}(t_n) - \underline{x}_A(n) = \left[\left\{ I + (nT)A + \frac{(nT)^2}{2!} A^2 + \frac{(nT)^3}{3!} A^3 + \frac{(nT)^4}{4!} A^4 + \dots \right\} \right. \\ & \quad \left. - \left\{ I + TA + \frac{1}{2}(TA)^2 \right\}^n \right] \underline{x}(0) \end{aligned} \quad (3.204)$$

The difference $\underline{x}(t_n) - \underline{x}_A(n)$, after a single step ($n = 1$), is

$$\begin{aligned} \text{Explicit Euler: } & \underline{x}(T) - \underline{x}_A(1) = \left[\left\{ I + TA + \frac{1}{2!} (TA)^2 + \frac{1}{3!} (TA)^3 + \frac{1}{4!} (TA)^4 \dots \right\} - (I + TA) \right] \underline{x}(0) \\ & \quad = \left[\frac{1}{2!} (TA)^2 + \frac{1}{3!} (TA)^3 + \frac{1}{4!} (TA)^4 \dots \right] \underline{x}(0) \end{aligned} \quad (3.205)$$

$$= \left[\frac{1}{2!} (TA)^2 + \frac{1}{3!} (TA)^3 + \frac{1}{4!} (TA)^4 \dots \right] \underline{x}(0) \quad (3.206)$$

$$\begin{aligned} \text{Improved or modified Euler: } & \underline{x}(T) - \underline{x}_A(1) = \left[\left\{ I + TA + \frac{1}{2!} (TA)^2 + \frac{1}{3!} (TA)^3 + \frac{1}{4!} (TA)^4 \dots \right\} \right. \\ & \quad \left. - \left\{ I + TA + \frac{1}{2}(TA)^2 \right\} \right] \underline{x}(0) \end{aligned} \quad (3.207)$$

$$= \left[\frac{1}{3!}(TA)^3 + \frac{1}{4!}(TA)^4 + \dots \right] \underline{x}(0) \quad (3.208)$$

The increase in accuracy of the improved and modified Euler integrators compared with the explicit Euler is apparent when comparing Equations 3.206 and 3.208.

EXAMPLE 3.10

Consider once again the interacting tanks in Example 3.8. Suppose there is no external input flows into either tank, that is, $F_1(t) = F_2(t) = 0$, $t \geq 0$. The state equations reduce to

$$\begin{bmatrix} \frac{dH_1}{dt} \\ \frac{dH_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_{12}} & \frac{1}{A_1 R_{12}} \\ \frac{1}{A_2 R_{12}} & -\frac{(R_1 + R_2)}{A_1 R_{12} R_2} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \quad (3.209)$$

The initial conditions are $H_1(0) = 10$ ft, $H_2(0) = 0$ ft.

The system parameter values are unchanged, namely

$R_{12} = 2$ ft per ft^3/min , $R_2 = 0.5$ ft per ft^3/min , $A_1 = 15 \text{ ft}^2$, $A_2 = 10 \text{ ft}^2$

The state vector is designated as $\underline{H}(t) = \begin{bmatrix} H_1(t) \\ H_2(t) \end{bmatrix}$

- a. Find the continuous system matrix A and the continuous transition matrix $\Phi(t)$.
- b. Find the discrete transition matrix $\Phi(n)$ for explicit and improved Euler integration.
- c. Compare $\Phi(t)|_{t=T}$ and $\Phi(n)|_{n=1}$ for $T = 0.1, 1$, and 2.5 .
- d. Using the results from Part (c), compare $\underline{H}(T)$ and $\underline{H}_A(1)$ for $T = 0.1, 0.5$, and 2.5 .
- e. Find and plot the transient responses for $\underline{H}_A(n)$ and $\underline{H}(t)$ for $T = 0.1, 0.5$, and 2.5 .

- a. The state equation for the autonomous system is

$$\begin{bmatrix} \frac{dH_1}{dt} \\ \frac{dH_2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_{12}} & \frac{1}{A_1 R_{12}} \\ \frac{1}{A_2 R_{12}} & -\frac{(R_1 + R_2)}{A_1 R_{12} R_2} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \quad (3.210)$$

Using the numerical values for the system parameters, $A = \begin{bmatrix} -0.0333 & 0.0333 \\ 0.05 & -0.25 \end{bmatrix}$

Finding the matrix $\Phi(t)$ is covered in [Chapter 4](#). The result is

$$\Phi(t) = \begin{bmatrix} 0.0321e^{-0.2574t} + 0.9679e^{-0.0259t} & 0.1440e^{-0.2574t} - 0.1440e^{-0.0259t} \\ 0.2159e^{-0.2574t} - 0.2159e^{-0.0259t} & 0.9679e^{-0.2574t} + 0.0321e^{-0.0259t} \end{bmatrix} \quad (3.211)$$

- b. From Equations 3.189 and 3.192

$$\text{Explicit Euler: } \Phi(n) = G^n = (I + TA)^n = \begin{bmatrix} 1 - 0.0333T & 0.0333T \\ 0.05T & 1 - 0.25T \end{bmatrix}^n \quad (3.212)$$

TABLE 3.7
Comparison of Discrete and Continuous Transition Matrices for Several Values of T

T	$\Phi(n) _{n=1}$		$\Phi(t) _{t=T}$
	Explicit Euler	Improved Euler	Continuous
0.1	$\begin{bmatrix} 0.9967 & 0.0033 \\ 0.0050 & 0.9750 \end{bmatrix}$	$\begin{bmatrix} 0.9967 & 0.0033 \\ 0.0049 & 0.9753 \end{bmatrix}$	$\begin{bmatrix} 0.9967 & 0.0033 \\ 0.0049 & 0.9753 \end{bmatrix}$
1	$\begin{bmatrix} 0.9967 & 0.0333 \\ 0.0500 & 0.7500 \end{bmatrix}$	$\begin{bmatrix} 0.9681 & 0.0286 \\ 0.0429 & 0.7821 \end{bmatrix}$	$\begin{bmatrix} 0.9680 & 0.0290 \\ 0.0435 & 0.7795 \end{bmatrix}$
2.5	$\begin{bmatrix} 0.9167 & 0.0833 \\ 0.1250 & 0.3750 \end{bmatrix}$	$\begin{bmatrix} 0.9253 & 0.0538 \\ 0.0807 & 0.5755 \end{bmatrix}$	$\begin{bmatrix} 0.9241 & 0.0593 \\ 0.0889 & 0.5386 \end{bmatrix}$

$$\text{Improved Euler: } \Phi(n) = G^n = \left[I + (TA) + \frac{1}{2}(TA)^2 \right]^n = \begin{bmatrix} 1 - 0.0333T + 0.0014T^2 & 0.0333T - 0.0047T^2 \\ 0.05T - 0.0071T^2 & 1 - 0.25T + 0.0321T^2 \end{bmatrix} \quad (3.213)$$

c. $\Phi(t)|_{t=T}$ and $\Phi(n)|_{n=1}$ for $T = 0.1, 1$ and 2.5 are shown in [Table 3.7](#).

d. From Equations 3.212 and 3.213, $\underline{H}_A(1)$ and $\underline{H}(T)$ are

$T = 0.1$:

$$\text{Explicit Euler: } \underline{H}_A(1) = \Phi(n)|_{n=1} \underline{H}(0) = \begin{bmatrix} 0.9967 & 0.0033 \\ 0.0050 & 0.9750 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.9667 \\ 0.0500 \end{bmatrix} \quad (3.214)$$

$$\text{Improved Euler: } \underline{H}_A(1) = \Phi(n)|_{n=1} \underline{H}(0) = \begin{bmatrix} 0.9967 & 0.0033 \\ 0.0049 & 0.9753 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.9668 \\ 0.0493 \end{bmatrix} \quad (3.215)$$

$$\text{Continuous: } \underline{H}(T) = \Phi(t)|_{t=T} \underline{H}(0) = \begin{bmatrix} 0.9967 & 0.0033 \\ 0.0049 & 0.9753 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.9668 \\ 0.0493 \end{bmatrix} \quad (3.216)$$

$T = 1$:

$$\text{Explicit Euler: } \underline{H}_A(1) = \Phi(n)|_{n=1} \underline{H}(0) = \begin{bmatrix} 0.9967 & 0.0333 \\ 0.0500 & 0.7500 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.6667 \\ 0.5000 \end{bmatrix} \quad (3.217)$$

$$\text{Improved Euler: } \underline{H}_A(1) = \Phi(n)|_{n=1} \underline{H}(0) = \begin{bmatrix} 0.9681 & 0.0286 \\ 0.0429 & 0.7821 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.6806 \\ 0.4292 \end{bmatrix} \quad (3.218)$$

$$\text{Continuous: } \underline{H}(T) = \Phi(t)|_{t=T} \underline{H}(0) = \begin{bmatrix} 0.9680 & 0.0290 \\ 0.0435 & 0.7795 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.6797 \\ 0.4349 \end{bmatrix} \quad (3.219)$$

$T = 2.5$:

$$\text{Explicit Euler: } \underline{H}_A(1) = \Phi(n)|_{n=1} \underline{H}(0) = \begin{bmatrix} 0.9167 & 0.0833 \\ 0.1250 & 0.3750 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.1667 \\ 1.2500 \end{bmatrix} \quad (3.220)$$

$$\text{Improved Euler: } \underline{H}_A(1) = \Phi(n)|_{n=1} \underline{H}(0) = \begin{bmatrix} 0.9253 & 0.0538 \\ 0.0807 & 0.5755 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.2535 \\ 0.8073 \end{bmatrix} \quad (3.221)$$

$$\text{Continuous: } \underline{H}(T) = \Phi(t)|_{t=T} \underline{H}(0) = \begin{bmatrix} 0.9241 & 0.0593 \\ 0.0889 & 0.5386 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 9.2408 \\ 0.8895 \end{bmatrix} \quad (3.222)$$

The results are summarized in [Table 3.8](#).

As expected, improved Euler integration is more accurate than explicit Euler integration. Furthermore, at the smallest time step, namely $T = 0.1$, the improved Euler and continuous responses agree to at least 4 places after the decimal point.

- e. The discrete response is $\underline{H}_A(n) = \Phi(n)\underline{H}_A(0)$, $n = 0, 1, 2, \dots$. However, it is computationally more efficient to use

$$\underline{H}_A(n+1) = \Phi(1)\underline{H}_A(n), n = 0, 1, 2, \dots \quad (3.223)$$

The continuous state response $\underline{H}(t) = \Phi(t)\underline{x}(0)$ is

$$\begin{bmatrix} H_1(t) \\ H_2(t) \end{bmatrix} = \begin{bmatrix} 0.0321e^{-0.2574t} + 0.9679e^{-0.0259t} & 0.1440e^{-0.2574t} - 0.1440e^{-0.0259t} \\ 0.2159e^{-0.2574t} - 0.2159e^{-0.0259t} & 0.9679e^{-0.2574t} + 0.0321e^{-0.0259t} \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad (3.224)$$

$\underline{H}_A(n)$ for $T = 0.1, 1$ and 2.5 min and $\underline{H}(t)$ are plotted in [Figures 3.23–3.28](#).

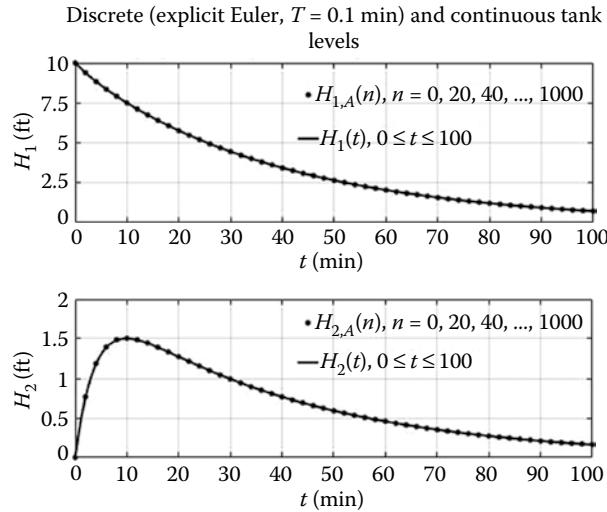
Note the moderate error in $H_{2,A}(n)$, early in the response, using explicit Euler integration with $T = 1$ min. As expected, the error is more pronounced when $T = 2.5$ min. With improved Euler integration, the error is moderate, even with the larger step size.

We now focus on the discrete approximation of a second-order system step response using modified Euler integration. Starting with

$$\frac{d^2}{dt^2}y(t) + 2\zeta\omega_n \frac{dy}{dt}y(t) + \omega_n^2y(t) = K\omega_n^2u(t) \quad (3.225)$$

TABLE 3.8
Comparison of Discrete and Continuous Responses for Several Values of T

T	$\underline{H}_A(1)$			$\underline{H}(T)$
	Explicit Euler	Improved Euler	Continuous	
0.1	$\begin{bmatrix} 9.9667 \\ 0.0500 \end{bmatrix}$	$\begin{bmatrix} 9.9668 \\ 0.0493 \end{bmatrix}$	$\begin{bmatrix} 9.9668 \\ 0.0493 \end{bmatrix}$	
1	$\begin{bmatrix} 9.6667 \\ 0.5000 \end{bmatrix}$	$\begin{bmatrix} 9.6806 \\ 0.4292 \end{bmatrix}$	$\begin{bmatrix} 9.6797 \\ 0.4349 \end{bmatrix}$	
2.5	$\begin{bmatrix} 9.1667 \\ 1.2500 \end{bmatrix}$	$\begin{bmatrix} 9.2535 \\ 0.8073 \end{bmatrix}$	$\begin{bmatrix} 9.2408 \\ 0.8895 \end{bmatrix}$	

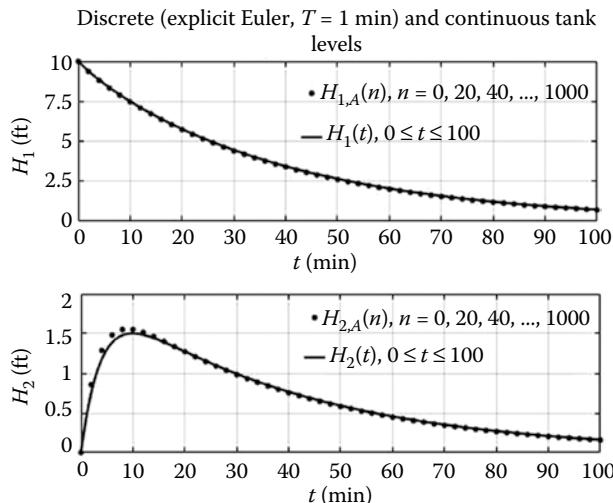
**FIGURE 3.23** Discrete (explicit Euler, $T = 0.1$ min) and continuous responses.

System parameters are $\zeta = 0.5$, $\omega_n = 0.4$ rad/s, $K = 2$. Both initial conditions are zero. Choosing states $x_1 = y$ and $x_2 = dy/dt$, the state equations for this second-order system are

$$\frac{dx_1}{dt} = x_2 \quad (3.226)$$

$$\frac{dx_2}{dt} = K\omega_n^2 u - \omega_n^2 x_1 - 2\zeta\omega_n x_2 \quad (3.227)$$

$$y_1 = y = x_1 \quad (3.228)$$

**FIGURE 3.24** Discrete (explicit Euler, $T = 1$ min) and continuous responses.

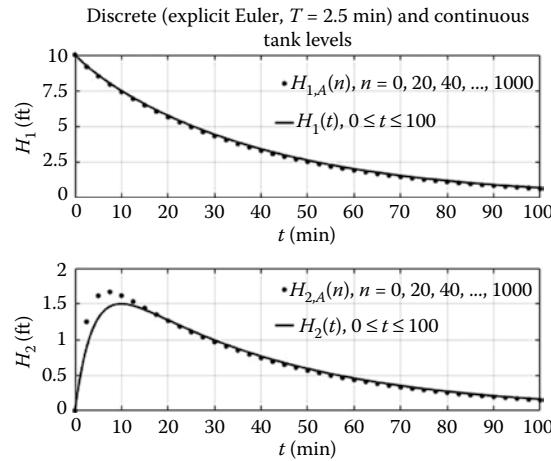


FIGURE 3.25 Discrete (explicit Euler, $T = 2.5$ min) and continuous responses.

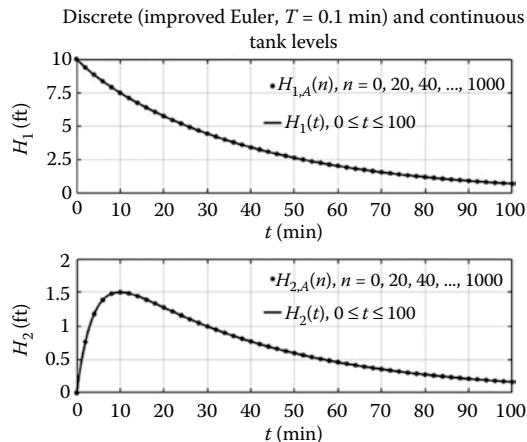


FIGURE 3.26 Discrete (improved Euler, $T = 0.1$ min) and continuous responses.

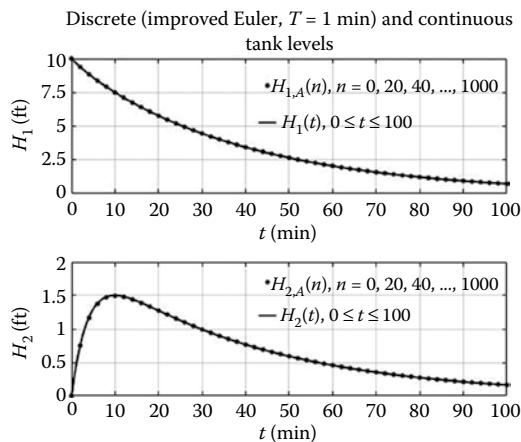


FIGURE 3.27 Discrete (improved Euler, $T = 1$ min) and continuous responses.

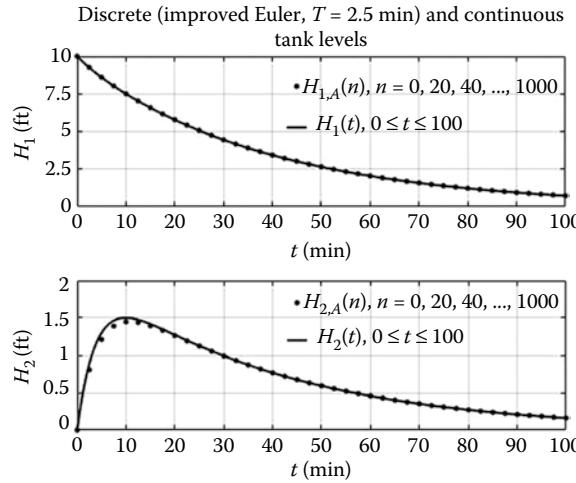


FIGURE 3.28 Discrete (improved Euler, $T = 2.5$ min) and continuous responses.

The matrices A , B , C , and D in the state equations are

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ K\omega_n^2 \end{bmatrix}, \quad C = [1 \quad 0], \quad D = [0]$$

The discrete system matrix (see Equation 3.193) is

$$\begin{aligned} G &= I + (TA) + \frac{1}{2}(TA)^2 \\ &= I + T \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} + \frac{1}{2}T^2 \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}^2 \end{aligned} \quad (3.229)$$

$$= \begin{bmatrix} 1 - \frac{1}{2}(\omega_n T)^2 & T(1 - \zeta\omega_n T) \\ -\omega_n^2 T(1 - \zeta\omega_n T) & 1 - 2\zeta\omega_n T + \frac{1}{2}(\omega_n T)^2(4\zeta^2 - 1) \end{bmatrix} \quad (3.230)$$

Substituting the numerical values for the system parameters gives $G = \begin{bmatrix} 0.980 & 0.45 \\ -0.072 & 0.80 \end{bmatrix}$. The discrete state is updated using Equation 3.187,

$$\underline{x}_A(n+1) = G\underline{x}_A(n) + \frac{1}{2}T^2AB\underline{u}(n) + TB\underline{u}(n + \frac{1}{2}) \quad (3.231)$$

$$= \begin{bmatrix} 0.980 & 0.45 \\ -0.072 & 0.80 \end{bmatrix} \underline{x}_A(n) + \frac{1}{2}(0.5)^2 \begin{bmatrix} 0 & 1 \\ -0.16 & -0.4 \end{bmatrix} \begin{bmatrix} 0 \\ 0.32 \end{bmatrix} [1] + (0.5) \begin{bmatrix} 0 \\ 0.32 \end{bmatrix} [1] \quad (3.232)$$

$$= \begin{bmatrix} 0.980 & 0.45 \\ -0.072 & 0.80 \end{bmatrix} \underline{x}_A(n) + \begin{bmatrix} 0.040 \\ 0.144 \end{bmatrix} [1] \quad (3.233)$$

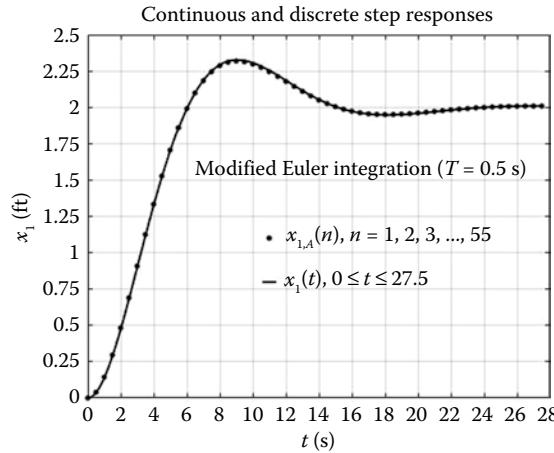


FIGURE 3.29 Continuous and discrete (modified Euler, $T = 0.5$ s) step responses of a second-order system.

Note that $\underline{u}(n)$ and $\underline{u}(n + (1/2))$ in Equation 3.231 are both equal to the 1×1 vector [1], Equation 3.233 is solved recursively in “Ch3_step.m” and $x_{1,A}(n)$ is plotted in Figure 3.29. The step response for $y(t) = x_1(t)$ is (see Chapter 2, Equation 2.23)

$$x_1(t) = K \left[1 - e^{-\zeta \omega_n t} \left(\cos \omega_d t + \frac{\zeta \omega_n}{\omega_d} \sin \omega_d t \right) \right], \quad t \geq 0 \quad (3.234)$$

The damped natural ω_d frequency is computed from its definition

$$\omega_d = \left(\sqrt{1 - \zeta^2} \right) \omega_n = \left(\sqrt{1 - 0.5^2} \right) 0.4 = \frac{\sqrt{3}}{5} \text{ rad/s}$$

Substituting the system parameter values into the equation for $x_1(t)$ and simplifying leads to

$$x_1(t) = 2 \left[1 - e^{-t/5} \left\{ \cos \left(\frac{\sqrt{3}}{5} t \right) + \frac{\sqrt{3}}{3} \sin \left(\frac{\sqrt{3}}{5} t \right) \right\} \right], \quad t \geq 0 \quad (3.235)$$

The discrete response, $x_{1,A}(n)$, with integration step size $T = 0.5$ s, is an accurate representation of the continuous response $x_1(t)$ at discrete times $t_n = nT$, $n = 0, 1, 2, \dots$

A sample of the results for the discrete and continuous responses are compiled in Table 3.9. As a final check on the results, the predicted steady-states (discrete and continuous) are

$$\lim_{x \rightarrow \infty} x_A(n) = \lim_{x \rightarrow \infty} x(t) = A^{-1} B \underline{u}^0 \quad \text{where} \quad \underline{u}^0 = [1].$$

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(0.4)^2 & -2(0.5)(0.4) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.16 & -0.4 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ K\omega_n^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2(0.4)^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.32 \end{bmatrix}$$

$$\underline{x}_A(\infty) = \underline{x}(\infty) = - \begin{bmatrix} 0 & 1 \\ -0.16 & -0.4 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0.32 \end{bmatrix} [1] = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

TABLE 3.9
Summary of Results for Discrete $x_{1,A}(n)$ and Continuous $x_1(t)$

n	$x_{1,A}(n)$	t_n	$x_1(t_n)$	n	$x_{1,A}(n)$	t_n	$x_1(t_n)$
0	0	0	0	30	2.0038	15	2.0046
5	0.6904	2.5	0.6806	35	1.9509	17.5	1.9487
10	1.7046	5	1.6989	40	1.9604	20	1.9580
15	2.2447	7.5	2.2487	45	1.9869	22.5	1.9859
20	2.2979	10	2.3062	50	2.0042	25	2.0043
25	2.1434	12.5	2.1492	55	2.0080	27.5	2.0086

in agreement with the graphs of $x_1(t)$ and $x_{1,A}(n)$. While $x_2(t)$ and $x_{2,A}(n)$ are not plotted, it's clear that $\lim_{n \rightarrow \infty} x_{2,A}(n) = \lim_{t \rightarrow \infty} x_2(t) = 0$ because $x_2(t) = \frac{dy}{dt}$ which is zero by definition at steady-state.

Our last example is that of a nonlinear second-order system. The equations developed in this and previous sections for linear systems are not applicable; however the implementation of numerical integration is nonetheless straightforward. A state variable model of the nonlinear system is required. The discrete state is updated using the state derivative functions in accordance with the desired numerical integration routine.

EXAMPLE 3.11

A simple nonlinear pendulum with damping is shown in Figure 3.30.

The mass of the rod is negligible compared to the mass m of the sphere. Linear damping at the fixed end is assumed. The angular position of the rod $\theta(t)$ satisfies the nonlinear differential equation

$$J\ddot{\theta} + c\dot{\theta} + mgr \sin \theta = 0 \quad (3.236)$$

- Find the nonlinear state equations when $x_1 = \theta$ and $x_2 = \dot{\theta}$.
- Find the difference equations for updating the discrete state components $x_{1,A}(n)$ and $x_{2,A}(n)$ when explicit Euler integration is used.
- Numerical values of the system parameters are $m = 0.25$ slugs, $r = 0.75$ ft, $c = 0.1$ ft lb per rad/s. The moment of inertia $J = 0.1406$ ft lb s².

Find a suitable value for T and solve the discrete state equations recursively under the following conditions:

- $\theta(0) = \pi/6$ rad, $\dot{\theta} = 0$ rad/s.
- $\theta(0) = 0$ rad, $\dot{\theta} = 0.5$ rad/s.

Graph $x_{1,A}(n)$ and $x_{2,A}(n)$ for both sets of initial conditions.

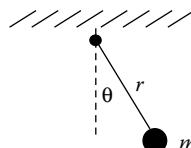


FIGURE 3.30 A simple nonlinear pendulum with damping.

$$\text{a. } \dot{x}_1 = \dot{\theta} = x_2 \quad (3.237)$$

$$\dot{x}_2 = \ddot{\theta} = \frac{1}{J}[-mgr \sin \theta - cx_2] \quad (3.238)$$

$$= \frac{1}{J}(-mgr \sin x_1 - cx_2) \quad (3.239)$$

The continuous state equations are

$$\dot{x}_1 = f_1(x_1, x_2) = x_2 \quad (3.240)$$

$$\dot{x}_2 = f_2(x_1, x_2) \quad (3.241)$$

$$= \frac{1}{J}(-mgr \sin x_1 - cx_2) \quad (3.242)$$

b. Using explicit Euler integration, the difference equations for updating the discrete state are

$$x_{1,A}(n+1) = x_{1,A}(n) + Tf_1[x_{1,A}(n), x_{2,A}(n)] \quad (3.243)$$

$$= x_{1,A}(n) + Tx_{2,A}(n) \quad (3.244)$$

$$x_{2,A}(n+1) = x_{2,A}(n) + Tf_2[x_{1,A}(n), x_{2,A}(n)] \quad (3.245)$$

$$= x_{2,A}(n) - \frac{T}{J}[mgr \sin x_{1,A}(n) + cx_{2,A}(n)] \quad (3.246)$$

c. Choosing $T = 0.0025$ s, a recursive solution of Equations 3.244 and 3.246 is easily obtained. The results for

1. $\theta(0) = \pi/6$ rad, $\dot{\theta} = 0$ rad/s
2. $\theta(0) = 0$ rad, $\dot{\theta} = 0.5$ rad/s

are shown in [Figures 3.31](#) and [3.32](#).

Exact solutions for the state components are not easily obtained owing to the nonlinearity in Equation 3.236. A “quasi exact” solution could be found by choosing an exceedingly small value of T and plotting the results on the same graph for comparison with the discrete approximations shown in [Figures 3.31](#) and [3.32](#). It’s left as an exercise to show that the discrete and “quasi exact” responses are in basic agreement.

Looking at the graphs in [Figures 3.31](#) and [3.32](#), we might be inclined to believe that the integration step size $T = 0.0025$ s is a “one size fits all” value for simulating the pendulum dynamics. However, [Figure 3.33](#) will quickly dispel this thinking. The results shown in [Figure 3.33](#) correspond to an undamped pendulum ($c = 0$) with the same initial conditions as in part (c) and the same step size of 0.0025 s. Every 20th point of the discrete state responses are plotted.

Clearly, explicit Euler integration using a step size of $T = 0.0025$ s is not advisable since the discrete state responses $x_{1,A}(n)$ and $x_{2,A}(n)$ bear no resemblance whatsoever to the real (continuous) system responses. A valuable lesson of this example is the need to exercise caution when choosing the integration step for numerical integration. If we are not careful, the numerical integrators may be “unstable” under certain conditions. This point is revisited in detail in [Chapter 8](#).

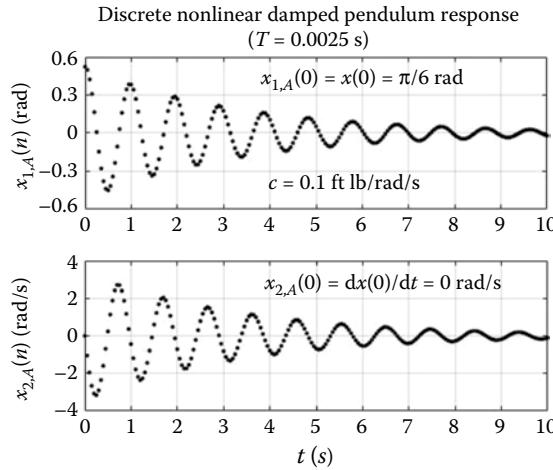


FIGURE 3.31 Damped pendulum response using explicit Euler ($T = 0.0025$ s) $x_1(0) = \pi/6$ rad, $x_2(0) = 0$ rad/s.

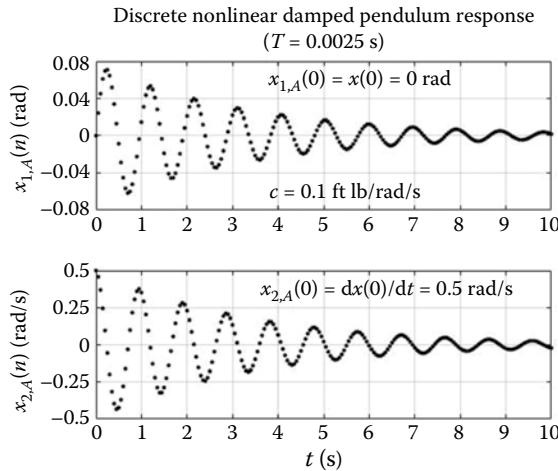


FIGURE 3.32 Damped pendulum response using explicit Euler ($T = 0.0025$ s) $x_1(0) = 0$ rad, $x_2(0) = 0.5$ rad/s.

EXERCISES

- 3.25 By trial and error, determine an acceptable value for the step size T in simulating the nonlinear pendulum response in Example 3.11 using improved Euler integration. The initial conditions are $x_1(0) = \pi/6$ rad, $x_2(0) = 0.5$ rad/s. Plot the discrete state $x_{1,A}(n)$, $n = 0, 1, 2, \dots, n_f$ where $n_f T = 10$ s for each value of T .
- 3.26 Repeat Problem 3.25 using trapezoidal integration instead of improved Euler.
- 3.27 Choose a very small time step, e.g. $T = 0.0001$ s in Example 3.11 to obtain the “quasi exact” solution and plot the results on the same graph with the discrete responses in Figures 3.31 and 3.32. Comment on the results.
- 3.28 The nonlinear pendulum model in Example 3.11 is often approximated by

$$J\ddot{\theta} + c\dot{\theta} + mrg\theta = 0$$

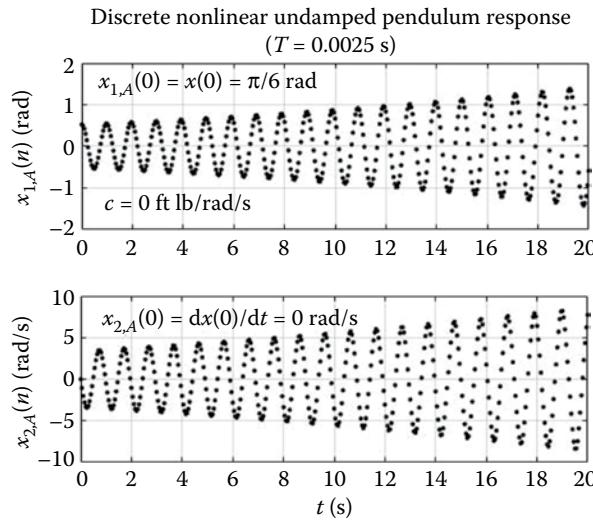


FIGURE 3.33 Undamped pendulum response using explicit Euler ($T = 0.0025$ s) $x_1(0) = \pi/6$ rad, $x_2(0) = 0$ rad/s.

when the angular displacement θ is small, i.e. the small angle approximation $\theta = \sin\theta$ is used resulting in the linear differential equation model above. Compare the results of simulating the linear and nonlinear models using modified Euler integration. The initial angle $\theta(0) = 5$ deg and the initial angular velocity $\dot{\theta}(0) = 0$ deg/s.

3.29 A logistic population growth model

$$\frac{dP}{dt} = cP(P_m - P)$$

is to be simulated in order to approximate the population $P(t)$ for a period of time.

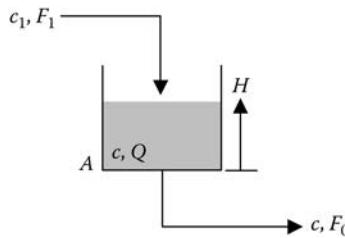
- Find the difference equation for $P_A(n)$ intended to approximate $P(t)$ based on the use of the following numerical integrators
 - Explicit Euler ($T = 0.25$ yr)
 - Trapezoidal ($T = 0.5$ yr)
 - Improved Euler ($T = 0.5$ yr)
- Fill in Table E3.29 with the simulated populations based on the three numerical integrators and the exact solution. Note, $c = 1.25 \times 10^{-9}$, $P_m = 25$ million and $P(0) = 5$ million. The exact solution is given by

$$P(t) = \frac{P_m P(0)}{P(0) + [P_m - P(0)]e^{-cP_m t}} \quad t \geq 0$$

TABLE E3.29

t (years)	0	50	100	150	200	250
Explicit Euler	5.0000					
Trapezoidal	5.0000					
Improved Euler	5.0000					
Exact	5.0000					

- 3.30** The tank in Figure E3.30 has a brine solution flowing into it. The solution is stirred well enough so that the concentration of salt in the tank is uniform.

**FIGURE E3.30**

where,

- c_1 is the brine concentration (lb/gal)
- F_1 is the brine flow (gal/min)
- c is the salt concentration in tank (lb/gal)
- Q is the quantity of salt in tank (lb)
- H is the liquid level in tank (ft)
- V is the volume of liquid in tank (gal)
- F_0 is the flow rate from tank (gal/min)

The mathematical model consists of the following equations.

$$\begin{aligned}\frac{dQ}{dt} &= c_1 F_1 - c F_0 \\ c &= \frac{Q}{V}, \quad V = AH \\ A \frac{dH}{dt} + F_0 &= F_1, \quad F_0 = \alpha H^{1/2}\end{aligned}$$

The system baseline parameter values are $A = 25 \text{ ft}^2$, $\alpha = 0.75 \text{ gal/min per ft}^{1/2}$.

(Note: 1 ft^3 of water is roughly 8.3 gal)

- Draw a simulation diagram of the system.
- Choose the state variables as $x_1 = Q$, $x_2 = H$ and the outputs $y_1 = c$, $y_2 = Q$ and $y_3 = V$. Write the state equations in the form

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, c_1, F_1), & y_1 &= g_1(x_1, x_2, c_1, F_1) \\ \dot{x}_2 &= f_2(x_1, x_2, c_1, F_1), & y_2 &= g_2(x_1, x_2, c_1, F_1) \\ && y_3 &= g_3(x_1, x_2, c_1, F_1)\end{aligned}$$

- Find expressions for the steady-state values of the states $x_1(\infty)$ and $x_2(\infty)$ and the outputs $y_1(\infty)$, $y_2(\infty)$ and $y_3(\infty)$ assuming c_1 and F_1 are constant.
- The tank is initially filled with 100 gal of water (no salt). Brine starts flowing in to the tank at a rate of 2 gal/min. The salt concentration of the brine is 0.25 lb/gal. Both the flowrate and salt concentration of the brine flow remain constant. Using explicit Euler and Improved Euler integration, find the discrete state equations

$$\begin{aligned}\underline{x}_A(n+1) &= \underline{f}[(\underline{x}_A(n), \underline{u}(n))] \\ \underline{y}_A(n) &= \underline{g}[(\underline{x}_A(n), \underline{u}(n))]\end{aligned}$$

which are used to obtain an approximate solution for the continuous states and outputs.

- Solve the discrete state equations recursively for the discrete states $x_{1,A}(n)$, $x_{2,A}(n)$ and outputs $y_{1,A}(n)$, $y_{2,A}(n)$ and $y_{3,A}(n)$. Graph the transient responses. Comment on the values of T used for each type of numerical integrator.

- f. Compare the steady-state results obtained in part (e) with the predicted values from part (c). Comment on the results.

3.8 CASE STUDY: VERTICAL ASCENT OF A DIVER

As a diver submerges, pressure increases in direct proportion to the depth. This pressure is caused by the combined weight of the surrounding water and the atmosphere above, and is called ambient pressure. At a depth of 70 feet, ambient pressure is equal to more than three atmospheres (three times the atmospheric pressure at sea level). In order to overcome this pressure and fill his lungs with vital air, the diver must breathe air supplied to him at the ambient pressure.

The air is a mixture of approximately 20 percent oxygen and 80 percent inert nitrogen. The oxygen component of the air is used by the body, and waste carbon dioxide is exhaled. Under normal atmospheric conditions, the nitrogen component of the mixture has no effect. But under pressure, it dissolves in the blood stream and in tissues and remains there after the diver begins to ascend. If the diver ascends too quickly, the nitrogen expands and equalizes with the decreasing ambient pressure. Nitrogen bubbles form in the blood stream and the tissues leading to an extremely painful condition known as Decompression Sickness (DCS), more commonly known as the “bends” which can cause paralysis and even death.

The focus of this study is an investigation of the cable forces that can be used to bring a diver safely to the surface. The mathematical model governing the diver's ascent consists of differential equations relating the forces acting on the diver and the dynamics of the diver's internal body pressure [Klamrock]. The following notation is used:

$h = h(t)$ is the depth of diver below sea level, ft

$\dot{h} = dh/dt$ is the velocity of diver, ft/s

$\ddot{h} = d^2h/dt^2$ is the acceleration of diver, ft/s²

$p = p(t)$ is the internal body pressure of diver, relative to atmospheric pressure at sea level, lb/ft²

$\dot{p} = dp/dt$ is the rate of change of diver's internal body pressure, lb/ft² per s

$f_C = f_C(t)$ external cable force on diver, lb

$f_D = f_D(t)$ is the drag force on diver, lb

f_B is the buoyant force on diver, lb

m is the mass of diver, slugs

W is the weight of diver and gear at sea level, lb

V is the volume of diver and gear, ft³

K is the body tissue constant of diver, s⁻¹

μ is the drag coefficient of diver under water, lb s/ft

γ is the weight density of water (62.4 lb/ft³)

g is the gravitational constant (32.2 ft/s²)

The forces acting on the diver are a cable force f_C , a drag force f_D , a buoyant force f_B , and the diver's weight W . From Newton's second law

$$\ddot{m\dot{h}} = W - f_B + f_D - f_C \quad (3.247)$$

with h and all forces measured positive in the downward direction. The drag force is modeled by

$$f_D = -\mu\dot{h} \quad (3.248)$$

The buoyant force is equal to the weight of water displaced by the diver and gear

$$f_B = \gamma V \quad (3.249)$$

Combining Equations 3.247, 3.248 and 3.249 gives

$$\frac{W}{g} \ddot{h} + \mu \dot{h} = (W - \gamma V) - f_C \quad (3.250)$$

The right-hand side of Equation 3.250 is the difference between the diver's effective weight in the water, $(W - \gamma V)$, and the cable force f_C . Denoting the net cable force by

$$f_N = (W - \gamma V) - f_C \quad (3.251)$$

leads to the second-order differential equation

$$\frac{W}{g} \ddot{h} + \mu \dot{h} = f_N \quad (3.252)$$

The rate of change of the diver's internal body pressure is assumed proportional to the difference between the local underwater (ambient) pressure and the diver's internal body pressure. That is,

$$\dot{p} = K(\gamma h - p) \quad (3.253)$$

We are interested in h , the diver's depth below the surface, and Δp , the difference between the internal body pressure of the diver and the ambient underwater pressure. The dynamic system under investigation is portrayed in [Figure 3.34](#).

The third order linear dynamic system can be modeled in state variable form. The state variables are chosen as

$$\left. \begin{array}{l} x_1 = h \\ x_2 = \dot{h} \\ x_3 = p \end{array} \right\} \quad (3.254)$$

Solving for the state derivatives

$$\dot{x}_1 = \dot{h} = x_2 \quad (3.255)$$

$$\dot{x}_2 = \ddot{h} = -\frac{\mu g}{W} \dot{h} + \frac{g}{W} f_N \quad (3.256)$$

$$= -\frac{\mu g}{W} x_2 + \frac{g}{W} f_N \quad (3.257)$$

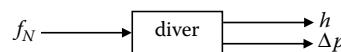


FIGURE 3.34 Dynamic system with input f_N and outputs h and Δp .

$$\dot{x}_3 = \dot{p} = K\gamma x_1 - Kx_3 \quad (3.258)$$

The outputs are expressed in terms of the states as

$$y_1 = h = x_1 \quad (3.259)$$

$$y_2 = p - \gamma h = x_3 - \gamma x_1 \quad (3.260)$$

The complete state equations are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{-\mu g}{W} & 0 \\ K\gamma & 0 & -K \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{g}{W} \\ 0 \end{bmatrix} [f_N] \quad (3.261)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\gamma & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.262)$$

The state equation matrices are given by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{-\mu g}{W} & 0 \\ K\gamma & 0 & -K \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{g}{W} \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ -\gamma & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.263)$$

In order to obtain a numerical solution to the state equations, the initial conditions, or initial state $\underline{x}(0)$, must be known. Assuming the diver is initially in equilibrium with his or her surroundings leads to

$$\dot{h}(0) = x_2(0) = 0 \quad (3.264)$$

$$\dot{p}(0) = K[\gamma h(0) - p(0)] \quad (3.265)$$

$$= K[\gamma x_1(0) - x_3(0)] \quad (3.266)$$

Setting $\dot{p}(0) = 0$ in Equation 3.266 and solving for $x_3(0)$ gives

$$x_3(0) = \gamma x_1(0) \quad (3.267)$$

Initial depth $x_1(0)$ is arbitrary; however, to be in equilibrium, the diver's effective weight in the water, $W - \gamma V$, must be counterbalanced by the initial cable force $f_c(0)$. Therefore,

$$f_c(0) = W - \gamma V \quad (3.268)$$

Note, the initial net force to maintain the diver in equilibrium is

$$f_N(0) = (W - \gamma V) - f_c(0) = 0 \quad (3.269)$$

A simulation of the diver's ascent subject to a constant cable force in excess of $f_c(0)$ in Equation 3.268 is needed. The discrete state equation depends on the choice of numerical integrator. Using trapezoidal integration for now and leaving the other discrete integrators for the exercise problems, the discrete state is updated according to

$$\underline{x}_A(n+1) = \left(I - \frac{1}{2}TA\right)^{-1} \left(I + \frac{1}{2}TA\right) \underline{x}_A(n) + \frac{1}{2} \left(I - \frac{1}{2}TA\right)^{-1} TB[\underline{u}(n) + \underline{u}(n+1)] \quad (3.270)$$

With a constant cable force $f_c = \bar{f}_c$, $t \geq 0$, the input f_N is likewise constant. That is

$$f_N = \bar{f}_N = (W - \gamma V) - \bar{f}_c, \quad t \geq 0 \quad (3.271)$$

The second term in Equation 3.270 can be simplified as follows.

$$\frac{1}{2} \left(I - \frac{1}{2}TA\right)^{-1} TB[\underline{u}(n) + \underline{u}(n+1)] = \frac{1}{2} \left(I - \frac{1}{2}TA\right)^{-1} TB[\bar{f}_N + \bar{f}_N] \quad (3.272)$$

$$= T \left(I - \frac{1}{2}TA\right)^{-1} B \bar{f}_N \quad (3.273)$$

Baseline numerical values for the system parameters are $K = 0.2$, $\mu = 6.5$, $W = 300$, $V = 3$ and the step size $T = 0.25$ s. Evaluating matrices A and B in Equation 3.263,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.6977 & 0 \\ 12.48 & 0 & -0.2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.1073 \\ 0 \end{bmatrix}$$

Combining Equations 3.270 and 3.273 along with the values for matrices A and B results in

$$\underline{x}_A(n+1) = \begin{bmatrix} 1 & 0.2299 & 0 \\ 0 & 0.8396 & 0 \\ 3.0439 & 0.3500 & 0.9512 \end{bmatrix} \underline{x}_A(n) + \begin{bmatrix} 0.0031 \\ 0.0247 \\ 0.0047 \end{bmatrix} \bar{f}_N \quad (3.274)$$

Before simulating the diver's ascent to the surface, we can make the cable force equal to its equilibrium value in Equation 3.268 and observe whether the system remains in equilibrium. Setting $f_c = W - \gamma V = 112.8$ lbs makes the net force $\bar{f}_N = 0$. Additionally, we must remember to make $x_3(0) = \gamma x_1(0)$ where $x_1(0)$ is the arbitrary initial depth.

Figure 3.35 shows the results of solving Equation 3.274 under these conditions with the diver starting at 500 ft below the surface. As expected, the system remains in an equilibrium state.

$$f_c = W - gV = 112.8 \text{ lb}$$

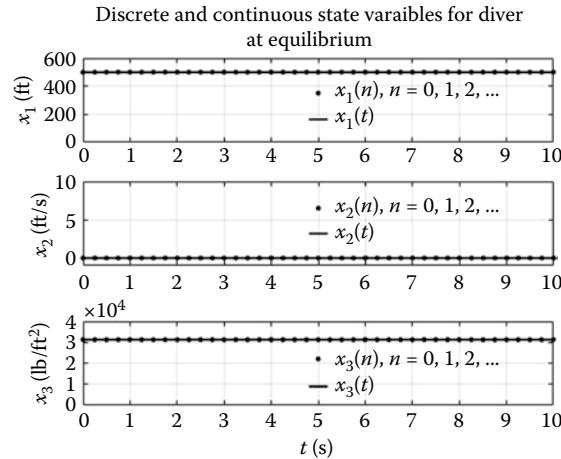


FIGURE 3.35 System at equilibrium: $x_1(0) = 500$ ft, $x_2(0) = 0$ ft/s, $x_3(0) = \gamma x_1(0) = 3120$ lb/ ft^2 $f_c = W - \gamma V = 112.8$ lb.

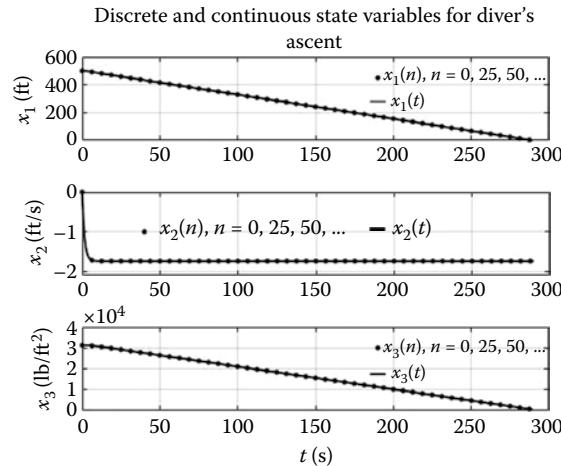


FIGURE 3.36 State variables for diver's ascent: $f_c = 1.1 \times (W - \gamma V) = 124.08$ lb.

Suppose the cable force is increased by 10% above its equilibrium value to $1.1 \times (W - \gamma V) = 1.1 \times 112.8 = 124.08$ lb. The Matlab script file “Ch3_CaseStudy.m” generates a recursive solution to Equation 3.274. The results are plotted in Figure 3.36 for a duration of time sufficient to bring the diver to the surface.

The integration step size T could be varied an order of magnitude in either direction and the results compared to those in Figure 3.36 to determine if the current value $T = 0.25$ s needs to be adjusted.

Since the system dynamics are linear, analytical solutions for the continuous state variables in Equation 3.261, with constant net force $f_N = \bar{f}_N$ are easily determined and given in Equations 3.275–3.277. The derivation is left as an exercise problem at the end of the section.

$$x_1(t) = h(0) + \frac{\bar{g}\bar{f}_N}{\alpha W} \left[t - \frac{1 - e^{-\alpha t}}{\alpha} \right] \quad (3.275)$$

$$x_2(t) = \frac{g\bar{f}_N}{\alpha W} (1 - e^{-\alpha t}) \quad (3.276)$$

$$x_3(t) = \gamma \left[h(0) + \frac{g\bar{f}_N}{\alpha W} \left\{ t + \frac{K(1 - e^{-\alpha t})}{\alpha(\alpha - K)} - \frac{\alpha(1 - e^{-Kt})}{K(\alpha - K)} \right\} \right] \quad (3.277)$$

where the constant $\alpha = \mu g/W$. The analytical solutions for the states are plotted in [Figure 3.36](#) along with the discrete states. There is close agreement between the numerical (discrete) and analytical (continuous) solutions for each of the state variables. Notice that after about 6 s, the diver is surfacing at a constant velocity and both depth and internal body pressure are decreasing linearly with time.

Equation 3.275 can be used to estimate the time required for the diver to surface. If the initial depth is great enough, the exponential term $e^{-\alpha t}$ in the transient component has died out when the diver surfaces. Consequently, the time to surface, t_s , can be estimated from

$$0 = h(0) + \frac{g\bar{f}_N}{\alpha W} \left[t_s - \frac{1}{\alpha} \right] \quad (3.278)$$

$$t_s = \frac{W}{\mu g} - \frac{\mu h(0)}{\bar{f}_N} \quad (3.279)$$

$$= \frac{W}{\mu g} - \frac{\mu h(0)}{(W - \gamma V) - \bar{f}_C} \quad (3.280)$$

$$= \frac{300}{6.5(32.2)} - \frac{6.5(500)}{112.8 - 1.1(112.8)}$$

$$= 289.6 \text{ s}$$

in agreement with the graphs of $x_{1,A}(n)$ and $x_1(t)$ in [Figure 3.36](#).

We have yet to look at the differential pressure $\Delta p = p - \gamma h$, the second component of the output vector y_2 in Equation 3.260. The discrete output $\underline{y}_A(n)$ is given by

$$\underline{y}_A(n) = C\underline{x}_A(n) + D\underline{u}(n) \quad (3.281)$$

The second component of $\underline{y}_A(n)$ reduces to

$$y_{2,A}(n) = C_{2,1}x_{1,A}(n) + C_{2,2}x_{2,A}(n) + C_{2,3}x_{3,A}(n) \quad (3.282)$$

since the direct transmission matrix D is zero. Substituting the components of C in Equation 3.263 into Equation 3.282 gives

$$y_{2,A}(n) = -\gamma x_{1,A}(n) + x_{3,A}(n) \quad (3.283)$$

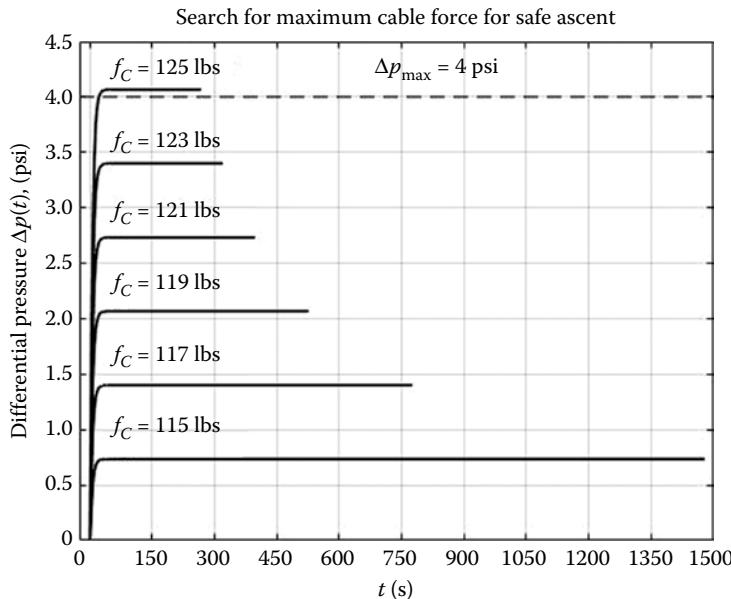


FIGURE 3.37 Differential pressure during ascent of diver for different cable forces.

We are now in a position to investigate the conditions necessary for a safe ascent. Suppose a safe ascent implies the differential pressure Δp is never to exceed a value denoted by Δp_{\max} . The maximum cable force for a safe ascent, $(f_C)_{\max}$, can be obtained by initializing the constant cable force \bar{f}_C slightly more than the diver's effective weight, W_{eff} . That is, $\bar{f}_C > W_{\text{eff}}$, where

$$W_{\text{eff}} = W - \gamma V \quad (3.284)$$

The diver's ascent is then simulated and if the maximum differential pressure during the ascent is less than Δp_{\max} , the ascent is simulated again with a larger cable force. The reverse is true if the maximum differential pressure exceeds Δp_{\max} . The process is repeated until the cable force producing a maximum differential pressure Δp_{\max} (within some tolerance) is obtained. Figure 3.37 shows the result of simulating several ascents to find $(f_C)_{\max}$ for the case when $\Delta p_{\max} = 4$ psi.

The discrete differential pressure responses are labeled and graphed as if they were continuous; however the points along each plot were obtained by recursive solution of difference equations. Note the dramatic increase in ascent time as the cable force approaches the equilibrium value of 112.8 lb.

A second approach to finding the maximum cable force, $(f_C)_{\max}$, for a safe ascent is based on analytical solutions for the state variables $x_1(t)$ and $x_3(t)$. From Equations 3.275 and 3.277, the steady-state responses, after the transient components have died out, are

$$x_1(t)_{ss} = h(0) + \frac{g\bar{f}_N}{\alpha W} \left[t - \frac{1}{\alpha} \right] \quad (3.285)$$

$$x_3(t)_{ss} = \gamma \left[h(0) + \frac{g\bar{f}_N}{\alpha W} \left\{ t + \frac{K}{\alpha(\alpha - K)} - \frac{\alpha}{K(\alpha - K)} \right\} \right] \quad (3.286)$$

The differential pressure at steady-state is

$$\Delta p_{ss} = x_3(t)_{ss} - \gamma x_1(t)_{ss} \quad (3.287)$$

Substituting Equations 3.285 and 3.286 into Equation 3.287 and simplifying results in

$$\Delta p_{ss} = -\frac{\gamma \bar{f}_N}{\mu K} \quad (3.288)$$

The cable force $(f_c)_{\max}$ for which $\Delta p_{ss} = \Delta p_{\max}$ is obtained by replacing Δp_{ss} with Δp_{\max} and solving for \bar{f}_N resulting in

$$\bar{f}_N = -\frac{\mu K}{\gamma} \Delta p_{\max} \quad (3.289)$$

Replacing \bar{f}_N with $(W - \gamma V) - (f_c)_{\max}$ and solving for $(f_c)_{\max}$,

$$\begin{aligned} (f_c)_{\max} &= (W - \gamma V) + \frac{\mu K}{\gamma} \Delta p_{\max} \\ &= [300 - 62.4(3)] + \frac{6.5(0.2)(4 \times 144)}{62.4} \\ &= 124.8 \text{ lb} \end{aligned} \quad (3.290)$$

in agreement with the response shown in [Figure 3.37](#).

A final observation about the diver model is relevant. The coupling between the second-order differential equation for h in Equation 3.250 and the first-order differential equation for p in Equation 3.253 is one way. That is, the diver's internal pressure p does not affect the depth h , and a second-order state model is suitable if the pressure is not of interest. On the other hand, the depth h influences the diver's internal pressure p , and hence Equation 3.253 cannot be solved independently of Equation 3.250.

EXERCISES

- 3.31 A simple study can be conducted to find the “best” value for T , the integration step size. Since a graph of the analytical solutions for the continuous state variables and the discrete state approximation are in close agreement ([Figure 3.36](#)) when $T = 0.25$ s, we would like to know if a larger value of T can be used without sacrificing significant accuracy. With this in mind, for $T = 0.5, 1, 2, 4, \dots$
- Find the discrete state equations resulting from the use of trapezoidal integration.
 - Solve the resulting discrete state equations for the discrete state vector $\underline{x}_A(n) = [x_{1,A}(n), x_{2,A}(n), x_{3,A}(n)]^T$ and plot the results on the same graph as the continuous response similar to [Figure 3.36](#). Stop when a noticeable difference between $\underline{x}_A(n)$ and $\underline{x}(nT)$ occurs.
- 3.32 Using the baseline conditions for the system parameters unless stated otherwise,
- Find the cable force $(f_c)_{\max}$ to bring up divers (plus gear) weighing 200, 250, 300, 350 and 400 lbs while not exceeding a maximum differential pressure $\Delta P_{\max} = 4$ psi. Enter the results in [Table E3.32](#). Prepare a graph of $(f_c)_{\max}$ vs W . Comment on the results.

TABLE E3.32

<i>W</i> (lbs)	200	250	300	350	400
$(f_c)_{\max}$					
t_s (s)					

- b. In part (a) record the time required for the diver to surface, t_s and enter in [Table E3.32](#). Plot a graph of t_s vs W .
- c. Suppose the volume V of the diver and gear vary with the diver's weight according to $V = 1 + (W/150)$. Repeat Parts (a) and (b).
- 3.33 For the diver with baseline conditions, find the “best” step size T for simulating the diver’s velocity during ascent from 250 ft using
- Explicit Euler integration
 - Improved Euler integration.
 - Modified Euler integration.
- Specify your criterion for determining the “best” step size.
- 3.34 Derive the analytical solutions for the continuous states given in Equations 3.275 through 3.277.
Hint: It may be necessary to defer this problem until after reading [Chapter 4](#), Section 4. 2 on The Laplace Transform.
- 3.35 Using the baseline conditions given for the diver, simulate the response using explicit Euler integration when the constant cable force is 15% below the equilibrium value. Prepare plots of the continuous and discrete states for a duration of 100 s.
- 3.36 It’s been suggested that a sinusoidal cable force $f_c(t) = \bar{F} + A \sin(2\pi t/P)$ is more effective in bringing the diver to the surface safely, i.e. $\Delta p(t) \leq \Delta P_{\max}$, $t \geq 0$ and in less time compared to a constant force. Using the baseline system parameters, choose a numerical integration method to approximate the system dynamics with the suggested type of cable force. That is, experiment with different values of \bar{F} , A , and P and comment on the validity of the claim about using the sinusoidal cable force.
- 3.37 For a diver with system parameters $W = 300$, $K = 0.2$, $\mu = 6.5$, $V = 3$
- Starting with Equation 3.290, plot the inverse relationship, that is Δp_{\max} vs. $(\bar{f}_c)_{\max}$.
 - Simulate several diver ascents from different initial depths using constant cable forces and compare the simulated maximum differential pressure with the values from the graph.
- 3.38 A 250 lb diver with gear weighing another 100 lbs is 400 ft below the surface in equilibrium with his surroundings. A winch cable begins bringing him to the surface using a constant force.
- Using the analytical solutions for the continuous state variables, find the required force needed for the diver to be ascending at a constant rate of 1.5 ft/s ($\dot{h} = -1.5$ ft/s) when he reaches the surface. The remaining parameter values are $K = 0.25$, $\mu = 5$, $V = 3.25$.
 - Simulate the diver ascent using the force determined in part (a) to verify the result. Use Euler integration with step size $T = 0.1$ s.
 - Plot the discrete state variables for the simulation in part (b).
- 3.39 A diver initially in equilibrium at a depth of 350 ft is ascending to the surface under the influence of a constant cable force equal to 10% greater than the equilibrium force. The cable snaps when the diver is 150 ft from the surface. Simulate the diver’s depth, velocity and differential pressure for 120 s. Use any of the numerical integrators presented. System parameters are $W = 325$, $K = 0.23$, $\mu = 4.8$, $V = 3.15$.
- 3.40 A diver initially in equilibrium at a depth of 150 ft is ascending to the surface under the influence of a constant cable force equal to 10% greater than the equilibrium force. The cable snaps when the diver is 50 ft from the surface. Simulate the diver’s depth, velocity, and differential pressure for 120 s. Use any of the numerical integrators presented. System parameters are $W = 325$, $K = 0.23$, $\mu = 4.8$, and $V = 3.15$.

4 Linear Systems Analysis

4.1 INTRODUCTION

Chapter 2 introduced first- and second-order linear time-invariant (LTI) systems in a very superficial way. A general form for the family of step responses, in the absence of input derivative terms, was presented for both types of systems. Alternate representations of LTI systems, namely, simulation diagrams and state-space models, were also discussed.

Chapters 1 and 3 outlined methods for transforming continuous-time differential equation models into discrete-time system models comprising difference equations. In doing so, the grounds were laid for the foundation of continuous-time system simulation.

A natural question that arises is “How accurate is the simulation?” In the case of continuous-time systems with LTI models, it helps to have a solid grasp of how LTI systems respond to elementary inputs such as a step, polynomials, exponentials, and periodic functions. The analytical solutions serve as a benchmark in comparing different simulation (discrete-time system) models.

This chapter begins with a review of the Laplace transformation and its use in finding the free and forced response of continuous-time LTI system models. The counterpart of the Laplace transform for discrete-time systems is the z -transform, and it is covered in the later sections along with examples of how it facilitates the process of finding the response of discrete-time LTI systems. Time and frequency domain characteristics of continuous- and discrete-time LTI system models are discussed. Mappings from the s -plane to the z -plane corresponding to specific numerical integrators are introduced as a quick way of obtaining discrete-time model approximations of continuous-time systems.

4.2 LAPLACE TRANSFORM

The Laplace transform, as the name implies, is a transformation of functions between two domains. The independent variables in the two domains are commonly denoted “ t ” and “ s ” as shown in Figure 4.1, and the domains are referred to as the time domain (or t -domain) and s -domain, respectively.

A class of functions $f(t)$ defined for $t \geq 0$ in the time domain are transformed into functions $F(s)$ in the s -domain according to

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (4.1)$$

Equation 4.1 is the definition of the one-sided Laplace transform of a function $f(t)$. The definition of $f(t)$ for $t < 0$ is irrelevant since the interval of integration in Equation 4.1 is 0 to ∞ . It is valid for functions $f(t)$, which are said to be of exponential order, that is, functions that are bounded by increasing exponentials as $t \rightarrow \infty$, assuring the convergence of the integral in Equation 4.1. This includes all real-world signals as well as certain functions for which $\lim_{t \rightarrow \infty} f(t) = \infty$.

The notation $\mathcal{L}\{f(t)\}$ is interpreted as the Laplace transform of $f(t)$, that is, the function of “ s ” resulting from evaluating the integral in Equation 4.1. The function $f(t)$ and its Laplace transform $F(s)$ are referred to as a Laplace transform pair using the symbol \Leftrightarrow with the function $f(t)$ on one side and its transform $F(s)$ on the other side. To illustrate, consider the unit step function $u(t)$ that equals 1 for $t \geq 0$ and zero for $t < 0$.

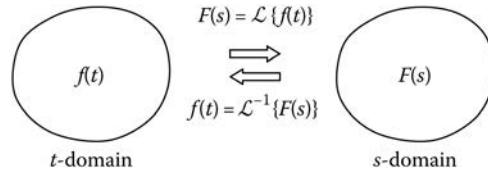


FIGURE 4.1 The Laplace transform $\mathcal{L}\{f(t)\} = F(s)$ and its inverse $f(t) = \mathcal{L}^{-1}\{F(s)\}$.

$$\hat{U}(s) = L\{\hat{u}(t)\} = \int_0^\infty \hat{u}(t) e^{-st} dt = \int_0^\infty 1 e^{-st} dt = \frac{e^{-st}}{-s} \Big|_0^\infty = 0 - \left(\frac{1}{-s} \right) = \frac{1}{s} \quad (4.2)$$

The contribution from the upper limit, $e^{-s(\infty)}$ in Equation 4.2, is zero provided $s > 0$. More specifically, $\text{Re}(s) > 0$ because s is a complex variable $s = \sigma + j\omega$. Therefore,

$$\mathcal{L}\{\hat{u}(t)\} = \frac{1}{s}, \quad \text{Re}(s) > 0 \quad (4.3)$$

indicating the integral in Equation 4.2 converges so long as the complex variable s is located in the right half of the complex plane. Note that the constant function $u(t) = 1$, $-\infty < t < \infty$ is identical to $\hat{u}(t)$ for $t \geq 0$ and consequently has the same Laplace transform.

Henceforth, we shall omit reference to the region of convergence for the integral in Equation 4.1 and simply be concerned with the result. The region of convergence is only of interest when we perform the inverse Laplace transformation using an integration formula to transform $F(s)$ into $f(t)$. Returning to the example of the unit step function, the Laplace transform pair is

$$\hat{u}(t) \Leftrightarrow \frac{1}{s} \quad (4.4)$$

The Laplace transform of other continuous-time functions $f(t)$, $t \geq 0$ is handled in the same manner. For example, the exponential function $f(t) = e^{at}$ has a Laplace transform

$$F(s) = \mathcal{L}\{f(t)\} = \mathcal{L}\{e^{at}\} = \int_0^\infty e^{at} e^{-st} dt = \int_0^\infty e^{-(s-a)t} dt = \frac{1}{s-a} \quad (4.5)$$

Additional time signals of importance are $f(t) = r^n$ ($n = 0, 1, 2, \dots$) along with the trigonometric functions $f(t) = \cos \omega t$ and $f(t) = \sin \omega t$. Applying the definition for the Laplace transform of $f(t)$ in Equation 4.1 produces the results shown in [Table 4.1](#).

4.2.1 PROPERTIES OF THE LAPLACE TRANSFORM

Certain properties of the Laplace transform enable $F(s)$ to be determined without resorting to the definition in Equation 4.1. Several of these properties are presented without proof. The first is the linearity property, which states that the Laplace transform of a linear combination of continuous-time functions is equal to the same linear combination of respective transforms.

P1:

$$\text{Given } \mathcal{L}\{f_1(t)\} = F_1(s) \quad \text{and} \quad \mathcal{L}\{f_2(t)\} = F_2(s)$$

$$\Rightarrow \mathcal{L}\{a_1 f_1(t) + a_2 f_2(t)\} = a_1 \mathcal{L}\{f_1(t)\} + a_2 \mathcal{L}\{f_2(t)\} = a_1 F_1(s) + a_2 F_2(s) \quad (4.6)$$

TABLE 4.1
**Table of Laplace Transform Pairs for
Elementary Continuous-Time Signals**

$f(t)$	$F(s) = L\{f(t)\}$
$\hat{u}(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$	$\frac{1}{s}$
$e^{\pm at}$	$\frac{1}{s \mp a}$
t^n	$\frac{n}{s^{(n+1)}}$
$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$

In properties P2 to P6 that follow, we start with $\mathcal{L}\{f(t)\} = F(s)$.

P2:

$$\mathcal{L}\{e^{\pm at} f(t)\} = F(s \mp a) \quad (4.7)$$

P3:

$$\mathcal{L}\{f(t - t_0) \hat{u}(t - t_0)\} = e^{-t_0 s} F(s) \quad (4.8)$$

P2 and P3 are shifting theorems with P2 applying to a function $f(t)$ multiplied by an exponential time function $e^{\pm at}$. Its Laplace transform $F(s)$ is shifted by an amount “ a ” in the s -domain. P3 applies to functions $f(t)$ delayed t_0 units, that is, shifted an amount t_0 to the right. Note the presence of the delayed unit step function $\hat{u}(t - t_0)$ that zeros out the portion of the signal $f(t)$, $-t_0 \leq t < 0$ shifted from the negative to the positive t -axis. The $\hat{u}(t - t_0)$ can be omitted in P3 if $f(t) = 0$, $t < 0$ (see Figure 4.2).

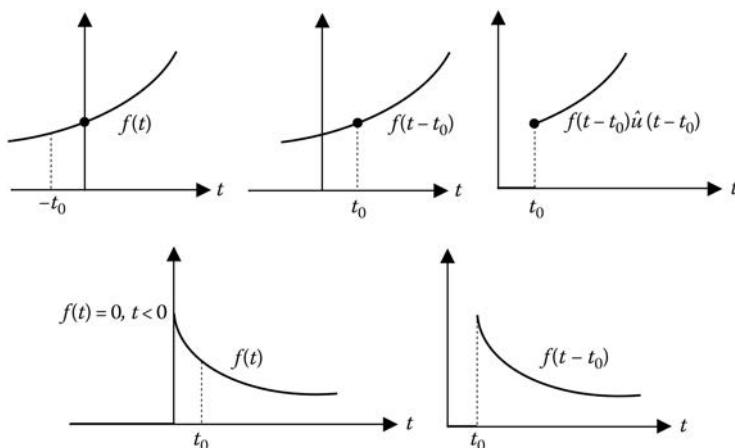


FIGURE 4.2 Illustration of the shifting property P3.

P4 applies to continuous-time functions in the t -domain expressible in product form when one of the factors is t^n .

P4:

$$\mathcal{L}\{t^n f(t)\} = (-1)^n \frac{d^n}{ds^n} F(s) \quad (4.9)$$

P5 shows that integration of functions in the t -domain is equivalent to division by the Laplace variable “ s ” in the s -domain.

P5:

$$\mathcal{L}\left\{\int_0^t f(t') dt'\right\} = \frac{F(s)}{s} \quad (4.10)$$

P6 expresses the Laplace transform of derivatives of $f(t)$ in terms of $F(s)$ and initial conditions. This property is central to solving linear differential equations using algebraic techniques in contrast to the classical time-domain approach.

P6:

$$\mathcal{L}\left\{\frac{d^n}{dt^n} f(t)\right\} = s^n F(s) - s^{n-1} f(0) - s^{n-2} \frac{d}{dt} f(0) - \dots - s \frac{d^{n-2}}{dt^{n-2}} f(0) - \frac{d^{n-1}}{dt^{n-1}} f(0) \quad (4.11)$$

Periodic signals occur frequently as inputs to dynamic systems. The following property applies to functions that are periodic for $t \geq 0$.

P7:

If $f(t)$ is periodic with period T , that is, $f(t - T) = f(t)$, $t \geq 0$

$$F(s) = \frac{1}{1 - e^{-Ts}} \int_0^T e^{-st} f(t) dt \quad (4.12)$$

The convolution of two functions $f(t)$ and $g(t)$ is defined in terms of an integral

$$f(t) * g(t) = \int_0^t f(t - \tau) g(\tau) d\tau \quad (4.13)$$

where $f(t) * g(t)$ denotes the operation of convolving the two continuous-time functions $f(t)$ and $g(t)$. The convolution $g(t) * f(t)$ is equivalent to $f(t) * g(t)$ because a change of variable $\lambda = t - \tau$ in Equation 4.13 leads directly to

$$\int_0^t f(t - \tau) g(\tau) d\tau = \int_0^t g(t - \lambda) f(\lambda) d\lambda = g(t) * f(t) \quad (4.14)$$

It will be shown in the following section that convolution can be used to represent the response of an LTI system to an arbitrary input, and the following property is useful in determining the response of LTI systems.

P8:

$$\mathcal{L}\{f(t)^* g(t)\} = \mathcal{L}\left\{\int_0^t f(t-\tau)g(\tau)d\tau\right\} = F(s)G(s) \quad (4.15)$$

The convolution property P8 is a useful reminder that

$$\mathcal{L}\{f(t)g(t)\} \neq F(s)G(s) \quad (4.16)$$

that is, the Laplace transform of the product of two functions is not equal to the product of the individual Laplace transforms. To illustrate, suppose $g(t)$ is the unit step function.

$$\mathcal{L}\{f(t)g(t)\} = \mathcal{L}\{f(t)\hat{u}(t)\} = \mathcal{L}\{f(t)\} = F(s) \quad (4.17)$$

If Equation 4.16 were an equality, Equation 4.17 would lead to

$$F(s)G(s) = F(s) \quad (4.18)$$

$$\Rightarrow G(s) = 1 \quad (4.19)$$

Equation 4.19 is false and the inequality in Equation 4.16 is correct. Several examples are presented to illustrate the use of these properties.

EXAMPLE 4.1

Find $\mathcal{L}\{4e^{-2t} \sin 3t - 5t \cos 3t\}$.

$$\mathcal{L}\{4e^{-2t} \sin 3t - 5t \cos 3t\} = 4\mathcal{L}\{e^{-2t} \sin 3t\} - 5\mathcal{L}\{t \cos 3t\} \quad (\text{P1}) \quad (4.20)$$

$$\mathcal{L}\{\sin 3t\} = \frac{3}{s^2 + 9} \quad (4.21)$$

$$\Rightarrow \mathcal{L}\{e^{-2t} \sin 3t\} = \frac{3}{s^2 + 9} \Big|_{s \leftarrow s+2} = \frac{3}{(s+2)^2 + 9} \quad (\text{P2}) \quad (4.22)$$

$$\mathcal{L}\{\cos 3t\} = \frac{3}{s^2 + 9} \quad (4.23)$$

$$\Rightarrow \mathcal{L}\{t \cos 3t\} = -\frac{d}{ds}\left(\frac{s}{s^2 + 9}\right) = \frac{s^2 - 9}{(s^2 + 9)^2} \quad (\text{P3}) \quad (4.24)$$

Hence,

$$\mathcal{L}\{4e^{-2t} \sin 3t - 5t \cos 3t\} = 4\left[\frac{3}{(s+2)^2 + 9}\right] - 5\left[\frac{s^2 - 9}{(s^2 + 9)^2}\right]$$

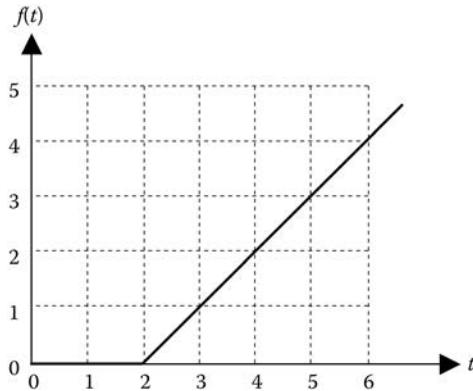


FIGURE 4.3 Graph of $f(t) = (t - 2)\hat{u}(t - 2)$.

EXAMPLE 4.2

$f(t) = (t - 2)\hat{u}(t - 2)$. Graph $f(t)$ and find $F(s)$.

First, the ramp function t , $-\infty < t < \infty$ is delayed (right shifted) two units of time to the right to produce the function $t - 2$, $-\infty < t < \infty$. Second, the shifted function is zeroed out for $t \leq 2$ as a result of the multiplicative term $\hat{u}(t - 2)$. The result is shown in [Figure 4.3](#).

Shifting property P3 is used to find the Laplace transform of $f(t)$.

$$t \Leftrightarrow \frac{1}{s^2} \Rightarrow (t - 2)\hat{u}(t - 2) \Leftrightarrow e^{-2s} \left(\frac{1}{s^2} \right) \quad (4.25)$$

EXAMPLE 4.3

Find the Laplace transform of the signal $f(t)$ shown in [Figure 4.4](#).

The piecewise continuous function $f(t)$ is defined in different intervals by

$$f(t) = \begin{cases} 0, & t < 1 \\ 2(t - 1), & 1 \leq t < 2 \\ 3, & 2 \leq t < 4 \\ -t + 7, & 4 \leq t < 7 \\ 0, & 7 \leq t \end{cases} \quad (4.26)$$

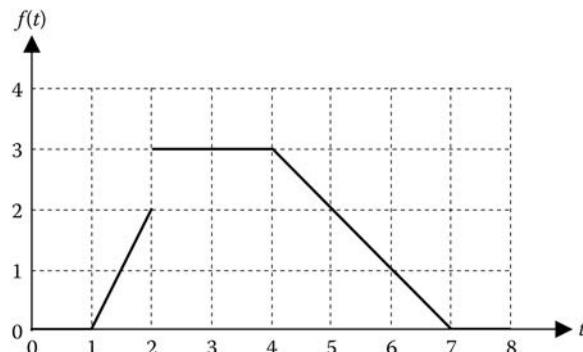


FIGURE 4.4 A piecewise continuous-time function.

Next, we write the function $f(t)$ in a single expression using unit step functions. The procedure is straightforward. The first nonzero term in Equation 4.26 is multiplied by the appropriate step function, so that it “turns on” at the correct time. This gives

$$f(t) = 2(t-1)\hat{u}(t-1) \quad (4.27)$$

that is valid for the first two intervals $t < 1$ and $1 \leq t < 2$. The description of $f(t)$ changes for $t \geq 2$ necessitating a new term that is activated (goes from 0 to 1) for $t \geq 2$. Suppose we write

$$f(t) = 2(t-1)\hat{u}(t-1) + []\hat{u}(t-2) \quad (4.28)$$

The missing term in brackets must subtract out the previous expression for $f(t)$, that is, $2(t-1)$ and add the expression that holds for $2 \leq t < 4$ —in this case the constant 3.

$$f(t) = 2(t-1)\hat{u}(t-1) + [-2(t-1) + 3]\hat{u}(t-2) \quad (4.29)$$

that is correct for $t < 1$, $1 \leq t < 2$, and $2 \leq t < 4$. You should check this yourself by choosing values of t from $-\infty < t < 4$. The same procedure is repeated until the function $f(t)$ is defined as follows:

$$\begin{aligned} f(t) = & 2(t-1)\hat{u}(t-1) + [-2(t-1) + 3]\hat{u}(t-2) + [-3 + (-t+7)]\hat{u}(t-4) \\ & + [-(-t+7) + 0]\hat{u}(t-7) \end{aligned} \quad (4.30)$$

Simplifying Equation 4.30 yields

$$f(t) = 2(t-1)\hat{u}(t-1) + (-2t+5)\hat{u}(t-2) - (t-4)\hat{u}(t-4) + (t-7)\hat{u}(t-7) \quad (4.31)$$

The first, third, and fourth terms in Equation 4.31 have the form $f(t-t_0)\hat{u}(t-t_0)$ and can be Laplace transformed using property P3. The second term in Equation 4.31 can be manipulated into a similar form by doing the following:

$$(-2t+5)\hat{u}(t-2) = [-2(t-2)+1]\hat{u}(t-2) \quad (4.32)$$

$$= -2(t-2)\hat{u}(t-2) \pm \hat{u}(t-2) \quad (4.33)$$

Consequently, $f(t)$ is expressible as

$$f(t) = 2(t-1)\hat{u}(t-1) - 2(t-2)\hat{u}(t-2) + \hat{u}(t-2) - (t-4)\hat{u}(t-4) + (t-7)\hat{u}(t-7) \quad (4.34)$$

Using property P3, the Laplace transform of $f(t)$ in Equation 4.34 is

$$F(s) = 2 \frac{e^{-s}}{s^2} - 2 \frac{e^{-2s}}{s^2} + \frac{e^{-2s}}{s} - \frac{e^{-4s}}{s^2} + \frac{e^{-7s}}{s^2} \quad (4.35)$$

Note that the second term in Equation 4.31, the only one not of the form $f(t-t_0)\hat{u}(t-t_0)$, is present due to the discontinuity in $f(t)$ at $t = 2$.

EXAMPLE 4.4

Find the Laplace transform of the periodic signal $u(t)$ shown in Figure 4.5.

The signal $u(t)$ is periodic for $t \geq 0$ with period $T = 3$. Let $u_1(t)$ represent the first cycle of $u(t)$, that is,

$$u_1(t) = \begin{cases} 2t+1, & 0 \leq t < 1 \\ -2t+5 & 1 \leq t < 2 \\ 1, & 2 \leq t < 3 \\ 0, & 3 \leq t \end{cases} \quad (4.36)$$

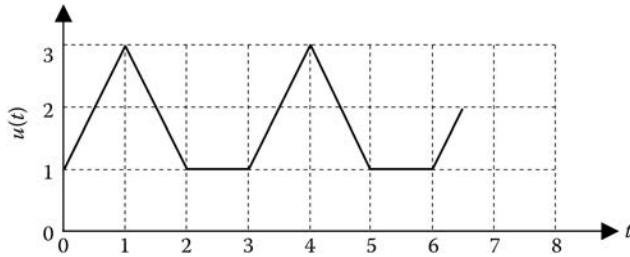


FIGURE 4.5 Graph of signal $u(t)$, periodic for $t \geq 0$.

From property P7,

$$U(s) = \frac{1}{1-e^{-3s}} \int_0^3 e^{-st} u(t) dt \quad (4.37)$$

Since $u(t) = u_1(t)$, $0 \leq t < 3$, Equation 4.37 can be written with $u(t)$ replaced by $u_1(t)$,

$$U(s) = \frac{1}{1-e^{-3s}} \int_0^\infty e^{-st} u_1(t) dt \quad (4.38)$$

$$= \frac{1}{1-e^{-3s}} U_1(s) \quad (4.39)$$

Using the same approach as in Example 4.3, the piecewise continuous function $u_1(t)$ is decomposed into a sum of terms involving step functions.

$$u_1(t) = (2t+1)\hat{u}(t) + [-(2t+1) + (-2t+5)]\hat{u}(t-1) + [-(-2t+5)+1]\hat{u}(t-2) - \hat{u}(t-3) \quad (4.40)$$

$$= 2t\hat{u}(t) + \hat{u}(t) - 4(t-1)\hat{u}(t-1) + 2(t-2)\hat{u}(t-2) - \hat{u}(t-3) \quad (4.41)$$

The shifting property P3 is used repeatedly to obtain the Laplace transform of $u_1(t)$ in Equation 4.41.

$$U_1(s) = \frac{2}{s^2} + \frac{1}{s} - 4 \frac{e^{-s}}{s^2} + 2 \frac{e^{-2s}}{s^2} - \frac{e^{-3s}}{s} \quad (\text{P3}) \quad (4.42)$$

From Equation 4.39, $U(s)$ is therefore

$$U(s) = \frac{1}{1-e^{-3s}} \left[\frac{1-e^{-3s}}{s} + \frac{2-4e^{-s}+2e^{-2s}}{s^2} \right] \quad (4.43)$$

An alternate approach to finding $U(s)$ is to evaluate the integral in Equation 4.37 in pieces, namely

$$U_1(s) = \frac{1}{1-e^{-3s}} \left[\int_0^1 e^{-st} (2t+1) dt + \int_1^2 e^{-st} (-2t+5) dt + \int_2^3 e^{-st} (1) dt \right] \quad (4.44)$$

In general, the second approach is more time-consuming because of the need to evaluate definite integrals.

EXAMPLE 4.5

Find the function $y(t)$ whose Laplace transform is $Y(s) = 1/(s^2(s + 2))$.

$Y(s)$ is expressed as a product of terms,

$$Y(s) = \frac{1}{s^2} \cdot \frac{1}{(s + 2)} = F(s) \cdot G(s) \quad (4.45)$$

where

$$F(s) = \frac{1}{s^2}, \quad G(s) = \frac{1}{(s + 2)} \quad (4.46)$$

From [Table 4.1](#), the functions $f(t)$ and $g(t)$ are

$$f(t) = t, \quad t \geq 0, \quad \text{and} \quad g(t) = e^{-2t}, \quad t \geq 0 \quad (4.47)$$

From the convolution property P8, $y(t)$ is the convolution of $f(t)$ and $g(t)$.

$$y(t) = f * g = \int_0^t f(t - \tau)g(\tau)d\tau = \int_0^t (t - \tau)e^{-2\tau}d\tau \quad (4.48)$$

Evaluating the definite integral in Equation 4.48 gives

$$y(t) = t \int_0^t e^{-2\tau}d\tau - \int_0^t \tau e^{-2\tau}d\tau = \frac{1}{4}(2t - 1 + e^{-2t}), \quad t \geq 0 \quad (4.49)$$

The same result is obtained using the alternate form of the convolution integral,

$$y(t) = g * f = \int_0^t g(t - \tau)f(\tau)d\tau = \int_0^t e^{-2(t-\tau)}\tau d\tau = e^{-2t} \int_0^t \tau e^{2\tau}d\tau \quad (4.50)$$

4.2.2 INVERSE LAPLACE TRANSFORM

The last example required us to find the function $y(t)$ satisfying $\mathcal{L}\{y(t)\} = Y(s)$. In this context, $y(t)$ is referred to as the inverse Laplace transform of $Y(s)$. In general, given a Laplace transform $F(s) = \mathcal{L}\{f(t)\}$, $f(t)$ can be determined using the inverse function $\mathcal{L}^{-1}\{F(s)\}$ (Ogata 1998),

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \int_{z_c-j\infty}^{z_c+j\infty} F(s)e^{st}ds, \quad t > 0 \quad (4.51)$$

where z_c is a real constant chosen to assure convergence of the integral. Finding $f(t)$ in Equation 4.51 requires evaluating an integral of a function of a complex variable over a contour in the s -plane.

Fortunately, there is a simpler approach to performing the inverse Laplace transformation when $F(s)$ involves a ratio of polynomials in s . To illustrate, suppose we are asked to find the inverse Laplace transform of $F(s)$ where

$$F(s) = \frac{1/2}{s^2} - \frac{1/4}{s} + \frac{1/4}{s+2} \quad (4.52)$$

Inverse Laplace transforming both sides of Equation 4.52 and using the linearity property P1, we can write

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \mathcal{L}^{-1}\left\{\frac{1/2}{s^2} - \frac{1/4}{s} + \frac{1/4}{s+2}\right\} \quad (4.53)$$

$$= \mathcal{L}^{-1}\left\{\frac{1/2}{s^2}\right\} - \mathcal{L}^{-1}\left\{\frac{1/4}{s}\right\} + \mathcal{L}^{-1}\left\{\frac{1/4}{s+2}\right\} \quad (4.54)$$

$$= \frac{1}{2} \mathcal{L}^{-1}\left\{\frac{1}{s^2}\right\} - \frac{1}{4} \mathcal{L}^{-1}\left\{\frac{1}{s}\right\} + \frac{1}{4} \mathcal{L}^{-1}\left\{\frac{1}{s+2}\right\} \quad (4.55)$$

From Table 4.1,

$$\mathcal{L}^{-1}\left\{\frac{1}{s^2}\right\} = t, \quad t \geq 0, \quad \mathcal{L}^{-1}\left\{\frac{1}{s}\right\} = 1, \quad t \geq 0, \quad \mathcal{L}^{-1}\left\{\frac{1}{s+2}\right\} = e^{-2t}, \quad t \geq 0 \quad (4.56)$$

$$\Rightarrow f(t) = \frac{1}{2}t - \frac{1}{4} + \frac{1}{4}e^{-2t}, \quad t \geq 0 \quad (4.57)$$

Note $y(t)$ in Equation 4.49 and $f(t)$ in Equation 4.57 are the same functions. Therefore, $Y(s)$ in Equation 4.45 and $F(s)$ in Equation 4.52 are equal, that is,

$$\frac{1}{s^2(s+2)} = \frac{1/2}{s^2} - \frac{1/4}{s} + \frac{1/4}{s+2} \quad (4.58)$$

This suggests that the inverse Laplace transform of a quotient like the one on the left-hand side of Equation 4.58 can be found by first expressing it as a summation of terms and then resorting to a table of Laplace transform pairs. This method is termed partial fraction expansion and will be discussed shortly. First, we establish the need for inverting Laplace transforms expressed in terms of proper fractions with polynomials in “ s ” in the numerator and denominator.

4.2.3 LAPLACE TRANSFORM OF THE SYSTEM RESPONSE

The significance of Laplace transforms in the analysis of LTI dynamic systems is in large part a consequence of property P6, which relates the Laplace transform of $df(t)/dt$ and higher derivatives to $F(s)$, the Laplace transform of $f(t)$. For example, consider a linear second-order system with input $u(t)$ and output $y(t)$ with initial conditions $y(0)$, $\dot{y}(0)$ modeled by

$$\frac{d^2}{dt^2}y(t) + a_1 \frac{d^2}{dt^2}y(t) + a_0y(t) = b_2 \frac{d^2}{dt^2}u(t) + b_1 \frac{d}{dt}u(t) + b_0u(t) \quad (4.59)$$

Laplace transforming both sides of Equation 4.59, with $\mathcal{L}\{u(t)\} = U(s)$ and $\mathcal{L}\{y(t)\} = Y(s)$ results in

$$\begin{aligned} s^2Y(s) - sy(0) - \dot{y}(0) + a_1[sY(s) - y(0)] + a_0Y(s) \\ = b_2[s^2U(s) - su(0) - \dot{u}(0)] + b_1[sU(s) - u(0)] + b_0U(s) \end{aligned} \quad (4.60)$$

where $u(0)$ and $\dot{u}(0)$ are the initial values of the input and its first derivative.

Collecting terms and solving for $Y(s)$ give

$$Y(s) = \underbrace{\left[\left(\frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \right) U(s) - \frac{b_2u(0)s + b_2\dot{u}(0) + b_1u(0)}{s^2 + a_1s + a_0} \right]}_{\text{terms involving input } u(t)} + \underbrace{\frac{y(0)s + \dot{y}(0) + a_1y(0)}{s^2 + a_1s + a_0}}_{\text{terms involving initial state, } [y(0), \dot{y}(0)]^T} \quad (4.61)$$

The complete response $y(t)$ consists of two components. The first,

$$y_{zs}(t) = \mathcal{L}^{-1} \left\{ \left(\frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \right) U(s) - \frac{b_2u(0)s + b_2\dot{u}(0) + b_1u(0)}{s^2 + a_1s + a_0} \right\} \quad (4.62)$$

is called the zero-state response, so named because it represents the system's response when the initial state is zero, that is, $y(0) = \dot{y}(0) = 0$. The terms in Equation 4.62 result from the presence of a forcing function $u(t)$, which explains why the zero-state response is also referred to as the forced response.

The second component is the zero-input response, which is the response of the unforced system, that is, $u(t) = 0$, $t \geq 0$. It is also known as the free response.

$$y_{zi}(t) = \mathcal{L}^{-1} \left\{ \frac{y(0)s + \dot{y}(0) + a_1y(0)}{s^2 + a_1s + a_0} \right\} \quad (4.63)$$

For an elementary type of input $u(t)$, its Laplace transform $U(s)$ will be a ratio of polynomials in s (see [Table 4.1](#)) with the order of the denominator higher than the order of the numerator by at least one. Thus, the terms inside the brackets in Equations 4.62 and 4.63 are also proper fractions with numerator and denominator polynomials in s .

For example, suppose $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$ and $u(t) = \sin \omega t$, $t \geq 0$. Since $u(0) = 0$ and $\dot{u}(0) = \omega$, $Y(s)$ becomes

$$Y(s) = \left[\left(\frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \right) \frac{\omega}{s^2 + \omega^2} - \frac{b_2\omega}{s^2 + a_1s + a_0} \right] + \frac{y_0s + \dot{y}_0 + a_1y_0}{s^2 + a_1s + a_0} \quad (4.64)$$

$$= \frac{y_0s^3 + (a_1y_0 + \dot{y}_0)s^2 + \omega(b_1 + y_0\omega)s + \omega(a_1y_0\omega + \dot{y}_0\omega + b_0 - b_2\omega^2)}{s^4 + a_1s^3 + (a_0 + \omega^2)s^2 + a_1\omega^2s + a_0\omega^2} \quad (4.65)$$

Inverting $Y(s)$ is facilitated by decomposing the right-hand side of Equation 4.65 into a sum of terms for which the inverse Laplace transform is readily determined from tables such as [Table 4.1](#). The same applies for higher order systems with arbitrary inputs.

4.2.4 PARTIAL FRACTION EXPANSION

The second-order system example demonstrates that $Y(s)$ will ordinarily be a proper fraction with polynomials in “ s ” in the numerator and denominator, that is,

$$Y(s) = \frac{N(s)}{D(s)} = \frac{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0}{s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0}, \quad (n > m) \quad (4.66)$$

We begin the process of expanding $Y(s)$ into a sum of terms by determining the roots of $D(s) = 0$. The nature of the n roots will dictate the form of the expansion. A number of cases will be considered.

Case I: All roots of $D(s) = 0$ are real and distinct

Let the n distinct roots be the real numbers p_1, p_2, \dots, p_n obtained by factoring the denominator $D(s)$ into

$$D(s) = (s - p_1)(s - p_2)(s - p_n) \quad (4.67)$$

$$\Rightarrow Y(s) = \frac{N(s)}{D(s)} = \frac{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0}{(s - p_1)(s - p_2) \cdots (s - p_n)} \quad (4.68)$$

p_1, p_2, \dots, p_n are called the poles of $Y(s)$. The partial fraction expansion of $Y(s)$ is

$$Y(s) = \frac{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0}{(s - p_1)(s - p_2) \cdots (s - p_n)} = \frac{c_1}{s - p_1} + \frac{c_2}{s - p_2} + \cdots + \frac{c_n}{s - p_n} \quad (4.69)$$

where the constants c_i , $i = 1, 2, \dots, n$, referred to as the residues of $Y(s)$ at the respective poles p_i , $i = 1, 2, \dots, n$, are obtained from

$$c_i = \left[(s - p_i) \frac{N(s)}{D(s)} \right]_{s=p_i} \quad (4.70)$$

$$= \left[\frac{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0}{(s - p_1)(s - p_2) \cdots (s - p_{i-1})(s - p_{i+1}) \cdots (s - p_n)} \right]_{s=p_i}, \quad i = 1, 2, \dots, n \quad (4.71)$$

EXAMPLE 4.6

Find the inverse Laplace transform of

$$Y(s) = \frac{s^2 + 1}{s^3 + 10.5s^2 + 14s + 4.5} \quad (4.72)$$

Factoring the denominator leads to the partial fraction expansion as follows:

$$Y(s) = \frac{s^2 + 1}{(s + 0.5)(s + 1)(s + 9)} = \frac{c_1}{s + 0.5} + \frac{c_2}{s + 1} + \frac{c_3}{s + 9} \quad (4.73)$$

The constants c_1 , c_2 , and c_3 are obtained from Equation 4.71 as follows:

$$c_1 = \left[(s=0.5) \frac{s^2 + 1}{(s+0.5)(s+1)(s+9)} \right]_{s=-0.5} = \frac{(-0.5)^2 + 1}{(-0.5+1)(-0.5+1)} = \frac{5}{17} \quad (4.74)$$

$$c_2 = \left[(s=1) \frac{s^2 + 1}{(s+0.5)(s+1)(s+9)} \right]_{s=-1} = \frac{(-1)^2 + 1}{(-1+0.5)(-1+9)} = -\frac{1}{2} \quad (4.75)$$

$$c_3 = \left[(s=9) \frac{s^2 + 1}{(s+0.5)(s+1)(s+9)} \right]_{s=-9} = \frac{(-9)^2 + 1}{(-9+0.5)(-9+1)} = \frac{41}{34} \quad (4.76)$$

$$\Rightarrow Y(s) = \frac{5/17}{s+0.5} - \frac{1/2}{s+1} + \frac{41/34}{s+9} \quad (4.77)$$

$y(t)$ is obtained by inverse Laplace transforming the terms in Equation 4.77,

$$y(t) = \mathcal{L}^{-1} \left\{ \frac{5/17}{s+0.5} - \frac{1/2}{s+1} + \frac{41/34}{s+9} \right\} \quad (4.78)$$

$$= \frac{5}{17} e^{-0.5t} - \frac{1}{2} e^{-t} + \frac{41}{34} e^{-9t}, \quad t \geq 0 \quad (4.79)$$

Case II: All roots of $D(s) = 0$ are real and at least one is a multiple root

Suppose p_1 has multiplicity m_1 and p_2 multiplicity m_2 . There are a total of $n - m_1 - m_2 + 2$ distinct pole values, that is, $p_1, p_2, p_3, \dots, p_{n-m_1-m_2+2}$. In factored form, $D(s)$ is

$$D(s) = (s - p_1)^{m_1} (s - p_2)^{m_2} (s - p_3) \cdots (s - p_{n-m_1-m_2+2}) \quad (4.80)$$

The partial fraction expansion of $Y(s)$ is

$$Y(s) = \frac{a_{m_1}}{(s - p_1)^{m_1}} + \frac{a_{m_1-1}}{(s - p_1)^{m_1-1}} + \cdots + \frac{a_1}{(s - p_1)} + \frac{b_{m_2}}{(s - p_2)^{m_2}} + \frac{b_{m_2-1}}{(s - p_2)^{m_2-1}} + \cdots + \frac{b_1}{(s - p_2)} + \frac{c_1}{s - p_3} + \frac{c_2}{s - p_4} + \cdots + \frac{c_{n-m_1-m_2}}{s - p_{n-m_1-m_2+2}} \quad (4.81)$$

Note the number of terms in the expansion corresponding to a particular pole is identical to the order (multiplicity) of the pole. The constants c_i , $i = 1, 2, \dots, n - m_1 - m_2$ are evaluated in the same way as in Case I. For example, c_1 is obtained from

$$c_1 = \left[(s - p_3) \frac{N(s)}{D(s)} \right]_{s=p_3} = \left[\frac{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0}{(s - p_1)^{m_1} (s - p_2)^{m_2} \cdots (s - p_4) (s - p_5) \cdots (s - p_{n-m_1-m_2+2})} \right]_{s=p_3} \quad (4.82)$$

The constants $a_{m_1}, a_{m_1-1}, \dots, a_2, a_1$ are evaluated using

$$a_k = \frac{1}{(m_1 - k)} \frac{d^{m_1-k}}{ds^{m_1-k}} \left[(s - p_i)^{m_1} \frac{N(s)}{D(s)} \right]_{s=p_1}, \quad k = m_1, m_1 - 1, \dots, 2, 1 \quad (4.83)$$

A similar formula applies for the constants $b_{m_2}, b_{m_2-1}, \dots, b_2, b_1$

EXAMPLE 4.7

$$Y(s) = \frac{1}{s^5 + 14s^4 + 75s^3 + 194s^2 + 244s + 120} \quad (4.84)$$

- a. Find the partial fraction expansion of $Y(s)$.
- b. Find $y(t)$.
- a. The poles of $Y(s)$ are found by using a root-solving program such as the MATLAB® function “`roots`” that returns the roots of a polynomial. The call is “`roots(a)`” where “`a`” is the array of coefficients in descending order of the polynomial. With $a = [1 14 75 194 244 120]$, “`roots (a)`” returns $-5, -3, -2, -2, -2$. $Y(s)$ is written with its denominator in factored form and then expanded as follows:

$$Y(s) = \frac{1}{(s+2)^3 + (s+3)(s+5)} = \frac{a_3}{(s+2)^3} + \frac{a_2}{(s+2)^2} + \frac{a_1}{s+2} + \frac{c_1}{s+3} + \frac{c_2}{s+5} \quad (4.85)$$

Evaluating c_1 and c_2 first,

$$c_1 = \left[(s+3) \frac{1}{(s+2)^3(s+3)(s+5)} \right]_{s=-3} = \left[\frac{1}{(-3+2)^3(-3+5)} \right] = -\frac{1}{2} \quad (4.86)$$

$$c_2 = \left[(s+5) \frac{1}{(s+2)^3(s+3)(s+5)} \right]_{s=-5} = \left[\frac{1}{(-5+2)^3(-5+3)} \right] = \frac{1}{54} \quad (4.87)$$

Next, the coefficients a_3 , a_2 , and a_1 are computed from Equation 4.83

$$a_3 = \frac{1}{(3-3)!} \frac{d^{3-3}}{ds^{3-3}} \left[(s+2)^3 \frac{1}{(s+2)^3 + (s+3)(s+5)} \right]_{s=-2} = \left[\frac{1}{(-2+3)(-2+5)} \right] = \frac{1}{3} \quad (4.88)$$

$$a_2 = \frac{1}{(3-2)!} \frac{d^{3-2}}{ds^{3-2}} \left[(s+2)^3 \frac{1}{(s+2)^3 + (s+3)(s+5)} \right]_{s=-2} \quad (4.89)$$

$$= \frac{d}{ds} \left[\frac{1}{(s+3)(s+5)} \right]_{s=-2} = \left[\frac{-1(2s+8)}{(s^2+8s+15^2)} \right]_{s=-2} = -\frac{4}{9} \quad (4.90)$$

$$a_1 = \frac{1}{(3-1)!} \frac{d^{3-1}}{ds^{3-1}} \left[(s+2)^3 \frac{1}{(s+2)^3 + (s+3)(s+5)} \right]_{s=-2} \quad (4.91)$$

$$= \frac{1}{2} \frac{d^2}{ds^2} \left[\frac{1}{(s+3)(s+5)} \right]_{s=-2} = \left[\frac{3s^2+24s+49}{(s^2+8s+15)^3} \right]_{s=-2} = \frac{13}{27} \quad (4.92)$$

An alternative approach to finding the constants c_1 , c_2 , a_3 , a_2 , and a_1 is to use the “`residue`” function in MATLAB that finds the poles of $Y(s)$ and the residues as well.

$Y(s)$ is defined by arrays $n = [1]$, $d = [1 14 75 194 244 120]$, and the statement “[R, P] = residue (n, d)” returns the poles $-5, -3, -2, -2, -2$ in array “P” and the residues $1/54, -1/2, 13/27, -4/9, 1/3$ in array “R.”

- b. From Table 4.1 and property P2, the inverse transform of $Y(s)$ in Equation 4.85 is

$$y(t) = a_3 e^{-2t} \frac{t^2}{2} + a_2 e^{-2t} t + a_1 e^{-2t} + c_1 e^{-3t} + c_2 e^{-5t} \quad (4.93)$$

Substituting the values for c_1 and c_2 from Equations 4.86 and 4.87 as well as a_3 , a_2 , and a_1 from Equations 4.88, 4.90, and 4.92 into Equation 4.93 gives

$$y(t) = \frac{1}{6} e^{-2t} t^2 - \frac{4}{9} e^{-2t} t + \frac{13}{27} e^{-2t} - \frac{1}{2} e^{-3t} + \frac{1}{54} e^{-5t} \quad (4.94)$$

Case III: Complex roots of $D(s) = 0$

When the polynomial $D(s)$ possesses nonrepeated complex roots, it is possible to apply Case I or II and obtain the partial fraction expansion. However, the coefficients will be complex numbers, and the partial fraction expansion will include complex exponentials, which have to be combined in a way to produce real-valued functions. There are two alternatives that eliminate the need for complex number arithmetic. Both are presented followed by an illustrative example.

Suppose the denominator of $Y(s)$ in Equation 4.66 has a single pair of complex roots. Factoring the denominator into a product of linear factors and a quadratic factor,

$$Y(s) = \frac{N(s)}{D(s)} = \frac{a_m s^m + a_{m-1} s^{m-1} + \dots + a_1 s + a_0}{(s - p_1)(s - p_2) \dots (s - p_{n-2})(as^2 + bs + c)} \quad (4.95)$$

where p_1, p_2, \dots, p_{n-2} are real and $as^2 + bs + c = 0$ has complex roots $\alpha \pm j\beta$ ($\beta > 0$). For simplicity, assume the poles p_1, p_2, \dots, p_{n-2} are distinct. The partial fraction expansion is

$$Y(s) = \frac{N(s)}{D(s)} = \frac{c_1}{s - p_1} + \frac{c_2}{s - p_2} + \dots + \frac{c_{n-2}}{s - p_{n-2}} + \frac{d_1 s + d_2}{as^2 + bs + c} \quad (4.96)$$

The constants $c_1, c_2, c_3, \dots, c_{n-2}$ are obtained as before (Case I). The constants d_1 and d_2 are obtained by recombining the terms on the right-hand side of Equation 4.96 and then equating the coefficients of powers of s in the numerator with the coefficients of like powers of s in the original form of the numerator $N(s)$. The inverse Laplace transform of the last term in Equation 4.96 is

$$\mathcal{L}^{-1} \left\{ \frac{d_1 s + d_2}{s^2 + as + b} \right\} = e^{\alpha t} \left[d_1 \cos \beta t \left(\frac{d_1 \alpha + d_2}{\beta} \right) + d_2 \sin \beta t \right] \quad (4.97)$$

To illustrate, consider

$$Y(s) = \frac{s+1}{s^4 + 5s^3 + 11s^2 + 15s} = \frac{s+1}{s(s+3)(s^2 + 2s + 5)} \quad (4.98)$$

$$= \frac{c_1}{s} + \frac{c_2}{s+3} + \frac{d_1 s + d_2}{s^2 + 2s + 5} \quad (4.99)$$

From the quadratic formula, the roots of $s^2 + 2s + 5$ are $-1 \pm j2$. Thus, $\alpha = -1$ and $\beta = 2$. The constants c_1 and c_2 are calculated from

$$c_1 = \left[s \frac{s+1}{s(s+3)(s^2+2s+5)} \right]_{s=0} = \frac{s+1}{(s+3)(s^2+2s+5)} \Big|_{s=0} = \frac{1}{15} \quad (4.100)$$

$$c_2 = \left[(s+3) \frac{s+1}{s(s+3)(s^2+2s+5)} \right]_{s=-3} = \frac{s+1}{s(s^2+2s+5)} \Big|_{s=-3} = \frac{1}{12} \quad (4.101)$$

Combining terms in Equation 4.99 over a common denominator and equating the numerator to $s + 1$, the numerator in Equation 4.98 gives

$$s+1 = \frac{1}{15}(s+3)(s^2+2s+5) + \frac{1}{12}s(s^2+2s+5) + (d_1s+d_2)s(s+3) \quad (4.102)$$

$$\Rightarrow s+1 = \left(\frac{1}{15} + \frac{1}{12} + d_1 \right) s^3 + \left(\frac{1}{3} + \frac{1}{6} + 3d_1 + d_2 \right) s^2 + \left(\frac{11}{15} + \frac{5}{12} + 3d_2 \right) s + 1 \quad (4.103)$$

Equating coefficients of like powers of s on both sides of Equation 4.103,

$$s^3: \quad 0 = \frac{1}{15} + \frac{1}{12} + d_1 \Rightarrow d_1 = -\frac{1}{15} - \frac{1}{12} = -\frac{3}{20}$$

$$s^2: \quad 0 = \frac{1}{3} + \frac{1}{6} + 3d_1 + d_2 \Rightarrow d_2 = -\frac{1}{3} - \frac{1}{6} - 3d_1 = -\frac{1}{2} - 3\left(-\frac{3}{20}\right) = -\frac{1}{20}$$

$$s^1: \quad 1 = \frac{11}{15} + \frac{5}{12} + 3d_2 \Rightarrow d_2 = \frac{1}{3}\left(-\frac{11}{15} - \frac{5}{12}\right) = -\frac{1}{20}$$

$$s^0: \quad 1 = 1$$

Note, only two of the first three equations are needed to solve for d_1 and d_2 , and the remaining two equations serve as a check. Solving for c_1 and c_2 directly in Equations 4.100 and 4.101 eliminates the need to solve four simultaneous equations for the unknown constants c_1 , c_2 , d_1 , and d_2 . Substituting the values for c_1 , c_2 , d_1 , and d_2 into Equation 4.99 yields

$$Y(s) = \frac{1/15}{s} + \frac{1/12}{s+3} + \frac{(-3/20)s - 1/20}{s^2 + 2s + 5} \quad (4.104)$$

$$= \frac{1}{15} \left(\frac{1}{s} \right) + \frac{1}{12} \left(\frac{1}{s+3} \right) - \frac{1}{20} \left(\frac{3s+1}{s^2+2s+5} \right) \quad (4.105)$$

The last term is inverted using Equation 4.97 with $d_1 = 3$, $d_2 = 1$, $\alpha = -1$, and $\beta = 2$.

$$Y(t) = \frac{1}{15} + \frac{1}{12} e^{-3t} - \frac{1}{20} e^{-t} \left[3 \cos 2t + \left(\frac{3(-1)+1}{2} \right) \sin 2t \right] \quad (4.106)$$

$$= \frac{1}{15} + \frac{1}{12} e^{-3t} - \frac{1}{20} e^{-t} (3 \cos 2t - \sin 2t) \quad (4.107)$$

The second method for inverse Laplace transforming terms like the one in Equation 4.97 is based on decomposing it into two terms that can be readily inverted. Starting with an expression

containing a quadratic in the denominator with complex roots, the first step is to complete the square as illustrated in the following equations:

$$F(s) = \frac{d_1 s + d_2}{s^2 + as + b} \quad (4.108)$$

$$= \frac{d_1 s + d_2}{(s^2 + as + a^2/4) + (b - a^2/4)} \quad (4.109)$$

$$= \frac{d_1 s + d_2}{(s + a/2)^2 + \omega^2} \left(\omega^2 = b - \frac{a^2}{4} \right) \quad (4.110)$$

After completing the square in the denominator, Equation 4.110 is expressed as the sum of two terms that are the Laplace transforms of shifted trigonometric functions.

$$F(s) = \frac{d_1[(s + a/2) - a/2] + d_2}{(s + a/2)^2 + \omega^2} \quad (4.111)$$

$$= d_1 \frac{(s + a/2)}{(s + a/2)^2 + \omega^2} + \left[\frac{d_2 - (a/2)d_1}{\omega} \right] \frac{\omega}{(s + a/2)^2 + \omega^2} \quad (4.112)$$

From [Table 4.1](#) and the shifting property P2, $f(t) = \mathcal{L}^{-1}\{F(s)\}$ is

$$f(t) = d_1 e^{-(a/2)t} \cos \omega t + \left[\frac{d_2 - (a/2)d_1}{\omega} \right] e^{-(a/2)t} \sin \omega t \quad (4.113)$$

Returning to the previous example,

$$\frac{1}{20} \left(\frac{3s+1}{s^2 + 2s + 5} \right) = \frac{1}{20} \left(\frac{d_1 s + d_2}{s^2 + as + b} \right) \quad (4.114)$$

making $d_1 = 3$, $d_2 = 1$, $a = 2$, $b = 5$, and $\omega^2 = b - a^2/4 = 4$. Substituting the values for d_1 , d_2 , a , b , and ω into Equation 4.113 leads to the inverse Laplace transform,

$$\frac{1}{20} \left\{ d_1 e^{-(a/2)t} \cos \omega t + \left[\frac{d_2 - (a/2)d_1}{\omega} \right] e^{-(a/2)t} \sin \omega t \right\} = \frac{1}{20} (3 e^{-t} \cos 2t - e^{-t} \sin 2t) \quad (4.115)$$

in agreement with the result shown in Equation 4.107.

Rather than having to remember

$$\mathcal{L}^{-1} \left\{ \frac{d_1 s + d_2}{s^2 + as + b} \right\} = d_1 e^{-(a/2)t} \cos \omega t + \left[\frac{d_2 - (a/2)d_1}{\omega} \right] e^{-(a/2)t} \sin \omega t \quad (4.116)$$

the inverse Laplace transform in the last example can be obtained directly by completing the square, that is,

$$F(s) = \frac{1}{20} \left(\frac{3s+1}{s^2 + 2s + 5} \right) \quad (4.117)$$

$$= \frac{1}{20} \left(\frac{3[s+1]-1+1}{(s+1)^2 + 2^2} \right) \quad (4.118)$$

$$= \frac{1}{20} \left(\frac{3(s+1)-2}{(s+1)^2 + 2^2} \right) \quad (4.119)$$

$$= \frac{1}{20} \left(3 \frac{(s+1)}{(s+1)^2 + 2^2} - \frac{2}{(s+1)^2 + 2^2} \right) \quad (4.120)$$

Inverse Laplace transformation of $F(s)$ gives the same $f(t)$ in Equation 4.115.

EXERCISES

- 4.1 Find the Laplace transforms of the functions $f(t)$ given below. Note that $\hat{u}(t - t_0)$ is the unit step function delayed t_0 units of time.

(a) $t^2 \sin 2t$	(b) $t\hat{u}(t - 1)$	(c) $(t - 1)\hat{u}(t)$	(d) $2[\hat{u}(t - 1) - \hat{u}(t - 4)]$
(e) $(d/dt)(te^{-t})$	(f) $\sin(2t + \pi/4)$	(g) $(1 - 3t)e^{-3t}$	(h) $e^{-3t} \int_0^t \sin 2\tau \cos 2\tau d\tau$
(i) $\sin^2 t$	(j) $te^{-2t} \sin 3t$	(k) $\int_0^t \tau e^{-2\tau} \cos 3(t - \tau) d\tau$	

- 4.2 Find the inverse Laplace transforms of the functions $F(s)$ given in the following:

(a) $\frac{1}{(s^2 - 1)}$	(b) $\frac{1}{(s+2)^2 + 9}$	(c) $\frac{s+1}{(s+2)(s+3)}$	(d) $\frac{s+1}{s(s^2 - 4)^3}$
(e) $\frac{e^{-s} - e^{-3s}}{s(s+1)}$	(f) $\frac{s+1}{s^2 + 1}$	(g) $\frac{2s+1}{(s^2 + s + 1)^2}$	(h) $\frac{s}{(s^2 + 2s + 5)(s^2 + 5s + 6)}$

- 4.3 Find the Laplace transform of the functions $f(t)$ in Figure E4.3a,b. In Figure E4.3b, the function is parabolic over the intervals $0 \leq t < 2$ and $4 \leq t < 6$ and passes through the points $(0, 0)$, $(1, 3)$, $(2, 4)$ and $(4, 4)$, $(5, 3)$, $(6, 0)$.

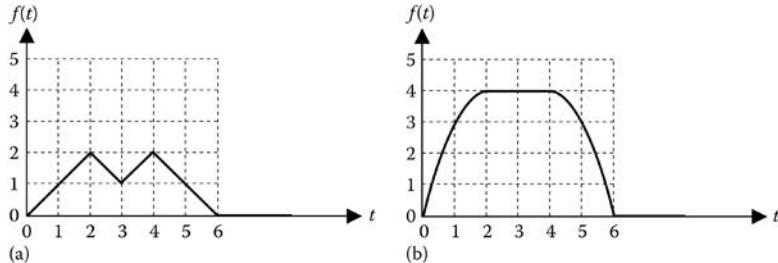


FIGURE E4.3

- 4.4 Graph the function $f(t)$ defined by

$$f(t) = t\hat{u}(t) + (t - 1)\hat{u}(t - 1) - 2t\hat{u}(t - 2) + \hat{u}(t - 3)$$

and find its Laplace transform.

- 4.5 Find the Laplace transform of the periodic function $f(t)$ shown in Figure E4.5:

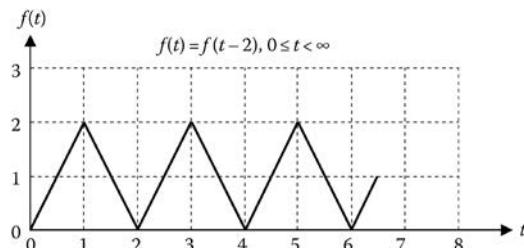


FIGURE E4.5

4.3 TRANSFER FUNCTION

Before we introduce the transfer function, the concept of an impulse function is presented because of its relevance to the response of LTI systems.

4.3.1 IMPULSE FUNCTION

An impulse function at $t = t_0$, denoted $\delta(t - t_0)$, is defined by its property of sifting the value of a function $f(t)$ at t_0 inside an integral, that is,

$$\int_{-\infty}^{\infty} \delta(t - t_0) f(t) dt = f(t_0), \quad -\infty < t_0 < \infty \quad (4.121)$$

The impulse function $\delta(t - t_0)$ is equal to zero wherever $t \neq t_0$ and is not finite at $t = t_0$. No such function exists in a physical sense; however, it can be used to approximate real signals $x(t)$, which occur over a very short duration Δ and satisfy the condition

$$\int_{t_0}^{t_0 + \Delta} x(t) dt = 1 \quad (4.122)$$

as illustrated in [Figure 4.6](#).

From Equation 4.121, the Laplace transform of $\delta(t - t_0)$ is given by

$$\int_0^{\infty} \delta(t - t_0) e^{-st} dt = e^{-st_0}, \quad t_0 > 0 \quad (4.123)$$

When $t_0 = 0$, Equation 4.123 with lower limit 0^- reduces to

$$\mathcal{L}\{\delta(t)\} = 1 \quad (4.124)$$

4.3.2 RELATIONSHIP BETWEEN UNIT STEP FUNCTION AND UNIT IMPULSE FUNCTION

The unit step function $\hat{u}(t)$ that equals 1 for $t \geq 0$ and 0 for $t < 0$ is discontinuous at $t = 0$. Although it cannot be implemented in a physical sense, it serves as an approximation to actual signals, which switch from one level to another in a very short period of time. The first derivative of a unit step function is zero everywhere except at the origin where it fails to exist as a result of the discontinuity. The unit impulse function $\delta(t)$ is likewise zero for all values of t except $t = 0$ where it is infinite.

The unit impulse function $\delta(t)$ can be thought of as the derivative of the unit step function $\hat{u}(t)$. This provides a framework for analyzing systems with discontinuous inputs that result when input

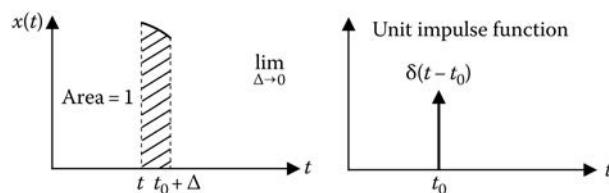


FIGURE 4.6 The unit impulse function $\delta(t - t_0)$ as a limit of a real function $x(t)$.

derivatives are present in the mathematical model (Ogata 1998). To illustrate, consider the first-order system differential equation model

$$\frac{dy}{dt} + 2y = \frac{du}{dt} + u \quad (4.125)$$

where the input $u = \hat{u}(t)$ and the system is initially at rest, that is, $y(0^-) = 0$. Note that the initial time is taken as 0^- to indicate the initial state value prior to application of the step input at $t = 0$. Substituting $\hat{u}(t)$ for u in Equation 4.125 and replacing $(d/dt)\hat{u}(t)$ with $\delta(t)$,

$$\frac{dy}{dt} + 2y = \frac{d}{dt} \hat{u}(t) + \hat{u}(t) = \delta(t) + \hat{u}(t) \quad (4.126)$$

We learned in [Chapter 2](#) that differential equations, where the highest order derivatives of the input and output are identical, possess a direct path between the input and output. We should therefore expect the output $y(t)$ in Equation 4.125 to be discontinuous at $t = 0$, that is, $y(0^+) \neq y(0^-)$ when the input is a unit step $\hat{u}(t)$. The impulse function on the right-hand side of Equation 4.126 is infinite at $t = 0$ accounting for the jump in $y(t)$ over the infinitesimal time period from $t = 0^-$ to $t = 0^+$.

It is possible to demonstrate this behavior without actually solving Equation 4.126 for $y(t)$. Solving for dy/dt in Equation 4.126,

$$\frac{dy}{dt} = \delta(t) + \hat{u}(t) - 2y \quad (4.127)$$

Integrating both sides of Equation 4.127 from 0^- to t ,

$$y(t) = \int_{0^-}^t [\delta(\lambda) + \hat{u}(\lambda) - 2y(\lambda)] d\lambda \quad (4.128)$$

Decomposing the integral in Equation 4.128 into two separate integrals,

$$y(t) = \int_{0^-}^{0^+} [\delta(\lambda) + \hat{u}(\lambda) - 2y(\lambda)] d\lambda + \int_{0^+}^t [\delta(\lambda) + \hat{u}(\lambda) - 2y(\lambda)] d\lambda \quad (4.129)$$

The first integral simplifies because $\hat{u}(t)$ and $y(t)$ are both finite at $t = 0$. The second integral simplifies by virtue of $\delta(t) = 0$ and $\hat{u}(t) = 1$ for $t \geq 0^+$. Equation 4.129 becomes

$$y(t) = \int_{0^-}^{0^+} \delta(\lambda) d\lambda + \int_{0^+}^t [1 - 2y(\lambda)] d\lambda \quad (4.130)$$

From the sifting property of the impulse function, Equation 4.121, the first term on the right-hand side of Equation 4.130 is 1. Evaluating $y(t)$ at $t = 0^+$,

$$y(0^+) = 1 + \int_{0^+}^{0^+} [1 - 2y(\lambda)] d\lambda = 1 \quad (4.131)$$

proving that $y(t)$ is discontinuous at $t = 0$ since $y(0^-) = 0$.

For functions that are discontinuous at the origin, the initial conditions in the differentiation property of Laplace transforms (P6) apply at $t = 0^-$. Hence, for $n = 1$

$$\mathcal{L}\left\{\frac{dy}{dt}\right\} = sY(s) - y(0^-) \quad (4.132)$$

Returning to Equation 4.126, Laplace transformation of both sides yields

$$sY(s) - y(0^-) + 2Y(s) = 1 + \frac{1}{s} \quad (4.133)$$

$$\Rightarrow Y(s) = \frac{s+1}{s(s+2)} + \frac{y(0^-)}{s+2} = \frac{s+1}{s(s+2)} = \frac{1}{2} \left[\frac{1}{s} + \frac{1}{s+2} \right] \quad (4.134)$$

$$\Rightarrow Y(t) = \frac{1}{2}(1 + e^{-2t}), \quad t \geq 0^+ \quad (4.135)$$

Substituting $t = 0^+$ in Equation 4.135 gives $y(0^+) = 1$ in agreement with Equation 4.131.

The initial condition $y(0^+)$ can be obtained by applying the initial value property of Laplace transforms that states

P9:

$$y(0^+) = \lim_{t \rightarrow 0^+} y(t) = \lim_{s \rightarrow \infty} sY(s) \quad (4.136)$$

In this example,

$$y(0^+) = \lim_{s \rightarrow \infty} sY(s) = \lim_{s \rightarrow \infty} s \left[\frac{s+1}{s(s+2)} \right] = 1 \quad (4.137)$$

4.3.3 IMPULSE RESPONSE

The response of LTI systems to an impulse forcing function is of great interest. We shall see why momentarily, but first an example is presented illustrating the process of finding the response of a simple system to an “impulse-like” input and comparing it with the true impulse response of the system.

EXAMPLE 4.8

A spring-mass-damper system is struck by a hammer resulting in a force $f(t)$ like the one shown in Figure 4.7.

- Find and graph the response $x(t)$ for $T = 1, 0.5, 0.1, 0.01$ s.
- Find and graph the impulse response.

- The differential equation model of the system is

$$m\ddot{x} + c\dot{x} + kx = f \Rightarrow \ddot{x} + 2\dot{x} + 4x = \frac{1}{T}[\hat{u}(t) - \hat{u}(t-T)] \quad (4.138)$$

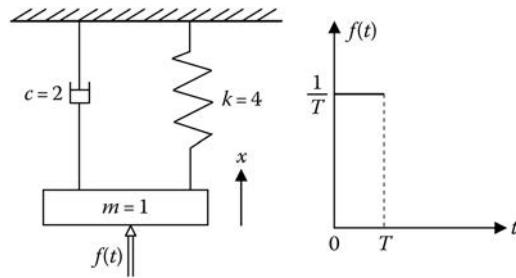


FIGURE 4.7 Mechanical system with pulse input $f(t)$.

Laplace transforming Equation 4.138 with zero initial conditions,

$$(s^2 + 2s + 4)X(s) = \frac{1}{T} \left(\frac{1}{s} - \frac{e^{-Ts}}{s} \right) \quad (4.139)$$

$$X(s) = \frac{1}{T} \left[\frac{1 - e^{-Ts}}{s(s^2 + 2s + 4)} \right] \quad (4.140)$$

Inverse Laplace transformation of Equation 4.140 eventually results in

$$\begin{aligned} x(t) = & \frac{1}{4T} \left[1 - e^{-t} \left(\cos \sqrt{3}t + \frac{1}{\sqrt{3}} \sin \sqrt{3}t \right) \right] \\ & - \frac{1}{4T} \left[1 - e^{-(t-T)} \left(\cos \sqrt{3}(t-T) + \frac{1}{\sqrt{3}} \sin \sqrt{3}(t-T) \right) \right] \hat{u}(t-T) \end{aligned} \quad (4.141)$$

which is simply a linear combination of the step response and a delayed version of the step response. Graphs of Equation 4.141 for $T = 1, 0.5, 0.1, 0.01$ s are generated in the M-file “Ch4_Ex3_1.m” and shown in [Figure 4.8](#).

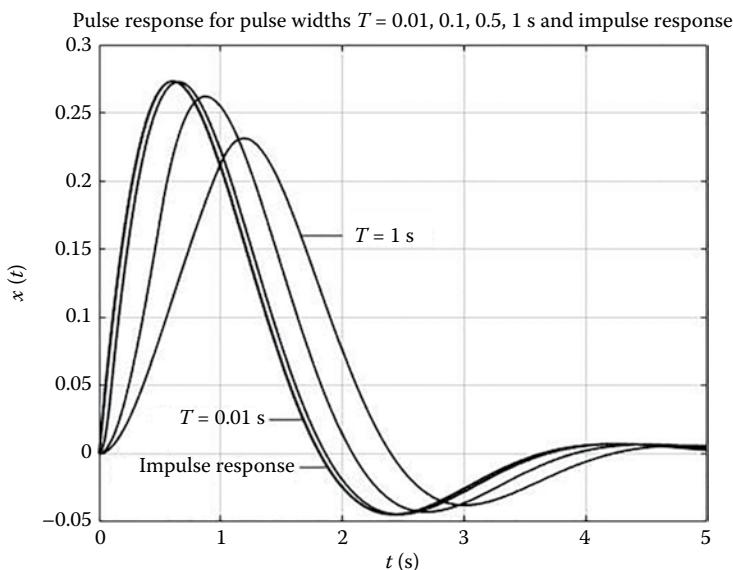


FIGURE 4.8 Pulse response of mechanical system for $T = 1, 0.5, 0.1, 0.01$ s and the impulse response.

b. The true impulse response is obtained by Laplace transforming

$$\ddot{x} + 2\dot{x} + 4x = \delta(t) \quad (4.142)$$

$$\Rightarrow (s^2 + 2s + 4)X(s) = 1 \quad (4.143)$$

Solving for $X(s)$ followed by inverse Laplace transformation results in

$$x_{\text{impulse response}}(t) = \frac{1}{\sqrt{3}} e^{-t} \sin \sqrt{3t}, \quad t \geq 0 \quad (4.144)$$

It is graphed in [Figure 4.8](#) and appears identical to the response $x(t)$ when the pulse width $T = 0.01$ s. Hence, the impulse response provides an accurate approximation of how the mechanical system responds to inputs of short (relative to the time constants of the system's natural modes) duration.

In Section 4.4.2, a mathematical model of a second-order system with input $u(t)$ and output $y(t)$ was introduced as an example of how Laplace transforms can be used to solve for the system response. The second-order LTI system shown in [Figure 4.9](#) is referred to as a single input–single output (SISO) system. The mathematical model is given by

$$\frac{d^2}{dt^2}y(t) + a_1 \frac{d}{dt}y(t) + a_0y(t) = b_2 \frac{d^2}{dt^2}u(t) + b_1 \frac{d}{dt}u(t) + b_0u(t) \quad (4.145)$$

Laplace transforming Equation 4.145, with zero initial conditions for the input, output, and their derivatives, leads to

$$Y(s) = \left(\frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \right) U(s) \quad (4.146)$$

The ratio of $Y(s)$ to $U(s)$, when all initial conditions are identically zero, is called the transfer function of the system. Denoting it by $H(s)$,

$$H(s) = \frac{Y(s)}{U(s)} = \left(\frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \right) \quad (4.147)$$

Consider an n th-order LTI system with transfer function expressible as the ratio of two polynomials in proper fraction form, that is, the denominator polynomial is higher order than the numerator polynomial as in

$$H(s) = \frac{b_ms^m + b_{m-1}s^{m-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} \quad (n > m) \quad (4.148)$$

The transfer function in Equation 4.148 is an alternative to the differential equation model representation of the system dynamics. It offers a convenient way of determining the forced

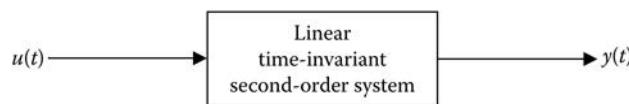


FIGURE 4.9 Second-order system with input $u(t)$ and output $y(t)$.

response of an LTI system. From Equation 4.147, $Y(s)$ is equal to the product of the system transfer function $H(s)$ and the Laplace transform of the input,

$$Y(s) = H(s)U(s) \quad (4.149)$$

and the response is obtained by inverse Laplace transformation $y(t) = \mathcal{L}^{-1}\{Y(s)\}$.

The following examples illustrate the use of the transfer function to obtain the forced response of an LTI system.

EXAMPLE 4.9

A first-order system is governed by the differential equation

$$\frac{dy}{dt} + 2y = u, \quad y(0) = 0 \quad (4.150)$$

- a. Find $H(s)$, the transfer function of the system.
- b. Find $y(t)$, the response when $u(t)$ is (i) $\sin 3t$, $t \geq 0$ and (ii) $\hat{u}(t)$.
- a. From Equation 4.148 with $n = 1$, $m = 0$, $b_0 = 1$, and $a_0 = 2$

$$H(s) = \frac{b_0}{s + a_0} = \frac{1}{s + 2} \quad (4.151)$$

- b. For $u(t) = \sin 3t$, $U(s) = 3/(s^2 + 9)$ and Equation 4.149 becomes

$$Y(s) = H(s)U(s) = \frac{1}{s + 2} \frac{3}{s^2 + 9} \quad (4.152)$$

$$= \frac{3}{13} \left[\frac{1}{s+2} - \frac{s-2}{s^2+9} \right] \quad (4.153)$$

$$= \frac{3}{13} \left[\frac{1}{s+2} - \frac{s}{s^2+9} + \frac{2}{3} \frac{3}{s^2+9} \right] \quad (4.154)$$

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = \frac{3}{13} \left(e^{-2t} - \cos 3t + \frac{2}{3} \sin 3t \right), \quad t \geq 0 \quad (4.155)$$

For $u(t) = \hat{u}(t)$, $U(s) = 1/s$ and Equation 4.149 reduces to

$$Y(s) = H(s)U(s) = \frac{1}{s+2} \cdot \frac{1}{s} = \frac{1}{2} \left[\frac{1}{s} - \frac{1}{s+2} \right] \quad (4.156)$$

$$y(t) = \mathcal{L}^{-1}\{Y(s)\} = \frac{1}{2} (1 - e^{-2t}), \quad t \geq 0 \quad (4.157)$$

EXAMPLE 4.10

For the system with transfer function,

$$H(s) = \frac{s^2 + 3s + 1}{(s+1)(s+3)(s+5)} \quad (4.158)$$

- a. Find the differential equation model of the system.
- b. Find the forced response to the input $u(t) = t$, $t \geq 0$.

a. The differential equation of the system is obtained from $H(s)$ as follows:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{s^2 + 3s + 1}{(s+1)(s+3)(s+5)} = \frac{s^2 + 3s + 1}{s^3 + 9s^2 + 23s + 15} \quad (4.159)$$

$$\Rightarrow (s^3 + 9s^2 + 23s + 15) Y(s) = (s^2 + 3s + 1) U(s) \quad (4.160)$$

$$\Rightarrow s^3 Y(s) + 9s^2 Y(s) + 23s Y(s) + 15Y(s) = s^2 U(s) + 3s U(s) + U(s) \quad (4.161)$$

Performing the inverse Laplace transformation of the individual terms with all initial conditions zero results in the differential equation

$$\frac{d^3}{dt^3} y(t) + 9 \frac{d^2}{dt^2} y(t) + 23 \frac{d}{dt} y(t) + 15y(t) = \frac{d^2}{dt^2} u(t) + 3 \frac{d}{dt} u(t) + u(t) \quad (4.162)$$

b. Substituting $U(s) = 1/s^2$ in Equation 4.149 gives

$$Y(s) = \frac{s^2 + 3s + 1}{s^3 + 9s^2 + 23s + 15} \cdot \frac{1}{s^2} \quad (4.163)$$

The MATLAB statements

```
n = [1 3 1]; d = [1 9 23 15 0 0];
[R, P] = residue(n, d)
```

result in the residues and poles of the partial fraction expansion leading to the following expansion for $Y(s)$,

$$Y(s) = \frac{1}{15} \left(\frac{1}{s^2} \right) + \frac{22}{225} \left(\frac{1}{s} \right) - \frac{1}{8} \left(\frac{1}{s+1} \right) - \frac{1}{36} \left(\frac{1}{s+3} \right) + \frac{11}{200} \left(\frac{1}{s+5} \right) \quad (4.164)$$

and the forced response is obtained by inverse Laplace transformation of $Y(s)$,

$$y(t) = \frac{1}{15} t + \frac{22}{225} - \frac{1}{8} e^{-t} - \frac{1}{36} e^{-3t} + \frac{11}{200} e^{-5t}, \quad t \geq 0 \quad (4.165)$$

4.3.4 RELATIONSHIP BETWEEN IMPULSE RESPONSE AND TRANSFER FUNCTION

The impulse response function and the transfer function of an LTI system are related. Suppose the input to an LTI system is a unit impulse function as illustrated in Figure 4.10.

Since $Y(s) = H(s)U(s) = H(s) \cdot 1 = H(s)$, it follows that

$$Y_{\text{impulse response}}(t) = \mathcal{L}^{-1}\{H(s)\} = h(t) \quad (4.166)$$

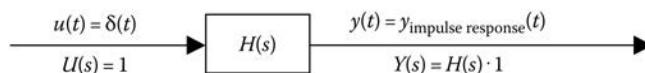


FIGURE 4.10 Linear time-invariant system with unit impulse input.

In other words, the impulse response of an LTI system is simply the inverse Laplace transform of the system transfer function. It is denoted $h(t)$ and referred to as the impulse response function. The impulse response function serves as alternative way of describing the dynamics of an LTI system. It can be used to find the forced response to an arbitrary input by first finding the transfer function $H(s) = \mathcal{L}\{h(t)\}$ and then proceeding in a similar manner to Example 4.10.

Alternatively, the forced response of an LTI system can be obtained directly from

$$y(t) = \mathcal{L}^{-1}\{H(s)U(s)\} = \int_0^t h(t-\tau)u(\tau)d\tau \quad (4.167)$$

that is, by convolution of the impulse response function $h(t)$ and the input $u(t)$. To illustrate, the unit step response of the third-order system in Example 4.10 is obtained using the convolution integral in Equation 4.167.

$$H(s) = \frac{s^2 + 3s + 1}{(s+1)(s+3)(s+5)} = \frac{1}{8} \left[-\frac{1}{s+1} - \frac{2}{s+3} + \frac{11}{s+5} \right] \quad (4.168)$$

$$h(t) = \mathcal{L}^{-1}\{H(s)\} = \frac{1}{8}(-e^{-t} - 2e^{-3t} + 11e^{-5t}) \quad (4.169)$$

$$y(t) = \int_0^t h(\tau)u(t-\tau)d\tau = \int_0^t \frac{1}{8}(-e^{-\tau} - 2e^{-3\tau} + 11e^{-5\tau}) \cdot 1 d\tau \quad (4.170)$$

$$= \frac{1}{8} \left[\left(e^{-\tau} + \frac{2}{3}e^{-3\tau} - \frac{11}{5}e^{-5\tau} \right) \right]_0^t \quad (4.171)$$

$$= \frac{1}{15} + \frac{1}{8}e^{-t} + \frac{1}{12}e^{-3t} - \frac{11}{40}e^{-5t}, \quad t \geq 0 \quad (4.172)$$

The initial condition $y(0^-) = 0$ and from Equation 4.172, $y(0^+) = 0$ as well. The response $y(t)$ is therefore continuous at $t = 0$ despite the discontinuity in the step input. In other words, a direct coupling from the input to the output does not exist. We should expect this result by observing that the third-order differential equation in Equation 4.162 does not contain a term on the right-hand side involving the third derivative of the input. If we express the system model in state variable form, the 1×1 direct coupling matrix D would be zero.

A simulation diagram for an LTI system offers a convenient way of defining the states and revealing whether a direct path (no integrators) exists from the input to the output. [Figure 4.11](#) shows a simulation diagram for the third-order system

$$\frac{d^3}{dt^3}y(t) + a_2 \frac{d^2}{dt^2}y(t) + a_1 \frac{d}{dt}y(t) + a_0y(t) = b_3 \frac{d^3}{dt^3}u(t) + b_2 \frac{d^2}{dt^2}u(t) + b_1 \frac{d}{dt}u(t) + b_0u(t) \quad (4.173)$$

in what is known as observer canonical form (Ogata 1998). This form clearly shows the direct path from the input u to the output y when $b_3 \neq 0$. For the case when $b_3 = 0$, the state x_3 is equal

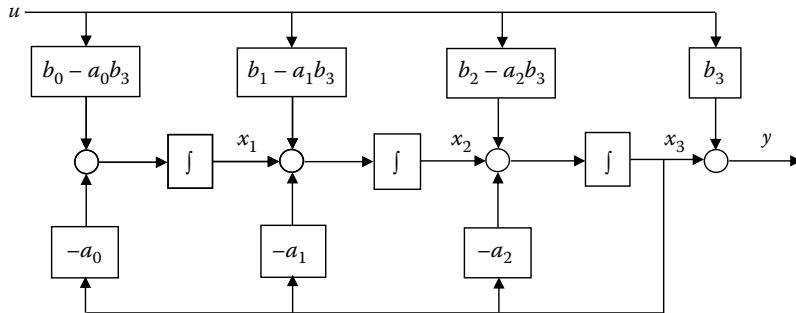


FIGURE 4.11 Simulation diagram of third-order system in observer canonical form.

to the output y and a direct path exists from u to \dot{x}_3 . For a unit step input, the following is true if $b_3 = 0$:

$$y(0^+)y(0^-), \dot{y}(0^+) = \dot{y}(0^-) + (b_2 - a_2b_3)u(0^+) = \dot{y}(0^-) + b_2 \quad (4.174)$$

Consider the third-order system with transfer function given in Equation 4.159 and modeled by the differential equation in Equation 4.162. Comparing Equations 4.162 and 4.173 implies $a_2 = 9$, $a_1 = 23$, and $a_0 = 15$ and $b_3 = 0$, $b_2 = 1$, $b_1 = 3$, and $b_0 = 1$. Assuming zero initial conditions, the first derivative jumps from $\dot{y}(0^-) = 0$ to $\dot{y}(0^+) = \dot{y}(0^-) + b_2 = 1$ at $t = 0$.

Differentiating the solution for the unit step response $y(t)$ in Equation 4.172 gives

$$\frac{dy}{dt} = \frac{1}{40}(-5e^{-t} - 10e^{-3t} + 55e^{-5t}) \quad (4.175)$$

At $t = 0^+$,

$$\frac{dy}{dt}(0^+) = \frac{1}{40}(-5 - 10 + 55) = 1 \quad (4.176)$$

The system transfer function provides a convenient way of finding the forced response of an SISO LTI system. However, finding the transfer function can be a challenge when the mathematical model of the system consists of coupled algebraic and differential equations as opposed to a single nth-order differential equation relating the system input and output. Fortunately, the Laplace transform can be used to reduce the problem of finding the transfer function into one of an algebraic nature. The alternative, namely, eliminating dependent signals and their derivatives in the time domain, is far more cumbersome.

For example, consider the bridged-T network shown in Figure 4.12.

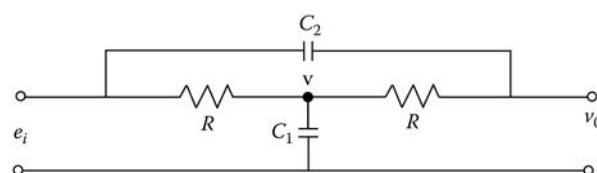


FIGURE 4.12 Circuit with input e_i and output v_0 .

The node voltage method for analyzing the circuit results in the following equations:

$$\frac{e_i - v}{R} = C_1 \frac{dv}{dt} + \frac{v - v_0}{R} \quad (4.177)$$

$$C_2 \frac{d}{dt}(e_i - v_0) + \frac{v - v_0}{R} = 0 \quad (4.178)$$

Rearranging Equations 4.177 and 4.178 with node voltage terms on one side and the input terms on the other gives

$$RC_1 \frac{dv}{dt} + 2v - v_0 = e_i \quad (4.179)$$

$$RC_2 \frac{dv_0}{dt} + v_0 - v = RC_2 \frac{de_i}{dt} \quad (4.180)$$

The node voltage v must be eliminated from Equations 4.179 and 4.180 to arrive at a second-order differential equation involving e_i and v_0 . Laplace transforming both equations with initial conditions set to zero and collecting terms produces the algebraic system of equations

$$\begin{aligned} (RC_1 s + 2)V(s) - V_0(s) &= E_i(s) \\ -V(s) + (RC_2 s + 1)V_0(s) &= RC_2 s E_i(s) \end{aligned} \quad (4.181)$$

Using Cramer's rule, the solution for $V_0(s)$ is

$$V_0(s) = \frac{\begin{vmatrix} RC_1 s + 2 & E_i(s) \\ -1 & RC_2 s E_i(s) \end{vmatrix}}{\begin{vmatrix} RC_1 s + 2 & -1 \\ -1 & RC_2 s + 1 \end{vmatrix}} = \frac{R^2 C_1 C_2 s^2 + 2 R C_2 s + 1}{R^2 C_1 C_2 s^2 + R(C_1 + 2 C_2)s + 1} E_i(s) \quad (4.182)$$

The transfer function is

$$\frac{V_0(s)}{E_i(s)} = \frac{R^2 C_1 C_2 s^2 + 2 R C_2 s + 1}{R^2 C_1 C_2 s^2 + R(C_1 + 2 C_2)s + 1} \quad (4.183)$$

Inverse Laplace transformation of Equation 4.183 leads to the differential equation

$$R^2 C_1 C_2 \frac{d^2 v_0}{dt^2} + R(C_1 + 2 C_2) \frac{dv_0}{dt} + v_0 = R^2 C_1 C_2 \frac{d^2 e_i}{dt^2} + 2 R C_2 \frac{de_i}{dt} + e_i \quad (4.184)$$

4.3.5 SYSTEMS WITH MULTIPLE INPUTS AND OUTPUTS

In general, linear systems (and nonlinear systems) have more than a single input and output. Those systems and their models are designated multiple input–multiple output, abbreviated as MIMO. The transfer function concept still applies.

Suppose, for example, an LTI system such as an electric circuit is driven by independent voltage sources $e_1(t)$ and $e_2(t)$, and signals $i_R(t)$, $v_C(t)$, and $v_{load}(t)$ appearing at various points in the circuit are defined as outputs. A total of six transfer functions exist, one from each of two inputs to each of three outputs. We can write

$$I_R(s) = G_{1,1}(s)E_1(s) + G_{1,2}(s)E_2(s) \quad (4.185)$$

$$V_C(s) = G_{2,1}(s)E_1(s) + G_{2,2}(s)E_2(s) \quad (4.186)$$

$$V_{\text{load}}(s) = G_{3,1}(s)E_1(s) + G_{3,2}(s)E_2(s) \quad (4.187)$$

where

$$G_{1,1}(s) = \left. \frac{I_R(s)}{E_1(s)} \right|_{E_2(s)=0}, \quad G_{1,2}(s) = \left. \frac{I_R(s)}{E_2(s)} \right|_{E_1(s)=0} \quad (4.188)$$

$$G_{2,1}(s) = \left. \frac{V_C(s)}{E_1(s)} \right|_{E_2(s)=0}, \quad G_{2,2}(s) = \left. \frac{V_C(s)}{E_2(s)} \right|_{E_1(s)=0} \quad (4.189)$$

$$G_{3,1}(s) = \left. \frac{V_{\text{load}}(s)}{E_1(s)} \right|_{E_2(s)=0}, \quad G_{3,2}(s) = \left. \frac{V_{\text{load}}(s)}{E_2(s)} \right|_{E_1(s)=0} \quad (4.190)$$

The notation $G_{ij}(s)$ denotes the transfer function from the j th input to the i th output.

Equations 4.185 through 4.187 are a consequence of the principle of superposition that applies to linear systems. Superposition implies that the response of a system to multiple inputs applied simultaneously is equivalent to the sum of the system responses to the individual inputs applied one at a time.

An MIMO system and a method for finding its transfer functions are the focus of the following example.

EXAMPLE 4.11

The amount of solute (drug or metabolite) introduced to or produced in the human body is often assumed to be stored in different compartments of the body. A separate equation for each compartment relates the rate of solute removal to the amount or concentration of the solute in the compartment. The solute can either be transported to another compartment or eliminated from the body by metabolism or excretion. Consider the linear compartment model described in Riggs (1970) for describing the quantities of iodine in humans. The state variables are

- x_1 : Amount of inorganic iodine in the thyroid gland
- x_2 : Amount of organic iodine in the thyroid gland
- x_3 : Amount of hormonal iodine in the extrathyroidal tissue
- x_4 : Amount of iodine in the inorganic iodide compartment

and the inputs are

- q_3 : Rate of entry of exogenous iodide
- q_4 : Rate of entry of exogenous hormonal iodine

The model equations are summarized by the diagram illustrated in [Figure 4.13](#), where k_{12} , k_{13} , k_{21} , k_{24} , k_{31} , k_{43} , k_u , and k_f are the rate constants governing the transfer of iodine between the compartments and its excretion from the body.

The outputs are

- $y_1 = x_1 + x_2 + x_3 + x_4$, total iodine in the body
- $y_2 = k_f x_3 + k_u x_4$, rate of iodine excretion from the body

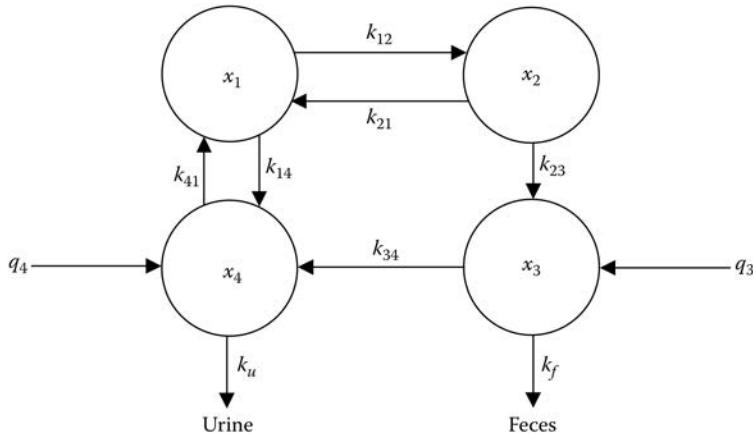


FIGURE 4.13 Compartmental model for iodine distribution in a human.

- Write the state equations for the system and find the matrices A , B , C , and D .
- Draw a block diagram of the system, and label the Laplace transforms of the states x_1 , x_2 , x_3 , and x_4 and outputs y_1 and y_2 .
- Find the transfer function $Y_2(s)/Q_4(s)$.
- Baseline values of the system parameters are

$$k_{12} = 0.8/\text{day}, \quad k_{21} = 0.005/\text{day}, \quad k_{23} = 0.01/\text{day}, \quad \text{and} \quad k_{34} = 0.3/\text{day}$$

$$k_{14} = 0.15/\text{day}, \quad k_{41} = 0.5/\text{day}, \quad k_f = 0.02/\text{day}, \quad \text{and} \quad k_u = 1.2/\text{day}$$

Find the steady-state iodine levels in each compartment in response to a daily intake of iodine, $q_4 = 150 \mu\text{g}/\text{day}$. Assume $q_3 = 0 \mu\text{g}/\text{day}$.

- Find and graph the step response of $x_2(t)$ if the daily intake of iodine drops from 150 (where it has been for a long time) to 50 $\mu\text{g}/\text{day}$.

- From Figure 4.13, the state equations are

$$\left. \begin{aligned} \dot{x}_1 &= -(k_{12} + k_{14})x_1 + k_{21}x_2 + k_{41}x_4 \\ \dot{x}_2 &= k_{12}x_1 - (k_{21} + k_{23})x_2 \\ \dot{x}_3 &= k_{23}x_2 - (k_{34} + k_f)x_3 + q_3 \\ \dot{x}_4 &= k_{14}x_1 + k_{34}x_3 - (k_{14} + k_u)x_4 + q_4 \end{aligned} \right\} \quad (4.191)$$

$$\left. \begin{aligned} y_1 &= x_1 + x_2 + x_3 + x_4 \\ y_2 &= k_f x_3 + k_u x_4 \end{aligned} \right\} \quad (4.192)$$

The matrices A , B , C , and D in $\dot{\underline{x}} = A\underline{x} + B\underline{u}$ and $\underline{y} = C\underline{x} + D\underline{u}$ where $\underline{u} = [q_3 \ q_4]^T$ are

$$A = \begin{bmatrix} -(k_{12} + k_{14}) & k_{21} & 0 & (k_{41}) \\ k_{12} & -(k_{21} + k_{23}) & 0 & 0 \\ 0 & k_{23} & -(k_{34} + k_f) & 0 \\ k_{14} & 0 & k_{34} & -(k_{41} + k_u) \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.193)$$

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & k_f & k_u \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.194)$$

- b. The block diagram is obtained by Laplace transforming the state equations with A , B , C , and D given in Equations 4.193 and 4.194, then solving for $X_1(s)$, $X_2(s)$, $X_3(s)$, and $X_4(s)$ in the respective equations. Introducing the notation $k_1 = k_{12} + k_{14}$, $k_2 = k_{21} + k_{23}$, $k_3 = k_{34} + k_u$ and $k_4 = k_{41} + k_u$ yields

$$X_1(s) = \frac{1}{s + k_1} [k_{21}X_2(s) + k_{41}X_4(s)] \quad (4.195)$$

$$X_2(s) = \left(\frac{k_{12}}{s + k_2} \right) X_1(s) \quad (4.196)$$

$$X_3(s) = \frac{1}{s + k_3} [k_{23}X_2(s) + Q_3(s)] \quad (4.197)$$

$$X_4(s) = \frac{1}{s + k_4} [k_{14}X_1(s) + k_{34}X_3(s) + Q_4(s)] \quad (4.198)$$

The block diagram follows immediately from Equations 4.195 through 4.198 and Equation 4.192. It is shown in Figure 4.14.

- c. The transfer function $Y_2(s)/Q_4(s)$ can be obtained by graphical methods from the block diagram or directly from the model equations. The latter approach is illustrated. Laplace transforming the second output equation in Equation 4.192 followed by division of each term by $Q_4(s)$,

$$Y_2(s) = k_f X_3(s) + k_u X_4(s) \quad (4.199)$$

$$\Rightarrow \frac{Y_2(s)}{Q_4(s)} = k_f \frac{X_3(s)}{Q_4(s)} + k_u \frac{X_4(s)}{Q_4(s)} \quad (4.200)$$

Setting $Q_3(s) = 0$ in Equation 4.197 and solving Equations 4.195 through 4.198 for $X_3(s)$ and $X_4(s)$,

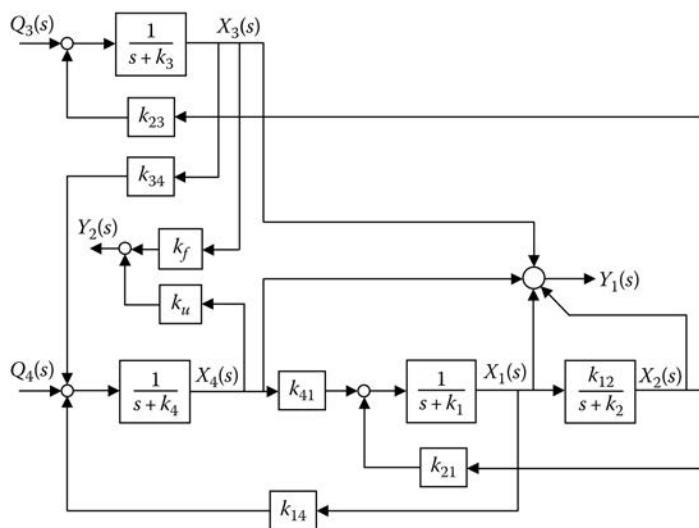


FIGURE 4.14 Block diagram of system modeled by Equations 4.192 and 4.195 through 4.198.

$$X_3(s) = \frac{\begin{vmatrix} s+k_1 & -k_{21} & 0 & -k_{41} \\ -k_{12} & s+k_2 & 0 & 0 \\ 0 & -k_{23} & 0 & 0 \\ -k_{14} & 0 & Q_4(s) & s+k_4 \end{vmatrix}}{\begin{vmatrix} s+k_1 & -k_{21} & 0 & -k_{41} \\ -k_{12} & s+k_{21} & 0 & 0 \\ 0 & -k_{23} & s+k_3 & 0 \\ -k_{14} & 0 & -k_{34} & s+k_4 \end{vmatrix}} \quad (4.201)$$

$$X_4(s) = \frac{\begin{vmatrix} s+k_1 & -k_{21} & 0 & 0 \\ -k_{12} & s+k_2 & 0 & 0 \\ 0 & -k_{23} & s+k_3 & 0 \\ -k_{14} & 0 & -k_{34} & Q_4(s) \end{vmatrix}}{\begin{vmatrix} s+k_1 & -k_{21} & 0 & -k_{41} \\ -k_{12} & s+k_2 & 0 & 0 \\ 0 & -k_{23} & s+k_3 & 0 \\ -k_{14} & 0 & -k_{34} & s+k_4 \end{vmatrix}} \quad (4.202)$$

Evaluation of the determinants in Equations 4.201 and 4.202 is a tedious process left as an exercise problem. The results are as follows:

$$X_3(s) = \left[\frac{\alpha_0}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \right] Q_4(s) \quad (4.203)$$

$$X_4(s) = \left[\frac{s^3 + \beta_2s^2 + \beta_1s + \beta_0}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \right] Q_4(s) \quad (4.204)$$

$$\left. \begin{array}{l} \alpha_0 = k_{12}k_{23}k_{41}, \quad \beta_0 = k_1k_2k_3 - k_{12}k_{21}k_3 \\ \beta_1 = k_1k_2 + k_1k_3 + k_2k_3 - k_{12}k_{21}, \quad \beta_2 = k_1 + k_2 + k_3 \end{array} \right\} \quad (4.205)$$

$$\left. \begin{array}{l} a_0 = k_1k_2k_3k_4 - k_{14}k_{41}k_2k_3 - k_{12}k_{21}k_3k_4 - k_{12}k_{23}k_{34}k_{41} \\ a_1 = k_1k_2k_3 + k_1k_2k_4 + k_1k_3k_4 + k_2k_3k_4 - k_{12}k_{21}(k_3 + k_4) - k_{14}k_{41}(k_2 + k_3) \\ a_2 = k_1k_2 + k_1k_3 + k_1k_4 + k_2k_3 + k_2k_4 + k_3k_4 - k_{12}k_{21} - k_{14}k_{41} \\ a_3 = k_1 + k_2 + k_3 + k_4 \end{array} \right\} \quad (4.206)$$

Combining Equations 4.200, 4.203, and 4.204 produces the desired transfer function,

$$\frac{Y_2(s)}{Q_4(s)} = \frac{k_f\alpha_0 + k_u(s^3 + \beta_2s^2 + \beta_1s + \beta_0)}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \quad (4.207)$$

- d. The steady-state iodine levels in each compartment are obtained from the state equations $\dot{\underline{x}} = A\underline{x} + B\underline{u}$ with $\dot{\underline{x}} = 0$.

$$\underline{x}_{ss} = -A^{-1}B\underline{u}_{ss} \quad \text{where } \underline{u}_{ss} = \begin{bmatrix} (q_3)_{ss} \\ (q_4)_{ss} \end{bmatrix} = \begin{bmatrix} 0 \\ 150 \mu\text{g/day} \end{bmatrix} \quad (4.208)$$

For the given values of the rate constants,

$$x_{ss} = - \begin{bmatrix} -0.95 & 0.005 & 0 & 0.5 \\ 0.8 & -0.015 & 0 & 0 \\ 0 & 0.01 & -0.32 & 0 \\ 0.15 & 0 & 0.3 & -1.7 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 89.6 \\ 4780.9 \\ 149.4 \\ 122.5 \end{bmatrix} \quad (4.209)$$

- e. Using the same method we employed to find $X_3(s)/Q_4(s)$ and $X_4(s)/Q_4(s)$, the transfer function $X_2(s)/Q_4(s)$ is

$$\frac{X_2(s)}{Q_4(s)} = \frac{\gamma_1 s + \gamma_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (4.210)$$

$$\gamma_0 = k_{12} k_{41} k_3, \quad \gamma_1 = k_{12} k_{41} \quad (4.211)$$

Working backward from the transfer function $X_2(s)/Q_4(s)$, the differential equation relating $x_2(t)$ and $q_4(t)$ is

$$\ddot{x}_2 + a_3 \ddot{x}_2 + a_2 \ddot{x}_2 + a_1 \dot{x}_2 + a_0 x_2 = \gamma_1 \dot{q}_4 + \gamma_0 q_4 \quad (4.212)$$

Once the initial conditions are established, Equation 4.212 can be solved to find the complete step response.

Let us assume the input $q_4(t)$ has been constant at 150 µg/day long enough for the system to reach the steady-state levels given in Equation 4.209. It is possible to redefine $t = 0$ as the instant when $q_4(t)$ switches from 150 to 50 µg/day. Figure 4.15 shows the input dropping from $q_4(0^-) = 150$ µg/day to $q_4(0^+) = 50$ µg/day.

With the system at steady-state at $t = 0^-$, the initial conditions are $x_2(0^-) = 4780.9$ µg, $\dot{x}_2(0^-) = \ddot{x}_2(0^-) = \ddot{x}_2(0^-) = 0$. Laplace transforming Equation 4.212,

$$\begin{aligned} s^4 X_2(s) - s^3 x_2(0^-) + a_3[s^3 X_2(s) - s^2 x_2(0^-)] + a_2[s^2 X_2(s) - s x_2(0^-)] \\ + a_1[s X_2(s) - x_2(0^-)] + a_0 X_2(s) = \gamma_1[s Q_4(s) - q_4(0^-)] + \gamma_0 Q_4(s) \end{aligned} \quad (4.213)$$

Solving for $X_2(s)$,

$$X_2(s) = \frac{\gamma_1 s + \gamma_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} Q_4(s) + \frac{x_2(0^-)(s^3 + a_3 s^2 + a_2 s + a_1) - \gamma_1 q_4(0^-)}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (4.214)$$

where

$$Q_4(s) = \mathcal{L}\{q_4(t)\} = \frac{q_4(0^+)}{s} \quad (4.215)$$

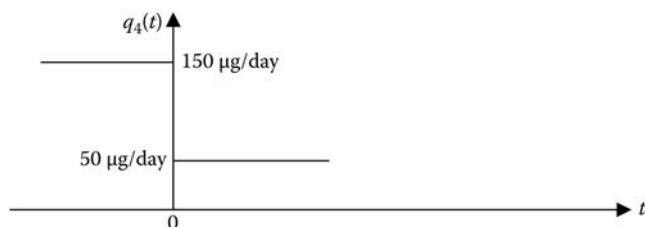


FIGURE 4.15 Step change in input $q_4(t)$.

M-file “*Ch3_Ex3_4.m*” uses the “residue” function to evaluate the partial fraction expansion of each term on the right-hand side of Equation 4.214. The final expression for $X_2(s)$ is of the form

$$X_2(s) = \sum_{i=1}^5 \frac{c_i}{s - p_i} \quad (4.216)$$

where the system poles are $p_1 = -1.7901$, $p_2 = -0.8621$, $p_3 = -0.3248$, and $p_4 = -0.0080$ and the input pole $p_5 = 0$. The residues are $c_1 = 13.6$, $c_2 = -59.1$, $c_3 = 2.4$, $c_4 = 3230.4$, and $c_5 = 1593.6$. The partial fraction expansion of $X_2(s)$ is

$$X_2(s) = \frac{13.6}{s + 1.7901} - \frac{59.1}{s + 0.8621} + \frac{2.4}{s + 0.3248} + \frac{3230.4}{s + 0.0080} + \frac{1593.6}{s} \quad (4.217)$$

Inverting $X_2(s)$ gives

$$x_2(t) = 13.6e^{-1.7901t} - 59.1e^{-0.8621t} + 2.4e^{-0.3248t} + 3230.4e^{-0.0080t} + 1593.6 \quad (4.218)$$

Note that a convenient check of $x_2(t)$ in Equation 4.218 is

$$x_2(0^-) = 13.6 - 59.1 + 2.4 + 3230.4 + 1593.6 = 4780.9$$

which agrees with the initial condition. The step response is shown in Figure 4.16.

The natural modes of the system are $e^{-1.7901t}$, $e^{-0.8621t}$, $e^{-0.3248t}$, and $e^{-0.0080t}$ and the dominant time constant $\tau_{\text{dominant}} = 1/0.008 = 125$ days. It takes approximately $5 \times \tau_{\text{dominant}} = 625$ days for x_2 to attain the new steady-state value of 1593.6 µg.

There is another property of Laplace transforms that is particularly useful when it comes to finding the steady-state response of a system. Known as the Final Value Theorem, it relates the steady state or final value of a signal to its Laplace transform, that is,

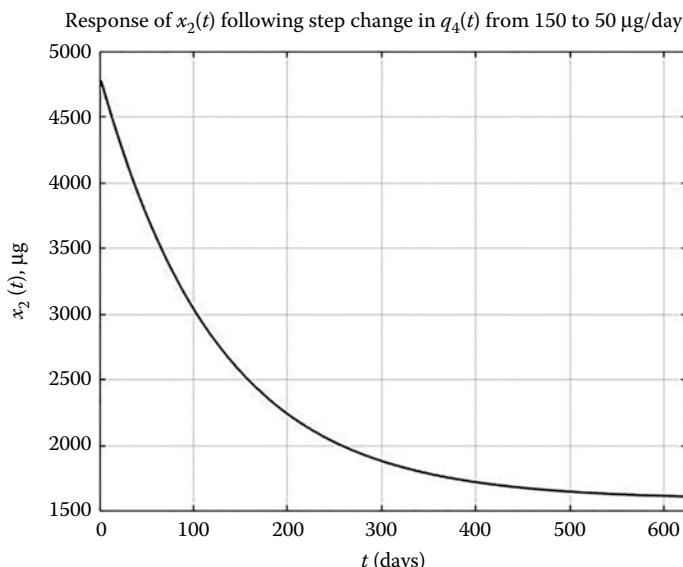


FIGURE 4.16 Step response for $x_2(t)$ following step change in $q_4(t)$ from 150 to 50 µg/day.

P10:

Given $Y(s) = \mathcal{L}\{y(t)\}$, if a final value $y(\infty)$ exists, it is given by

$$y(\infty) = \lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s) \quad (4.219)$$

For a system with transfer function $G(s)$, the steady-state response to a step input of magnitude U_0 is

$$y(\infty) = \lim_{s \rightarrow \infty} sY(s) = \lim_{s \rightarrow 0} sG(s)U(s) = \lim_{s \rightarrow 0} sG(s) \frac{U_0}{s} = G(0)U_0 \quad (4.220)$$

$G(0)$ is referred to as the steady-state gain of the system. The final value property makes it possible to determine the final value $y(\infty)$ from $Y(s)$ without having to find $y(t)$. This is particularly useful when trying to find the steady-state response of a system to a constant input. The input must be constant long enough to allow the transient response to vanish. Practically speaking, this is roughly four to five times the largest effective time constant of the system.

In Example 4.11, the transfer function $Y_1(s)/Q_4(s)$ can be expressed as

$$\frac{Y_1(s)}{Q_4(s)} = \frac{X_1(s) + X_2(s) + X_3(s) + X_4(s)}{Q_4(s)} \quad (4.221)$$

$$= \frac{X_1(s)}{Q_4(s)} + \frac{X_2(s)}{Q_4(s)} + \frac{X_3(s)}{Q_4(s)} + \frac{X_4(s)}{Q_4(s)} \quad (4.222)$$

where the last three terms on the right-hand side are obtained from Equations 4.210, 4.203, and 4.204. The remaining term is left as an exercise. The result is

$$\frac{X_1(s)}{Q_4(s)} = \frac{\delta_2 s^2 + \delta_1 s + \delta_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (4.223)$$

$$\delta_0 = k_{41}k_2k_3, \quad \delta_1 = k_{41}(k_2 + k_3), \quad \delta_2 = k_{41} \quad (4.224)$$

making the transfer function

$$G(s) = \frac{Y_1(s)}{Q_4(s)} = \frac{s^2 + (\beta_2 + \delta_2)s^2 + (\beta_1 + \gamma_1 + \delta_1)s + \alpha_0 + \beta_0 + \gamma_0 + \delta_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (4.225)$$

The final value $y_1(\infty) = x_1(\infty) + x_2(\infty) + x_3(\infty) + x_4(\infty)$ when $q_4(t) = 150$, $t \geq 0$ (same initial input in Example 4.11) is

$$\begin{aligned} y_1(\infty) &= G(0) \cdot 150 = \left(\frac{\alpha_0 + \beta_0 + \gamma_0 + \delta_0}{a_0} \right) \cdot 150 \\ &= \left(\frac{0.004 + 0.00328 + 0.128 + 0.0024}{0.004016} \right) \cdot 150 = 5142.4 \mu\text{g} \end{aligned} \quad (4.226)$$

in agreement with the sum of the components of x_{ss} in Equation 4.209.

A word of caution when applying the final value property. A function $y(t)$ could theoretically grow without bound, that is, $\lim_{t \rightarrow \infty} y(t) = \infty$ or have an undamped oscillatory component, and the final value property will nevertheless produce a finite value. Clearly, the result does not represent a final or steady-state value. We shall investigate the conditions that produce theoretical unbounded outputs of a linear system in a future section.

4.3.6 TRANSFORMATION FROM STATE VARIABLE MODEL TO TRANSFER FUNCTION

The state-space representation offers several advantages over the input–output transfer function method of describing the dynamics of a linear system. For one, it is a more complete representation since the states provide useful information about the internal behavior of the system. Properties of linear systems such as observability and controllability as well as system identification and state feedback are topics normally covered in modern control theory, which rely on state-space models. However, there are times when the transfer function of an SISO system (or transfer functions if the system is MIMO) is required for a system modeled in state variable form.

Consider an MIMO system with inputs u_1, u_2, \dots, u_r and outputs y_1, y_2, \dots, y_m , modeled in state space by

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \quad (4.227)$$

$$\underline{y} = C\underline{x} + D\underline{u} \quad (4.228)$$

where \underline{x} is the n -dimensional state vector $[x_1 \ x_2 \ \dots \ x_n]^T$ and the matrices A, B, C , and D are appropriately dimensioned. Laplace transformation of Equation 4.227 with $\underline{x}(0) = \underline{0}$ gives

$$s\underline{X}(s) = A\underline{X}(s) + B\underline{U}(s) \quad (4.229)$$

$$\Rightarrow \underline{X}(s) = (sI - A)^{-1}B\underline{U}(s) \quad (4.230)$$

Laplace transforming $\underline{y} = C\underline{x} + D\underline{u}$ and substituting $\underline{X}(s)$ from Equation 4.230 gives

$$\underline{Y}(s) = [C(sI - A)^{-1}B + D]\underline{U}(s) \quad (4.231)$$

$$= G(s)\underline{U}(s) \quad (4.232)$$

where $G(s)$, known as the transfer matrix, is a matrix of transfer functions from each of the r inputs to each of the m outputs, that is,

$$G_{ij}(s) = \frac{Y_i(s)}{U_j(s)}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, r \quad (4.233)$$

To illustrate, let us revisit the state variable model for iodine storage in Example 4.11 where the matrices A, B, C , and D are given in Equations 4.193 and 4.194. There are two inputs $u_1(t) = q_3(t)$ and $u_2(t) = q_4(t)$, and outputs $y_1(t)$ and $y_2(t)$ are defined in Equation 4.192. One of the four transfer functions, namely, $Y_1(s)/Q_4(s)$, is given in Equation 4.225. Using the baseline parameter values in Example 4.11 results in

$$\frac{Y_1(s)}{Q_4(s)} = \frac{s^3 + 1.785s^2 + 0.88655s + 0.13768}{s^3 + 2.985s^2 + 2.42855s^2 + 0.52054s + 0.004016} \quad (4.234)$$

The matrix $\Phi(s) = (sI - A)^{-1}$ in Equation 4.231 is computed according to

$$\Phi(s) = (sI - A)^{-1} = \left(sI - \begin{bmatrix} -0.95 & 0.005 & 0 & 0.5 \\ 0.8 & -0.015 & 0 & 0 \\ 0 & 0.01 & -0.32 & 0 \\ 0.15 & 0 & 0.3 & -1.7 \end{bmatrix} \right)^{-1} \quad (4.235)$$

$$= \begin{bmatrix} s+0.95 & -0.005 & 0 & -0.5 \\ -0.8 & s+0.015 & 0 & 0 \\ 0 & -0.01 & s+0.32 & 0 \\ -0.15 & 0 & -0.3 & s+1.7 \end{bmatrix}^{-1} \quad (4.236)$$

$\Phi(s)$ is the Laplace transform of the continuous-time system transition matrix $\Phi(t)$, used to obtain the state response in the time domain. Inverting $(sI - A)$ results in

$$\Phi(s) = \begin{bmatrix} \phi_{11}(s) & \phi_{12}(s) & \phi_{13}(s) & \phi_{14}(s) \\ \phi_{21}(s) & \phi_{22}(s) & \phi_{23}(s) & \phi_{24}(s) \\ \phi_{31}(s) & \phi_{32}(s) & \phi_{33}(s) & \phi_{34}(s) \\ \phi_{41}(s) & \phi_{42}(s) & \phi_{43}(s) & \phi_{44}(s) \end{bmatrix} \quad (4.237)$$

where

$$\phi_{11}(s) = \frac{1}{\Delta(s)} [(s+0.015)(s+0.32)(s+1.7)] \quad (4.238)$$

$$\phi_{12}(s) = \frac{1}{\Delta(s)} [-0.003(s+1.2)] \quad (4.239)$$

$$\phi_{13}(s) = \frac{1}{\Delta(s)} [0.15(s+0.015)] \quad (4.240)$$

$$\phi_{14}(s) = \frac{1}{\Delta(s)} [0.5(s+0.015)(s+0.32)] \quad (4.241)$$

$$\phi_{21}(s) = \frac{1}{\Delta(s)} [0.8(s+0.32)(s+1.7)] \quad (4.242)$$

$$\phi_{22}(s) = \frac{1}{\Delta(s)} [s^3 + 2.97s^2 + 2.388s + 0.4928] \quad (4.243)$$

$$\phi_{23}(s) = \frac{1}{\Delta(s)} [-0.12] \quad (4.244)$$

$$\phi_{24}(s) = \frac{1}{\Delta(s)} [0.4(s+0.32)] \quad (4.245)$$

$$\phi_{31}(s) = \frac{1}{\Delta(s)} [0.008(s+1.7)] \quad (4.246)$$

$$\phi_{32}(s) = \frac{1}{\Delta(s)} [0.01(s^2 + 2.65s + 0.865)] \quad (4.247)$$

$$\phi_{33}(s) = \frac{1}{\Delta(s)} [s^2 + 2.665s^2 + 1.57575s + 0.0163] \quad (4.248)$$

$$\phi_{34}(s) = \frac{1}{\Delta(s)} [0.004] \quad (4.249)$$

$$\phi_{42}(s) = \frac{1}{\Delta(s)} [0.00375(s + 0.824)] \quad (4.250)$$

$$\phi_{43}(s) = \frac{1}{\Delta(s)} [0.3(s^2 + 0.965s + 0.01025)] \quad (4.251)$$

$$\phi_{44}(s) = \frac{1}{\Delta(s)} [s^3 + 1.285s^3 + 0.31905s + 0.00328] \quad (4.252)$$

$$\Delta(s) = |sI - A| = s^4 + 2.985s^3 + 2.42855s^2 + 0.52054s + 0.004016 \quad (4.253)$$

Finally, the transfer function matrix $G(s)$ in Equation 4.232 is given by

$$G(s) = C\Phi(s)B + D \quad (4.254)$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & k_f & k_u \end{bmatrix} \begin{bmatrix} \phi_{11}(s) & \phi_{12}(s) & \phi_{13}(s) & \phi_{14}(s) \\ \phi_{21}(s) & \phi_{22}(s) & \phi_{23}(s) & \phi_{24}(s) \\ \phi_{31}(s) & \phi_{32}(s) & \phi_{33}(s) & \phi_{34}(s) \\ \phi_{41}(s) & \phi_{42}(s) & \phi_{43}(s) & \phi_{44}(s) \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.255)$$

$$= \begin{bmatrix} \phi_{13}(s) + \phi_{23}(s) + \phi_{33}(s) + \phi_{43}(s) & \phi_{14}(s) + \phi_{24}(s) + \phi_{34}(s) + \phi_{44}(s) \\ k_f \phi_{33}(s) + k_u \phi_{43}(s) & k_f \phi_{43}(s) + k_u \phi_{44}(s) \end{bmatrix} \quad (4.256)$$

The component $G_{12}(s)$ in Equation 4.256 is the transfer function $Y_1(s)/Q_4(s)$ previously obtained in Equation 4.234. The reader can verify that the two are identical.

EXERCISES

- 4.6 Show that the step response of a system whose impulse response function $h(t) = 3e^{-2t} + 5\delta(t)$ is discontinuous at $t = 0$.
- 4.7 The differential equation of an LTI system is

$$\frac{d^3y}{dt^3} + 5\frac{d^2y}{dt^2} + 11\frac{dy}{dt} + 15y = 2\frac{d^3u}{dt^3} + u$$

- Find the transfer function $H(s) = Y(s)/U(s)$ of the system.
- Find the impulse response function $h(t)$ for the system.

- c. Find the step response when the initial conditions at $t = 0^-$ are identically zero.
 - d. Find $y(\infty)$ using the final value property, and check your answer with the result obtained in part (c) as $t \rightarrow \infty$.
 - e. Find $y(0^+)$ using the initial value property and check your answer with the result obtained in part (c) as $t \rightarrow 0^+$.
 - f. Find the step response by convolution and compare your answer to the step response found in part (c).
 - g. Draw a simulation diagram for the system in observer canonical form.
 - h. Represent the system in state variable form $\dot{x} = Ax + Bu$, $y = Cx + Du$.
 - i. Find the 1×1 transfer function $G(s) = Y(s)/U(s)$ using Equation 4.254.
- 4.8 Repeat Exercise 4.7 when the system differential equation is
- $\frac{dy}{dt} + 5y = 10u$
 - $\frac{d^2y}{dt^2} + 5\frac{dy}{dt} + 6y = u$
 - $\frac{d^3y}{dt^3} + 5\frac{d^2y}{dt^2} + 11\frac{dy}{dt} + 15y = u$
- 4.9 Use convolution to find the response of the systems with transfer functions
- $H(s) = \frac{s+3}{s^2 + 2s + 1}$
 - $H(s) = \frac{1}{s^2 + 3s + 2}$
 - $H(s) = \frac{s+1}{s^2 + 2s + 2}$
- to the following inputs: (i) $u(t) = \hat{u}(t)$, (ii) $u(t) - \hat{u}(t) - \hat{u}(t-2)$, and (iii) $u(t) = t\hat{u}(t)$.
- 4.10 The circuit in Figure E4.10 is governed by the differential equation

$$\frac{d^2v_0}{dt^2} + \frac{1}{RC} \frac{dv_0}{dt} + \frac{1}{LC} v_0 = \frac{1}{C} \frac{di_g}{dt}$$

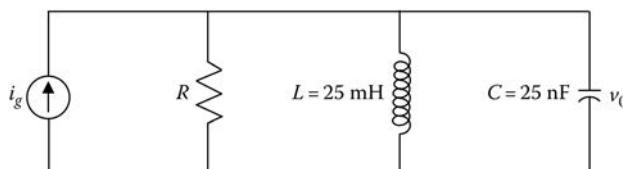
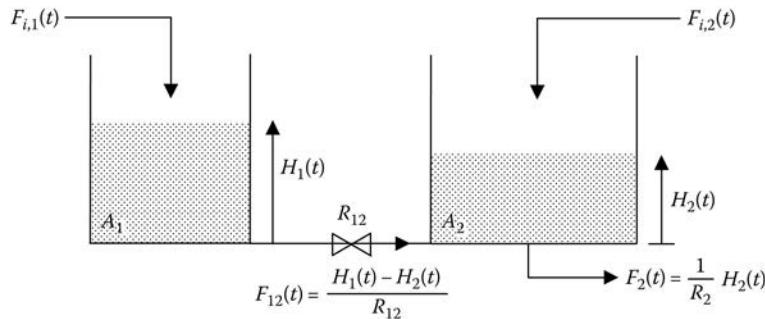


FIGURE E4.10

- Find the impulse response function and plot the results when (a) $R = 400 \Omega$, (b) $R = 500 \Omega$, and (c) $R = 625 \Omega$.
- 4.11 Repeat Example 4.10 with $H(s) = 1/[(s+1)(s+3)(s+5)]$.
- 4.12 Find the transfer function of the bridged-T circuit in Figure 4.12 using equations in the time domain only to find the differential equation of the circuit.
- 4.13 For the system of interacting tanks shown in Figure E4.13:
- Find the transfer functions $\frac{H_2(s)}{F_{i,1}(s)}, \frac{H_1(s)}{F_{i,2}(s)}, \frac{H_1(s)}{F_{i,1}(s)}, \frac{F_2(s)}{F_{i,2}(s)}$
 - Find the differential equation relating $H_1(t)$ and $F_{i,2}(t)$.

**FIGURE E4.13**

4.14 The unit step response of a system is

$$y(t) = 1 + e^{-2t}(\cos 3t + 4 \sin 3t)$$

- a. Find the transfer function of the system.
 - b. Find the impulse response of the system.
 - c. Find the differential equation of the system.
- 4.15 In Example 4.11, find $x_3(\infty)$ and $x_4(\infty)$ when $q_4(t) = 150 \text{ } \mu\text{g/day}$, $t \geq 0$ and $q_3(t) = 0$, $t \geq 0$ using the final value property and the expressions for $X_3(s)$ and $X_4(s)$ in Equations 4.203 and 4.204. Compare your answer with the results in Equation 4.209.
- 4.16 In Example 4.11, find $X_1(s)/Q_4(s)$ and compare your answer with the expression in Equation 4.223.
- 4.17 In Example 4.11,
 - a. Find the transfer functions $Y_1(s)/Q_3(s)$, $Y_2(s)/Q_3(s)$ in a form similar to Equation 4.225.
 - b. Find the step responses for $y_1(t)$ and $y_2(t)$ to inputs $q_4(t) = 50 \text{ } \mu\text{g/day}$, $t \geq 0$ and $q_3(t) = 0$, $t \geq 0$. Assume the initial state is x_{ss} in Equation 4.209.
- 4.18 In Example 4.11, verify that the transfer function $Y_2(s)/Q_4(s)$ in Equation 4.207 is the same as $G_{22}(s)$ in Equation 4.256.

4.4 STABILITY OF LINEAR TIME INVARIANT CONTINUOUS-TIME SYSTEMS

In order for a physical system to operate as intended, it must be capable of generating output(s) in a stable fashion. Regulation of a process temperature is unsatisfactory if the heat source cycles continuously between extremes, that is, off or operating at maximum output, unless it is designed to operate that way like a room thermostat. A control system for maintaining a fixed amount of material in a storage tank in the presence of a fluctuating input may not be performing as intended if the regulating valve in the input line continually cycles between its limits. Each is a real-world example of a control system operating in an unstable manner.

The starting point of an investigation concerning the stability of a system is its mathematical model. The discussion is confined to LTI systems. Excluding nonlinear systems may appear to significantly limit the range of systems considered. However, nonlinear systems can be linearized about specific operating points and stability analyses performed with respect to each operating point. The subject of linearization is treated in [Chapter 7](#).

Consider the second-order system model from the previous section,

$$\frac{d^2}{dt^2} y(t) + a_1 \frac{d}{dt} y(t) + a_0 y(t) = b_2 \frac{d^2}{dt^2} u(t) + b_1 \frac{d}{dt} u(t) + b_0 u(t) \quad (4.257)$$

Applying the differentiation property of the Laplace transform and collecting terms, the Laplace transform of the system output is

$$Y(s) = \left[H(s)U(s) - \frac{b_2u(0^-)s + b_2\dot{u}(0^-) + b_1u(0^-)}{s^2 + a_1s + a_0} \right] + \frac{y(0^-)s + \dot{y}(0^-) + a_1y(0^-)}{s^2 + a_1s + a_0} \quad (4.258)$$

where $H(s)$ is the transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_2s^2 + b_1s + b_0}{s^2 + a_1s + a_0} \quad (4.259)$$

For zero input, $Y(s)$ reduces to the Laplace transform of the free response, that is,

$$Y_{\text{free}}(s) = \frac{y(0)s + \dot{y}(0) + a_1y(0)}{s^2 + a_1s + a_0} \quad (4.260)$$

Note that in the absence of an input, the “ $-$ ” superscript on the initial conditions is no longer necessary. The free response $y_{\text{free}}(t) = \mathcal{L}^{-1}\{Y_{\text{free}}(s)\}$ depends on the roots of the equation $s^2 + a_1s + a_0 = 0$. Denoting the roots as p_1 and p_2 , $y_{\text{free}}(t)$ assumes one of the forms in

$$y_{\text{free}}(t) = \begin{cases} c_1 e^{p_1 t} + c_2 e^{p_2 t}, & p_1, p_2 \text{ real and distinct} \\ e^{\sigma t} [c_1 \cos \omega t + c_2 \sin \omega t], & p_1, p_2 \text{ complex} \\ (c_1 + c_2 t) e^{pt}, & p_1 = p_2 = p \end{cases} \quad (4.261)$$

Constants c_1 and c_2 depend on the initial conditions $y(0)$ and $\dot{y}(0)$. The constants σ , ω , p_1 , p_2 , and p depend on the values of a_0 and a_1 , which are related to the physical parameters of the system. For example, a_0 and a_1 depend on M , B , and K in a mechanical system or R , L , and C for an electrical circuit. The free response in Equation 4.261 is also referred to as the natural response of the system. It consists of a linear combination of the system's natural modes.

4.4.1 CHARACTERISTIC POLYNOMIAL

The denominator of the transfer function $H(s)$ in Equation 4.259 is

$$\Delta(s) = s^2 + a_1s + a_0 = (s - p_1)(s - p_2) \quad (4.262)$$

It is called the characteristic polynomial of the system and $\Delta(s) = 0$ is the characteristic equation. The roots of the characteristic polynomial are referred to as the poles of the system transfer function, and from Equations 4.259 and 4.262, $H(p_1) = H(p_2) = \infty$.

The stability of the system is related to the free response, specifically the limit $L = \lim_{t \rightarrow \infty} y_{\text{free}}(t)$, when one or both initial conditions are nonzero. The following possibilities exist:

1. $L = 0$.
2. $L = \text{constant} \neq 0$.
3. L fails to exist because the free response oscillates with constant amplitude.
4. L fails to exist because the magnitude of the free response approaches infinity.

The system is said to be asymptotically stable in the first case, marginally stable in the second and third cases, and unstable in the last case.

Since the poles p_1 and p_2 dictate the behavior of the free response, they also determine the nature of the system's stability. As a result, we can infer that the stability of the second-order linear system in Equation 4.257 is an inherent system property, that is, it depends on the values of the system parameters and not on the system inputs. The previous statement is entirely general and not restricted to the second-order system under consideration. The different possibilities for the poles of $H(s)$ in Equation 4.259 are illustrated in Figure 4.17.

In (a), (b), (c), (d), and (e), the poles p_1 and p_2 are real and distinct. From Equation 4.261, the free response is the linear combination of natural modes $e^{p_1 t}$ and $e^{p_2 t}$. Since

$$\lim_{t \rightarrow \infty} e^{pt} = \begin{cases} 0, & p < 0 \\ 1, & p = 0 \\ \infty, & p > 0 \end{cases} \quad (4.263)$$

the two natural modes decay to zero in (a), and the limit $L = 0$. Therefore, (a) corresponds to an asymptotically stable system. In (b), one of the natural modes grows monotonically over time and L fails to exist. Hence, (b) represents an unstable system. A similar analysis of the remaining cases (c) through (k) leads to the results shown in Table 4.2.

In summary, the second-order system with transfer function in Equation 4.260 is asymptotically stable provided the two poles are located entirely in the left half of the complex plane. The system is unstable if one or both of its poles lie in the right half of the complex plane or if it has a double pole at the origin. Lastly, it is marginally stable if there is a single pole at the origin and the other pole is negative or there exists a pair of purely imaginary poles located on the imaginary axis. The Routh–Hurwitz stability condition is a simple test for the presence of right-half-plane poles of the transfer function for an nth order LTI system (Dorf and Bishop 2005).

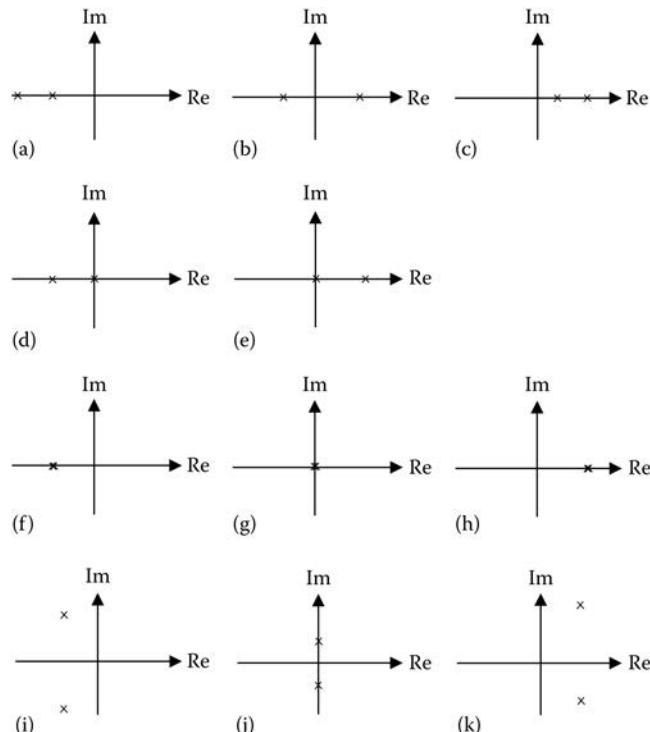


FIGURE 4.17 Possible locations for transfer function poles of a second-order system.

TABLE 4.2
Poles, Natural Modes, and Stability for a Second-Order System

Poles	Natural Modes	System Stability
(a) $p_1 < 0, p_2 < 0$	$e^{p_1 t}, e^{p_2 t}$	Asymptotically stable
(b) $p_1 < 0, p_2 > 0$	$e^{p_1 t}, e^{p_2 t}$	Unstable
(c) $p_1 > 0, p_2 > 0$	$e^{p_1 t}, e^{p_2 t}$	Unstable
(d) $p_1 < 0, p_2 = 0$	$1, e^{p_1 t}$	Marginally stable
(e) $p_1 = 0, p_2 > 0$	$1, e^{p_2 t}$	Unstable
(f) $p_1 = p_2 = p < 0$	e^{pt}, te^{pt}	Asymptotically stable
(g) $p_1 = p_2 = p = 0$	$1, t$	Unstable
(h) $p_1 = p_2 = p > 0$	e^{pt}, te^{pt}	Unstable
(i) $p_1, p_2 = \sigma \pm j\omega (\sigma < 0)$	$e^{\sigma t} \cos \omega t, e^{\sigma t} \sin \omega t$	Asymptotically stable
(j) $p_1, p_2 = \pm j\omega$	$\cos \omega t, \sin \omega t$	Marginally stable
(k) $p_1, p_2 = \sigma \pm j\omega (\sigma > 0)$	$e^{\sigma t} \cos \omega t, e^{\sigma t} \sin \omega t$	Unstable

An alternate definition of asymptotic stability is based on the system's forced response. It states that for a system to be asymptotically stable, its response to any bounded input must remain bounded over time. The same conclusions with respect to the pole locations of an asymptotically stable system shown in Table 4.2 apply to this alternate definition as well.

Systems that are not asymptotically stable according to this definition, that is, bounded input–bounded output (BIBO), are classified as marginally stable or unstable. In the case of a marginally stable system, the forced response to a bounded input may or may not be bounded depending on the input. Consider case (d) in Figure 4.17 where one of the poles is $s = 0$ and the other is located along the negative real axis. In particular, suppose the second pole is $s = -2$ and the second-order system transfer function is

$$H(s) = \frac{s+3}{s(s+2)} \quad (4.264)$$

The forced response to input $u_1(t) = \sin t, t \geq 0$ is obtained as follows:

$$Y_1(s) = H(s)U_1(s) = \frac{s+3}{s(s+2)} \frac{1}{s^2 + 1} = \frac{1.5}{s} - \frac{0.1}{s+2} - \frac{1.4s}{s^2 + 1} - \frac{0.2}{s^2 + 1} \quad (4.265)$$

$$y_1(t) = 1.5 - 0.1e^{-2t} - 1.4 \cos t - 0.2 \sin t, \quad t \geq 0 \quad (4.266)$$

The forced response to input $u_2(t) = 1, t \geq 0$ is obtained in similar fashion.

$$Y_2(s) = H(s)U_2(s) = \frac{s+3}{s(s+2)s} = \frac{1.5}{s^2} - \frac{0.25}{s} + \frac{0.25}{s+2} \quad (4.267)$$

$$y_2(t) = 1.5t - 0.25 + 0.25e^{-2t}, \quad t \geq 0 \quad (4.268)$$

In both instances, the input is a bounded function of time. The output $y_1(t)$ remains bounded while the system response $y_2(t)$ is unbounded as a result of the first term. Careful examination of the system

transfer function in Equation 4.264 reveals that the only bounded inputs capable of producing an unbounded output are those whose Laplace transform contains a pure “ s ” term in the denominator. In other words, the input must either be a constant or a sum of bounded time functions containing a constant.

The forced response of an unstable system to a bounded input is always unbounded due to the presence of an unstable natural mode (see [Table 4.2](#)) which appears in the response. For example, the forced response of a second-order system with a double pole at $s = 0$ (case [g] in [Figure 4.17](#)) to any bounded input contains the unstable mode “ t ” and is always unbounded.

A higher order LTI system is unstable if the transfer function contains one or more right-half-plane poles, the same as for a second-order system. It is not surprising since the characteristic polynomial of an n th-order system can always be factored into a number of linear and quadratic factors with real coefficients. Using partial fraction expansion, the transfer function with factored denominator can be decomposed into a sum of first- and second-order systems. For example, consider the fifth-order system with transfer function given by

$$H(s) = \frac{Y(s)}{U(s)} = \frac{7s^4 + 19s^3 + 45s^2 + 62s + 52}{s^5 + 5s^4 + 12s^3 + 26s^2 + 32s + 24} \quad (4.269)$$

With the help of the MATLAB “residue” function,

$$H(s) = \frac{s}{s^2 + 4} + \frac{s+1}{s^2 + 2s + 2} + \frac{5}{s+3} \quad (4.270)$$

and the output $Y(s) = H(s)U(s)$ of the fifth-order system can be expressed as

$$Y(s) = \frac{s}{s^2 + 4}U(s) + \frac{s+1}{s^2 + 2s + 2}U(s) + \frac{5}{s+3}U(s) \quad (4.271)$$

The system is marginally stable as a result of the complex poles at $s = \pm j2$ located on the imaginary axis. The remaining poles at $s = -1 \pm j$ and $s = -3$ are associated with stable natural modes. The step response of the system with transfer function in Equation 4.269 remains bounded. However, the bounded inputs $u(t) = \sin 2t$ or $u(t) = \cos 2t$ result in an $(s^2 + 4)^2$ term in the denominator of $Y(s)$ and $t \sin 2t$ or $t \cos 2t$ terms in the output $y(t)$. Hence, a bounded step response is necessary but not a sufficient condition for asymptotic stability of LTI systems.

For MIMO systems, the number of transfer functions can grow quickly. However, since stability is an intrinsic property of the system, that is, independent of the system inputs, it is not necessary to investigate each and every transfer function to determine if the system is stable. We shall soon see that the denominator polynomial of each transfer function is identical and, therefore, must be the characteristic polynomial of the system, $\Delta(s)$.

The transfer function matrix $G(s)$ of a MIMO system is the matrix whose ij th element is the transfer function $Y_i(s)/U_j(s)$. From the previous section,

$$G(s) = C(sI - A)^{-1}B + D = C\Phi(s)B + D \quad (4.272)$$

where

A is the $n \times n$ coefficient matrix

B , C , and D are the other matrices in the state variable model description

The inverse of $sI - A$ is $\Phi(s)$, which can be expressed in terms of the adjoint of matrix $sI - A$ and its determinant according to

$$\Phi(s) = (sI - A)^{-1} = \frac{1}{|sI - A|} \text{Adj}(sI - A) \quad (4.273)$$

It follows from Equations 4.272 and 4.273 that every component transfer function of $G(s)$ has the same denominator, that is, the n th-order polynomial

$$|sI - A| = s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1s + a_0 \quad (4.274)$$

Hence, the stability of a linear system described by the state variable model $\dot{\underline{x}} = A\underline{x} + B\underline{u}$, $\underline{y} = C\underline{x} + D\underline{u}$ depends solely on the coefficient matrix A . Furthermore, it is immaterial whether the system is SISO with one transfer function or MIMO with several transfer functions; the coefficient matrix A is all we need to determine whether the system is asymptotically stable, marginally stable, or unstable.

This is consistent with the earlier statement that the stability of the second-order system modeled by the differential equation in Equation 4.258 depends strictly on the constants a_0 and a_1 . After all, the 2×2 coefficient matrix A , while not unique, is determined entirely by a_0 and a_1 . One choice for the states is $x_1 = y$ and $x_2 = \dot{y}$ that leads to

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \quad (4.275)$$

The characteristic polynomial in Equation 4.262 and the n th-order polynomial in Equation 4.274 with $n = 2$ are identical, that is,

$$\Delta(s) = s_2 + a_1s + a_0 = |sI - A| \quad (4.276)$$

A compartment model for iodine storage in humans was presented in Example 4.11. The M-file “Ch4_iodine.m” computes the coefficient matrix

$$A = \begin{bmatrix} -0.95 & 0.005 & 0 & 0.5 \\ 0.8 & -0.015 & 0 & 0 \\ 0 & 0.01 & -0.32 & 0 \\ 0.15 & 0 & 0.3 & -1.7 \end{bmatrix}$$

The characteristic polynomial was given as

$$\Delta(s) = s^4 + 2.985s^3 + 2.42855s^2 + 0.52054s + 0.004016 \quad (4.277)$$

It is left as an exercise (Exercise 4.21) to show that expansion of the determinant $|sI - A|$ produces the characteristic polynomial given in Equation 4.277. The characteristic roots (poles of the system transfer functions) can be obtained by finding the roots of $\Delta(s) = 0$ in Equation 4.277 or equivalently the roots of

$$\Delta(s) = |sI - A| = 0 \quad (4.278)$$

that are also referred to as the eigenvalues of matrix A. The MATLAB functions '`roots[1 2.985 2.43855 0.52054 0.004016]`' and "`eig(A)`" both return the characteristic roots -1.7901 , -0.8621 , -0.3248 , and -0.0080 . Since all the characteristic roots are in the left half of the complex plane, the system is asymptotically stable.

4.4.2 FEEDBACK CONTROL SYSTEM

Real-world processes are nonlinear and may possess one or more equilibrium states. Linear models used to approximate the dynamics in the neighborhood of the equilibrium points are for the most part stable. However, control systems designed to improve some aspect of the system's performance may in fact produce the opposite effect. An example is presented of a stable open-loop system under closed-loop control, which can produce unstable modes in the natural response if the control system parameters are chosen incorrectly.

[Figure 4.18](#) shows a simplified block diagram of a feedback control system for controlling the heading or yaw angle of a small ship. The open-loop system consists of the power converter (motor and gears that control the ship's rudder) modeled by a first-order lag with gain $K_p = 10^\circ/\text{V}$ and time constant $\tau_p = 7.5 \text{ s}$. The ship's yaw dynamics include a gain $K_s = 0.5^\circ/\text{s}^\circ$ (heading)/s $^\circ$ (rudder) and time constant $\tau_s = 7.5 \text{ s}$ resulting in a sluggish response to changes in rudder position. A feedback closed-loop control system is implemented to improve the response. $\theta_{\text{com}}(s)$ and $\theta(s)$ are Laplace transforms of the commanded and actual ship headings, respectively. $E(s)$ is the Laplace transform of the error signal input to the controller.

The closed-loop system transfer function $\theta(s)/\theta_{\text{com}}(s)$ is obtained by eliminating $E(s)$ and $U(s)$ from the following three equations:

$$E(s) = \theta_{\text{com}}(s) - \theta(s) \quad (4.279)$$

$$U(s) = K_C \left(\frac{s+1}{s+10} \right) E(s) \quad (4.280)$$

$$\theta(s) = \left[\frac{0.5}{s(7.5s+1)} \right] \left[\frac{10}{(0.2s+1)} \right] U(s) \quad (4.281)$$

The result is

$$\frac{\theta(s)}{\theta_{\text{com}}(s)} = \frac{5K_C(s+1)}{1.5s^4 + 22.7s^3 + 78s^2 + 5(K_C + 2)s + 5K_C} \quad (4.282)$$

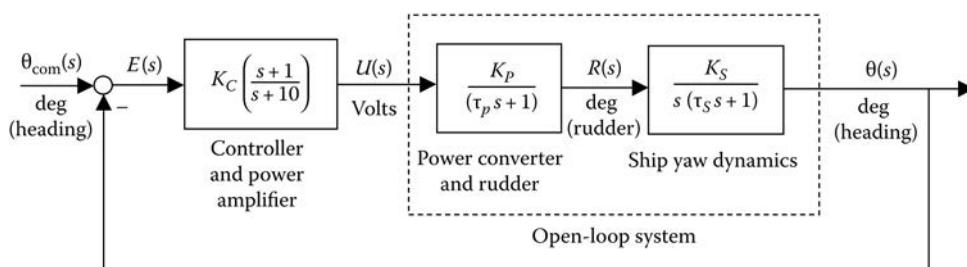


FIGURE 4.18 Block diagram of control system for ship heading.

The characteristic polynomial is

$$\Delta(s) = 1.5s^4 + 22.7s^3 + 78s^2 + 5(K_C + 2)s + 5K_C \quad (4.283)$$

For every value of controller gain K_C , there are four closed-loop system poles, which are the solutions to the characteristic equation, $\Delta(s) = 0$. Root-locus (Dorf and Bishop 2005) is a graphical design method used by control system engineers to plot the poles as the gain parameter K_C varies from 0 to ∞ . There are four branches or loci, each containing one of the poles.

The M-file “Ch4_feedback_yaw.m” produces a root-locus plot shown in Figure 4.19a. When the gain $K_C = 10$, $\Delta(s)$ has two linear factors with real poles at $s = -3.922$ and $s = -10.525$ and a quadratic factor with a pair of complex poles located at $-0.343 \pm j0.831$ (see Figure 4.19b).

The quadratic factor damping ratio, natural frequency, damped natural frequency, and effective time constant are shown in Table 4.3.

The natural response of the closed-loop system ($K_C = 10$) is given by

$$\theta_{\text{nat}}(t) = c_1 e^{-t/0.095} + c_2 e^{-t/0.255} + e^{-t/2.914} [c_3 \cos(0.831t) + c_4 \sin(0.831t)] \quad (4.284)$$

The closed-loop system response, when $K_C = 10$, is faster than the open-loop system as evidenced by the reduction in dominant time constant from 7.5 to 2.914 s.

Suppose the ship is maintaining a heading of 0° (with the rudder angle at 0°) when it becomes necessary to increase the heading by 5° . In the open-loop system, a pulse input to the power converter and rudder subsystem is selected to produce the new desired heading. A pulse is specified rather than a step input because the rudder angle must return to zero once the new heading is achieved. What would happen if a step input were applied? For a pulse input of magnitude A and duration T ,

$$u(t) = A - Au(t - T), \quad t \geq 0 \quad (4.285)$$

the ship's heading is from Equation 4.281

$$\theta_{\text{open-loop}}(s) = \left[\frac{0.5}{s(7.5s + 1)} \right] \left[\frac{10}{(0.2s + 1)} \right] \frac{A(1 - e^{-Ts})}{s} \quad (4.286)$$

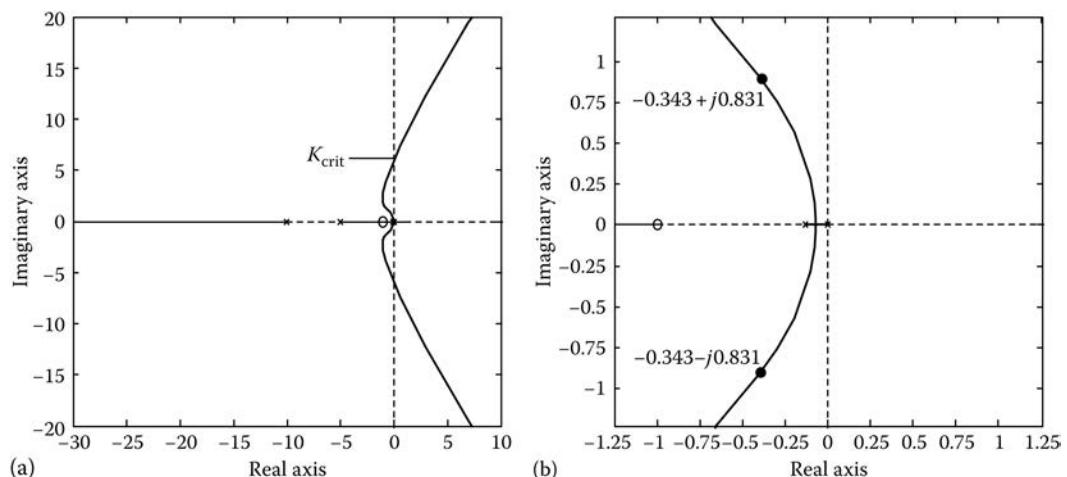


FIGURE 4.19 (a) Root-locus plot. (b) Zoom in near complex poles where $K_C = 10$.

TABLE 4.3
Closed-Loop System Properties ($K_C = 10$)

Characteristic polynomial	$\Delta(s) = 1.5s^4 + 22.7s^3 + 78s^2 + 60s + 50$
Poles	$p_1 = -10.525, p_2 = -3.922, p_3, p_4 = -0.343 \pm j0.831$
Factors	$s^2 + 0.686s + 0.808, s + 10.53, s + 3.92$
Damping ratio	$\zeta = 0.382$
Natural frequency	$\omega_n = 0.899 \text{ rad/s}$
Damped natural frequency	$\omega_d = 0.831 \text{ rad/s}$
Time constants	$\tau_1 = \frac{1}{-p_1} = 0.095 \text{ s}, \tau_2 = \frac{1}{-p_2} = 0.255 \text{ s}, \tau = \frac{1}{\zeta\omega_n} = 2.914 \text{ s}$

The inverse Laplace transform, $\theta_{\text{open-loop}}(t) = \mathcal{L}^{-1}\{\theta_{\text{open-loop}}(s)\}$, is obtained by partial fraction expansion of Equation 4.286 without the $1 - e^{-Ts}$ followed by the shifting property P3 introduced in Section 4.4.2. It is left as an exercise to find $\theta_{\text{open-loop}}(t)$ and show that the final value, that is, new heading, is

$$\theta_{\text{open-loop}}(\infty) = K_p K_s A T = 5AT \quad (4.287)$$

The closed-loop system response with $K_C = 10$ to a command heading of 5° is obtained from Equation 4.282 as

$$\theta_{\text{closed-loop}}(s) = \frac{50(s+1)}{1.5s^4 + 22.7s^3 + 78s^2 + 60s + 50} \cdot \frac{5}{s} \quad (4.288)$$

Using the MATLAB “residue” function to find the residues (partial fraction expansion coefficients) and poles of $\theta_{\text{closed-loop}}(s)$ in Equation 4.288 results in

$$\begin{aligned} R_1 &= -0.2188, R_2 = 1.3934, R_3, R_4 = -3.0873 \mp j0.6270, R_5 = 0 \\ p_1 &= -10.5254, p_2 = -3.9215, p_3, p_4 = -0.3432 \mp j0.8305, p_5 = 0 \end{aligned} \quad (4.289)$$

enabling $\theta_{\text{closed-loop}}(s)$ to be expressed as the sum

$$\theta_{\text{closed-loop}}(s) = \sum_{i=1}^5 \left(\frac{R_i}{s - p_i} \right) \quad (4.290)$$

Invert Laplace transforming Equation 4.290 gives the time domain response

$$\theta_{\text{closed-loop}}(t) = \sum_{i=1}^5 R_i e^{p_i t}, \quad t \geq 0 \quad (4.291)$$

The third and fourth terms involve complex coefficients and complex exponentials,

$$R_3 e^{p_3 t} + R_4 e^{p_4 t} = (-3.087 - j0.627)e^{(-0.343+j0.831)t} + (-3.087 + j0.627)e^{(-0.343-j0.831)t} \quad (4.292)$$

It is inadvisable to express the real-valued closed-loop response $\theta_{\text{closed-loop}}(t)$ in terms of complex exponentials with complex coefficients. However, computing and plotting the response using MATLAB to evaluate the terms in Equation 4.292 produce real numbers because $R_3 e^{p^3 t} + R_4 e^{p^4 t}$ is real-valued for all values of t . In fact, it is easily shown that $\theta_{\text{closed-loop}}(t)$ reduces to the real expression

$$\begin{aligned}\theta_{\text{closed-loop}}(t) = & -0.2188e^{-10.5254t} + 1.3934e^{-3.9215t} \\ & - e^{-0.3432t}[6.175 \cos(0.8305t) - 1.254 \sin(0.8305t)] + 5, \quad t \geq 0\end{aligned}\quad (4.293)$$

The open-loop response with $A = 0.1$, $T = 10$ s and closed-loop response with $K_C = 10$ are plotted in Figure 4.20.

Figure 4.19a shows that the quadratic factor poles migrate to the right-half plane producing a pair of unstable modes when the gain K_C is larger than the critical gain K_{crit} . An approximation of K_{crit} is possible by varying K_C in Equation 4.283 until the MATLAB “roots” function indicates the presence of a pair of imaginary poles located on the imaginary axis. After several attempts at locating the critical gain, the approximate result is $K_C = 166.19$, and the poles of the marginally stable closed-loop system are located at approximately $-14.0705, -0.000011 \pm j6.086566, 1.0627$.

Increasing K_C further produces an unstable system. Figure 4.21 shows the heading response for the closed-loop system with $K_C = 166.19$. Note the sustained oscillations in the marginally stable system. An unstable response corresponding to $K_C = 175$ is also shown in Figure 4.21. The increasing magnitude of oscillations in the unstable system results from a pair of complex poles in the right-half plane at $0.0601 \pm j6.2285$.

Applying the final value property to the closed-loop transfer function in Equation 4.282 gives

$$\theta_{ss} = \lim_{s \rightarrow \infty} s \left\{ \frac{5K_C(s+1)}{1.5s^4 + 22.7s^3 + 78s^2 + 5(K_C + 2)s + 5K_C} \right\} \frac{\theta_{\text{com}}}{s} = \theta_{\text{com}} \quad (4.294)$$

Equation 4.294 holds as long as the control system is asymptotically stable, that is, $K_C < K_{\text{crit}}$.

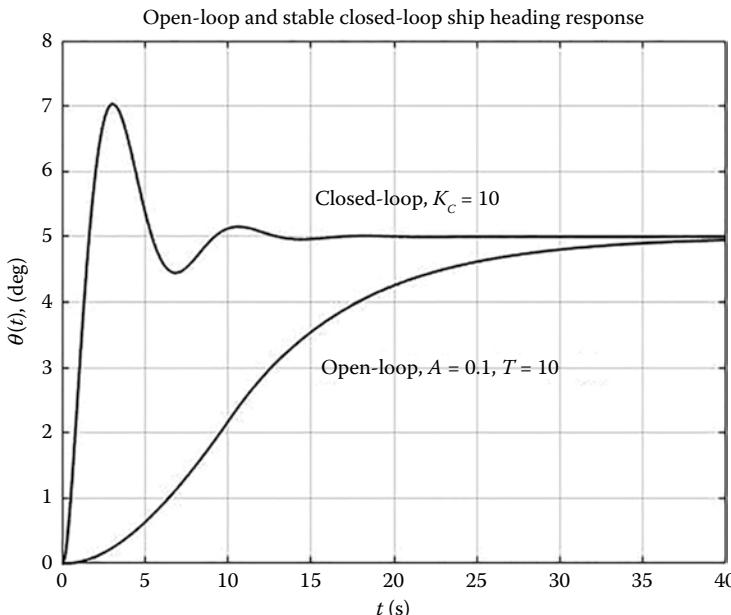


FIGURE 4.20 Ship heading response with open- and closed-loop control.

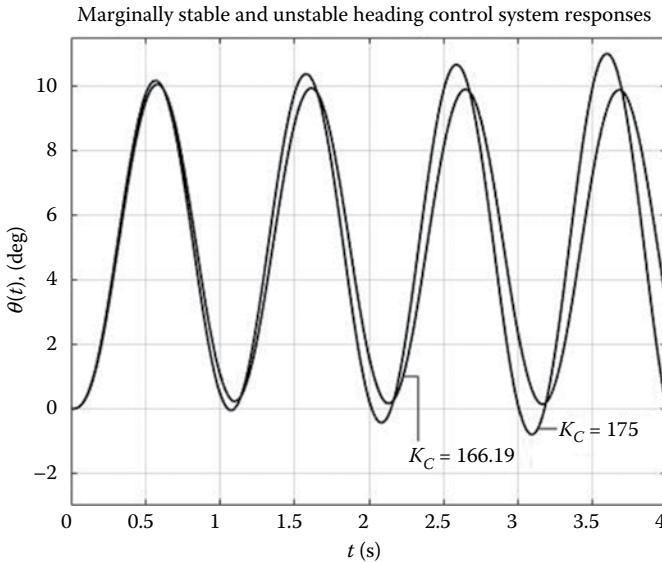


FIGURE 4.21 Heading response for marginally stable and unstable closed-loop system.

The previous example illustrates the concept of stability for an LTI system. The results are predicated on the system response being confined to a range of values for which the linear model is an accurate representation of the actual system's dynamics. Furthermore, limitations on power consumption, component displacements, velocities, etc., must also be satisfied. For example, the design of the ship heading control system using the proportional controller with gain $K_C = 10$ could result in an unrealizable rudder response. A strong argument for simulation is that it allows us to check and monitor such assumptions.

EXERCISES

4.19 For the systems governed by the following differential equations:

- | | |
|--|---|
| (a) $\dot{y} = u$ (an integrator) | (b) $\ddot{y} = u$ (a double integrator) |
| (c) $\dot{y} + 2y = u$ | (d) $\dot{y} - 2y = u$ |
| (e) $\dot{y} + 1.5\dot{y} + 0.5y = u$ | (f) $\ddot{y} + 4y = u$ |
| (g) $\ddot{y} - 9y = u$ | (h) $\dddot{y} + 4\ddot{y} + 6\dot{y} + 5y = u$ |
| (i) $\ddot{y} + 2.5\ddot{y} + 2\dot{y} + 2.5y = u$ | |

determine whether the system is asymptotically stable, marginally stable, or unstable, and find the natural response, that is, a linear combination of the natural modes.

4.20 Find the characteristic polynomial and characteristic roots of the system with state equation

a. $\dot{\underline{x}} = \begin{bmatrix} 0 & 1 \\ 2 & -3 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \quad [1 \quad 0] \underline{x}$

b. $\dot{\underline{x}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -2 & -1 & -2 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$$\text{c. } \dot{\underline{x}} = \begin{bmatrix} 20 & -4 & 8 \\ -40 & 8 & -20 \\ -60 & 12 & -26 \end{bmatrix} \underline{x}, [1 \quad 0 \quad 1] \underline{x}$$

- 4.21 Show that $|sI - A| = s^4 + 2.985s^3 + 2.42855s^2 + 0.52054s + 0.004016$ when A is the coefficient matrix given by

$$A = \begin{bmatrix} -0.95 & 0.005 & 0 & 0.5 \\ 0.8 & -0.015 & 0 & 0 \\ 0 & 0.01 & -0.32 & 0 \\ 0.15 & 0 & 0.3 & -1.7 \end{bmatrix}$$

- 4.22 Derive the expression for the closed-loop transfer function $\theta(s)/\theta_{\text{com}}(s)$ in Equation 4.282.

- 4.23 Starting with the Laplace transform $\theta_{\text{open-loop}}(s)$ of the open-loop system

$$\theta_{\text{open-loop}}(s) = \left[\frac{K_p}{s(\tau_p s + 1)} \right] \left[\frac{K_s}{(\tau_s s + 1)} \right] U(s)$$

- a. Find $\theta_{\text{open-loop}}(t)$ in response to the pulse input given in Equation 4.285. Leave your answer in terms of the K_p, K_s, τ_p, τ_s and the pulse parameters A and T .
 - b. Verify Equation 4.287 for the final value $\theta_{\text{open-loop}}(\infty)$.
 - c. Verify the open-loop pulse response shown in Figure 4.20.
 - d. Find and plot the open-loop step response
 - i. As the limit as $T \rightarrow \infty$ of the open-loop pulse response.
 - ii. By inverse Laplace transformation of $\theta_{\text{open-loop}}(s)$ when $U(s) = A/s$.
- 4.24 In the ship heading example, the input to the ship yaw dynamics in Figure 4.18 is $R(s)$, the rudder angle in degree.
- a. Find the transfer function $R(s)/\theta_{\text{com}}(s)$.
 - b. Find and plot a graph of $r(t)$ for the case where $\theta_{\text{com}}(t) = 5^\circ, t \geq 0$ and $K_c = 10$. Comment on the results.
 - c. For the same command input $\theta_{\text{com}}(t) = 5^\circ, t \geq 0$ as in part (b), find the maximum controller gain K_c for which the rudder deflection never exceeds 30° . Plot $r(t)$ and $\theta(t)$ for a time sufficient for the system to reach steady state.
- 4.25 For the closed-loop system to control the ship's heading
- a. Find the fourth-order differential equation relating the output $\theta(t)$ and input $\theta_{\text{com}}(t)$.
 - b. Find a suitable choice for matrices A, B, C , and D in the state variable form $\dot{\underline{x}} = A\underline{x} + Bu$, $y = C\underline{x}$ where $u = \theta_{\text{com}}$ and $y = \theta$. Leave your answers in terms of the system parameters K_c, K_p, K_s, τ_p , and τ_s .
- Hint:* Draw a simulation diagram.
- c. Choose the same values for K_p, K_s, τ_p , and τ_s as in the example. Find the characteristic polynomial $\Delta(s)$ as a function of K_c by evaluating $|sI - A|$.
 - d. Prepare a table with two columns. The first column contains values of $K_c = 1, 5, 10, 25, 50, 75, \dots, 200$ V/deg heading, and the second column lists the four closed-loop system poles.
 - e. Use the MATLAB M-file “Ch4_feedback_yaw.m” or write your own to find the value(s) of K_c that results in an underdamped quadratic factor of $\Delta(s)$ with damping ratio equal to 0.5.

- 4.26 The water current speed $v_w(t)$ influences the angle of the ship's rudder and is considered a load variable or disturbance. The open-loop system is redrawn to reflect the disturbance input in Figure E4.26:

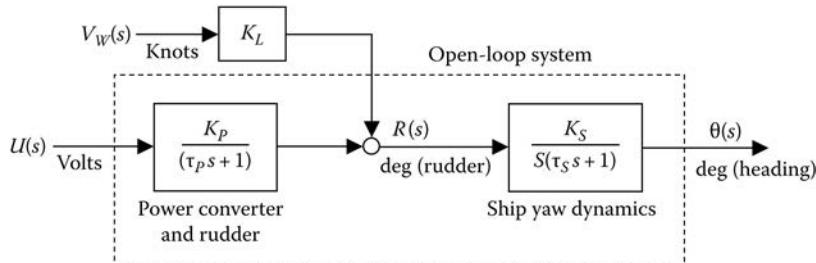


FIGURE E4.26

The load gain K_L can be assumed constant if the angle between the ship's rudder and the water current direction is relatively constant.

a. Find the closed-loop transfer functions $\frac{\theta(s)}{\theta_{\text{com}}(s)} \Big|_{V_W(s)=0}$ and $\frac{\theta(s)}{V_W(s)} \Big|_{\theta_{\text{com}}(s)=0}$

where

$$\theta(s) = \left[\frac{\theta(s)}{\theta_{\text{com}}(s)} \Big|_{V_W(s)=0} \right] \theta_{\text{com}}(s) + \left[\frac{\theta(s)}{V_W(s)} \Big|_{\theta_{\text{com}}(s)=0} \right] V_W(s)$$

- b. Find $\theta(t)$ when $\theta_{\text{com}}(t) = 0$, $t \geq 0$ and $v_w(t) = 2$ kn, $t \geq 0$. Assume the parameter values K_p , K_S , τ_p , and τ_S are the same as in the example. The controller gain $K_C = 7.5$ V/deg heading and the load gain $K_L = 0.5^\circ$ rudder/kn.
- 4.27 A ship with parameters K_p , K_S , τ_p , and τ_S given in the text is traveling in its intended direction, due North as shown in Figure E4.27. The ship cruising speed is 20 kn. The ocean current suddenly switches from zero to five knots in an east-to-west direction. Find the ship's heading $\theta(t)$ with the control system gain $K_C = 5$ V/deg heading.

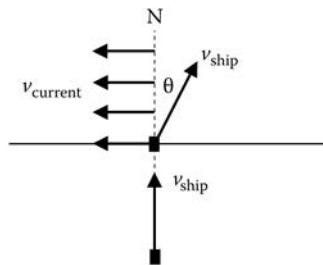


FIGURE E4.27

Hint: Find the new command heading to keep the ship traveling due north.

4.5 FREQUENCY RESPONSE OF LTI CONTINUOUS-TIME SYSTEMS

The response of LTI continuous-time systems to sinusoidal inputs is of interest because it provides an alternative to time domain methods based on the impulse response function to characterize

the system's dynamics. A nonperiodic signal $f(t)$ can be resolved into sinusoidal functions over a continuum of frequencies according to Jackson (1991)

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega t} d\omega \quad (4.295)$$

where the sinusoidal functions are the complex exponentials

$$e^{j\omega t} = \cos \omega t + j \sin \omega t \quad (-\infty < \omega < \infty) \quad (4.296)$$

and the function $F(j\omega)$ is given by

$$F(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (4.297)$$

The complex-valued function $F(j\omega)$ is called the Fourier integral or Fourier transform of the signal $f(t)$. Entire books have been written on the Fourier transform and its applications (Papoulis 1962; Bracewell 1986) while other books in the area of signals and systems (Kailath 1980; Jackson 1991; Kraniauskas 1992) include considerable coverage of the topic. $F(j\omega)$ is a function that assumes complex values over the frequency range $(-\infty, \infty)$. In polar form, $F(j\omega)$ is written as

$$F(j\omega) = A(j\omega)e^{j\phi(j\omega)}, \quad A(j\omega) = |F(j\omega)| \quad \text{and} \quad \phi(j\omega) = \text{Arg}[F(j\omega)] \quad (4.298)$$

where the magnitude $A(j\omega)$ is called the Fourier spectrum of $f(t)$.

In rectangular form,

$$F(j\omega) = R(j\omega) + jX(j\omega), \quad R(j\omega) = \text{Re}\{F(j\omega)\}, \quad X(j\omega) = \text{Im}\{F(j\omega)\} \quad (4.299)$$

If $f(t)$ is causal, that is, $f(t) = 0, t < 0$, it can be expressed as a continuum of the real sinusoidal functions $\cos \omega t$ or $\sin \omega t$ (Papoulis 1962)

$$f(t) = \frac{1}{\pi} \int_0^{\infty} R(j\omega) \cos \omega t d\omega = -\frac{2}{\pi} \int_0^{\infty} X(j\omega) \sin \omega t d\omega, \quad t > 0 \quad (4.300)$$

implying that $R(j\omega)$ and $X(j\omega)$ are not independent.

Suppose an LTI system with transfer function $H(s)$ is subjected to an input $u(t)$ with Fourier transform $U(j\omega)$. By a convolution property similar to the one for Laplace transforms, the Fourier transform of the output $y(t)$ is given by

$$Y(j\omega) = H(j\omega)U(j\omega) \quad (4.301)$$

where $H(j\omega)$ is the system transfer function with s replaced by $j\omega$. $H(j\omega)$ is called the frequency response function of the system. It follows from Equation 4.295

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\omega)U(j\omega) e^{j\omega t} d\omega \quad (4.302)$$

and, therefore, each input component $(1/2\pi)U(j\omega)e^{j\omega t}$ in the continuum of frequencies from $-\infty$ to ∞ is scaled by $H(j\omega)$ and integrated over $(-\infty, \infty)$ to form the output $y(t)$. If the input $u(t) = U_0 \cos \omega_0 t$, its Fourier transform is (Jackson 1991)

$$U(j\omega) = U_0 \pi [\delta(\omega + \omega_0) + \delta(\omega - \omega_0)] \quad (4.303)$$

and Equation 4.302 reduces to (see Exercise 4.28)

$$y(t) = U_0 \cdot |H(j\omega_0)| \cos\{\omega_0 t + \text{Arg}[H(j\omega_0)]\} \quad (4.304)$$

The amplitude of the output is equal to the amplitude of the input multiplied by the magnitude of the frequency response function evaluated at ω_0 . The phase angle (with respect to the input) equals the argument of the frequency response function at ω_0 . Equation 4.304 is an essential property of linear systems and the foundation of AC steady-state analysis of electric circuits. Equation 4.304, valid for stable LTI systems, applies only in the steady state, that is, after the system's natural response has vanished.

In the case of nonlinear systems, the steady-state output in response to a sinusoidal input with frequency ω_0 contains sinusoids at harmonic frequencies $2\omega_0, 3\omega_0, 4\omega_0, \dots$ along with a sinusoidal component at the fundamental frequency ω_0 . Example 4.12 illustrates the property in Equation 4.304 for a simple first-order system.

EXAMPLE 4.12

For the first-order system in Figure 4.22,

- Find the transient and steady-state responses to the input $u(t) = A \sin \omega_0 t$. Leave your answer in terms of the system parameters K and τ and input parameters A and ω_0 .
- Find the frequency response function of the system.
- $A = 1$, $\omega_0 = 2$ rad/s, $K = 3$, and $\tau = 0.5$ s. Plot $u(t)$ and $y(t)$ on the same graph.
- Find the time lag between the input and output at steady state, and verify the result from the graphs of $u(t)$ and $y(t)$.

- For input $u(t) = A \sin \omega_0 t$, $Y(s)$ is given by

$$Y(s) = \frac{K}{\tau s + 1} U(s) = \frac{K}{\tau s + 1} \left(\frac{A \omega_0}{s^2 + \omega_0^2} \right) = \frac{KA \omega_0}{\tau} \left[\frac{1}{(s + 1/\tau)(s^2 + \omega_0^2)} \right] \quad (4.305)$$

Performing a partial fraction expansion of the last term in Equation 4.305 and simplifying,

$$Y(s) = \frac{KA \omega_0}{1 + (\omega_0 \tau)^2} \left[\frac{\tau}{s + 1/\tau} + \frac{1}{s^2 + \omega_0^2} - \frac{\tau s}{s^2 + \omega_0^2} \right] \quad (4.306)$$

The inverse Laplace transform of $Y(s)$ is

$$y(t) = \frac{KA \omega_0}{1 + (\omega_0 \tau)^2} [\tau e^{-t/\tau} + \frac{1}{\omega_0} \sin \omega_0 t - \tau \cos \omega_0 t] \quad (4.307)$$

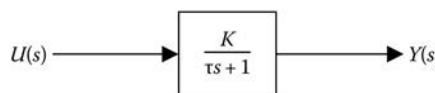


FIGURE 4.22 First-order system ($K > 0$).

Using the trigonometric relationship

$$A \cos \omega_0 t + B \sin \omega_0 t = C \sin(\omega_0 t + \varphi) \quad (4.308)$$

where

$$C = (A^2 + B^2)^{1/2}, \varphi = \tan^{-1}(A/B) \quad (4.309)$$

the $\sin \omega_0 t$ and $\cos \omega_0 t$ terms in Equation 4.307 may be combined into a single term, that is,

$$y(t) = KA \left\{ \frac{\omega_0 \tau}{1 + (\omega_0 \tau)^2} e^{-t/\tau} + \frac{1}{[1 + (\omega_0 t)]^{1/2}} \sin(\omega_0 t + \varphi) \right\} \quad (4.310)$$

where

$$\varphi = -\tan^{-1}(\omega_0 \tau) \quad (4.311)$$

From Equation 4.310, the transient and steady-state responses are

$$y_{tr}(t) = \frac{KA\omega_0 \tau}{1 + (\omega_0 \tau)^2} e^{-t/\tau} \quad (4.312)$$

$$y_{ss}(t) = \frac{KA}{[1 + (\omega_0 \tau)^2]^{1/2}} \sin(\omega_0 t + \varphi) \quad (4.313)$$

b. The frequency response function is

$$H(j\omega) = H(s) \Big|_{s=j\omega} = \frac{K}{\tau s + 1} \Big|_{s=j\omega} \quad (4.314)$$

$$= \frac{K}{1 + j\omega\tau} \quad (4.315)$$

From Equation 4.314, the magnitude and phase angle of $H(j\omega)$ are

$$|H(j\omega)| = \left| \frac{K}{1 + j\omega\tau} \right| \quad (4.316)$$

$$= \frac{K}{[1 + (\omega\tau)^2]^{1/2}} \quad (K > 0) \quad (4.317)$$

$$\text{Arg } H(j\omega) = -\tan^{-1}(\omega\tau) \quad (4.318)$$

c. Substituting the given values for A , K , τ , and $\omega = \omega_0$ gives

$$y_{tr}(t) = \frac{(3)(1)(2)(0.5)}{[1 + [(2)(0.5)]^2]^{1/2}} e^{-t/\tau} = 1.5e^{-2t} \quad (4.319)$$

$$y_{ss}(t) = \frac{(3)(1)}{\{1 + [(2)(0.5)]^2\}^{1/2}} \sin\{2t - \tan^{-1}[(2)(0.5)]\} \quad (4.320)$$

$$= 1.5\sqrt{2} \sin\left(2t - \frac{\pi}{4}\right) \quad (4.321)$$

The input $u(t) = \sin 2t$ and output $y(t) = 1.5e^{-2t} + 1.5\sqrt{2}\sin(2t - \pi/4)$ are shown in [Figure 4.23](#). The transient response dies out in approximately $5\tau = 5(0.5) = 2.5$ s.

- d. [Figure 4.24](#) is a close-up of [Figure 4.23](#) near the peaks of $u(t)$ and $y(t)$. The lag time T is estimated as $T \approx 4.31 - 3.92 = 0.39$ s in agreement with the exact value

$$\omega_0 T = \varphi \Rightarrow T = \frac{\varphi}{\omega_0} = \frac{\pi/4}{2} = \frac{\pi}{8} = 0.393\text{s} \quad (4.322)$$

This example illustrates how the steady-state sinusoidal response of an LTI system can be obtained considerably faster using the frequency response function compared to methods that determine the complete response.

Graphical tools exist for conveying the magnitude and phase properties of an LTI continuous-time system with transfer function $H(s)$. The simplest one consists of graphs of $|H(j\omega)|$ and $\text{Arg } H(j\omega)$ vs. ω . The graphs are typically plotted over a frequency range of interest. Control systems engineers and analog filter designers prefer a variation of the frequency response plots in which $20 \log |H(j\omega)|$, the magnitude measured in decibels (db), is plotted vs. ω on a logarithmic scale. The result (along with the phase plot) is called a Bode diagram or Bode plot.

To illustrate, consider a system with transfer function

$$H(s) = \frac{\omega_b^3}{(s + \omega_b)(s^2 + \omega_b s + \omega_b^2)} \quad (4.323)$$

which describes a third-order low-pass Butterworth filter designed to pass frequencies in the band 0 (DC) to ω_b and reject all others. The M-file “*Ch4_Fig4_25.m*” includes statements to evaluate the magnitude and phase of $H(s)$ when $\omega_b = 100$ rad/s for frequencies between 10^0 and 10^4 rad/s. The Bode plot is shown in [Figure 4.25](#).

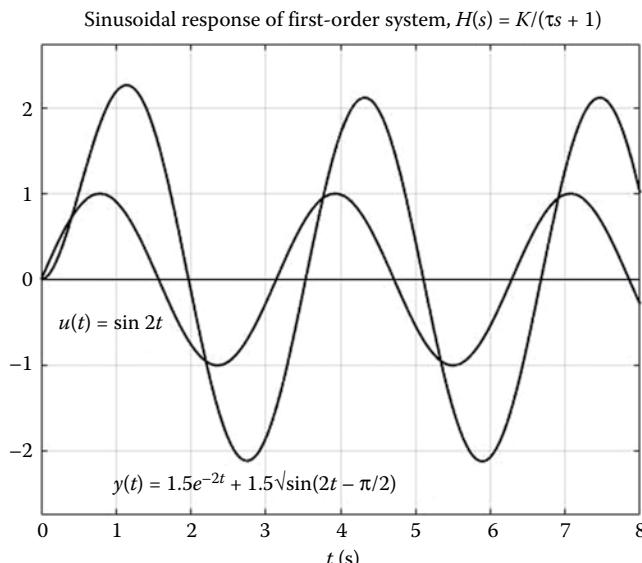


FIGURE 4.23 Graph of input $u(t)$ and output $y(t)$.

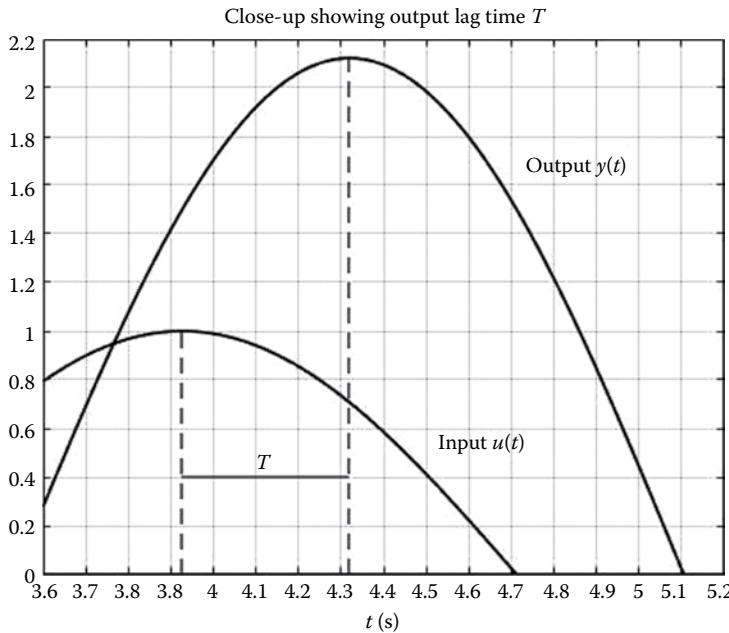


FIGURE 4.24 Close-up of input and response near peaks.

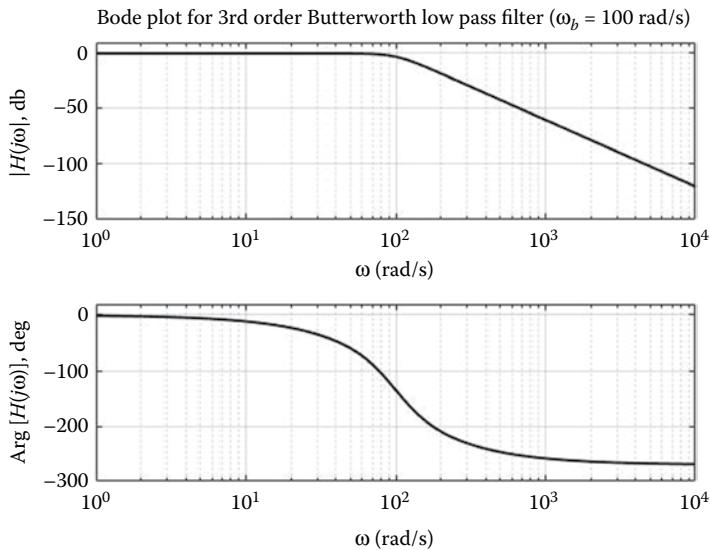


FIGURE 4.25 Bode plot for third-order Butterworth low-pass filter ($\omega_b = 100$ rad/s).

The control system toolbox, a complementary suite of utilities designed for use with the MATLAB environment, includes a function “`bode`” for drawing the Bode plot of an LTI system. The control system toolbox is covered later in Section 4.4.10.

The magnitude measured in db (sometimes referred to as the gain) is close to zero, and, hence, the magnitude is close to 1 over a considerable portion of the interval $0 \leq \omega \leq \omega_b$. At $\omega = \omega_b$,

$$|H(j\omega_b)| = \left| \frac{\omega_b^3}{(s + \omega_b)(s^2 + \omega_b s + \omega_b^2)} \right|_{s=j\omega_b} = \frac{1}{|-1 + j|} = \frac{1}{\sqrt{2}} \quad (4.324)$$

$$\Rightarrow 20 \log |H(j\omega_b)| = 20 \log \frac{1}{\sqrt{2}} \approx -3 \text{ db} \quad (4.325)$$

The gain is -3 db at $\omega = \omega_b$ and starts falling off from ω_b at approximately 60 db for every 10-fold increase in frequency (decade) (see [Figure 4.25](#)).

The frequency response function of a system dictates the extent to which sinusoidal inputs at specific frequencies are passed or rejected by the system, and coupled with the fact that input time signals can be resolved into sinusoids over a continuum of frequencies, explains why linear systems are often called linear filters.

The individual components in a linear feedback control system such as sensors, controllers, and power converters are examples of continuous-time filters, which transmit the range of frequencies in the input according to their frequency response function. Control system design based on frequency response relies on assumptions related to the frequency content of the command inputs and the uncontrollable inputs, referred to as load variables or disturbances.

A simple unity feedback control system is shown in [Figure 4.26](#). $R(s)$ and $D(s)$ are the reference (command) and disturbance inputs.

The open-loop system model is

$$Y(s) = G_P(s)[U(s) + G_D(s)D(s)] \quad (4.326)$$

The control system output $Y(s)$ can be written as

$$Y(s) = T_R(s)R(s) + T_D(s)D(s) \quad (4.327)$$

where

$$T_R(s) = \frac{G_C(s)G_P(s)}{1 + G_C(s)G_P(s)}, \quad T_D(s) = \frac{G_D(s)G_P(s)}{1 + G_C(s)G_P(s)} \quad (4.328)$$

It frequently happens that the command input $r(t)$ is a slow varying signal compared to the disturbance input $d(t)$. Assuming $G_P(s)$ and $G_D(s)$ are fixed, proper design entails selecting a controller transfer function $G_C(s)$ to simultaneously make $|T_R(j\omega)|$ close to 1 at the lower frequencies contained in $r(t)$ and $|T_D(j\omega)|$ close to zero for the frequencies present in $d(t)$. Suppose the command input is band-limited from 0 to 0.25 Hz (1.57 rad/s) and the disturbance frequencies start at roughly 10 Hz (62.8 rad/s) and the open-loop system transfer functions are

$$G_P(s) = \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{s^2 + 2.25s + 0.5625} \quad (4.329)$$

$$G_D(s) = K_D = 40 \quad (4.330)$$

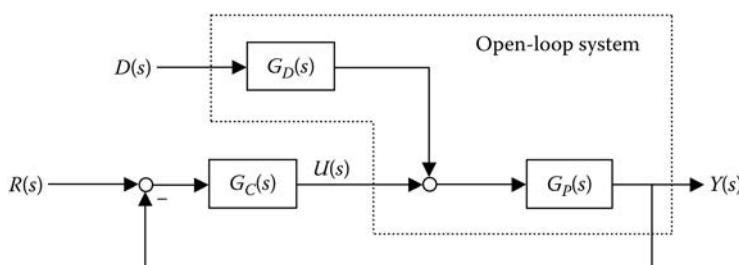


FIGURE 4.26 A feedback control system with command and disturbance inputs.

The controller is of the proportional plus integral (P-I) type,

$$G_C(s) = K_C + \frac{K_I}{s} = 5 + \frac{2}{s} \quad (4.331)$$

Bode plots of $T_R(j\omega)$ and $T_D(j\omega)$ are generated in "Ch4_Fig4_27.m" and shown in Figure 4.27. The frequency content of the command input $r(t)$ is confined primarily to frequencies below 1.57 rad/s. The output will track the input closely since the gain $20 \log |T_R(j\omega)|$ is roughly 0 db, corresponding to a magnitude of 1 from DC ($\omega = 0$) to approximately 1 rad/s. The phase angle $\text{Arg}[T_R(j\omega)]$ is close to 0° from $\omega = 0$ to $\omega \approx 0.5$ rad/s and is -36.1° at $\omega = 1.57$ rad/s.

Conversely, the gain $20 \log |T_D(j\omega)| = -40$ db, which is equivalent to a magnitude of 0.01 at approximately 62 rad/s. The control system effectively filters out the disturbances by attenuating all frequencies above 62.8 rad/s.

The steady-state error, $e_{ss} = y(\infty) = r(\infty)$, is zero when $r(t)$ or $d(t)$ is constant. This can be demonstrated by showing that the DC gains $T_R(j0) = 1$ and $T_D(j0) = 0$, a direct consequence of the open-loop gain $G_C(0)G_p(0) = \infty$. The infinite open-loop gain results from the presence of the integrator in $G_C(s)$. While zero steady-state error is a desirable condition, we must still be mindful of the location of the control system's characteristic roots since it determines the transient response.

The transfer functions of real-world components and complete systems possess Bode plots in which the gain "rolls off" at high frequencies. Properly designed closed-loop control systems track low-frequency command inputs reasonably well. Further increases in frequency require excessive power be delivered to control system components, thus limiting the system's ability to track higher frequency command inputs.

Any component or system with transfer function $G(s)$ given by the ratio of polynomials in proper fraction form, that is, numerator polynomial, is lower order than denominator will satisfy

$$\lim_{\omega \rightarrow \infty} |G(j\omega)| = 0 \Rightarrow \lim_{\omega \rightarrow \infty} 20 \log |G(j\omega)| = -\infty \quad (4.332)$$

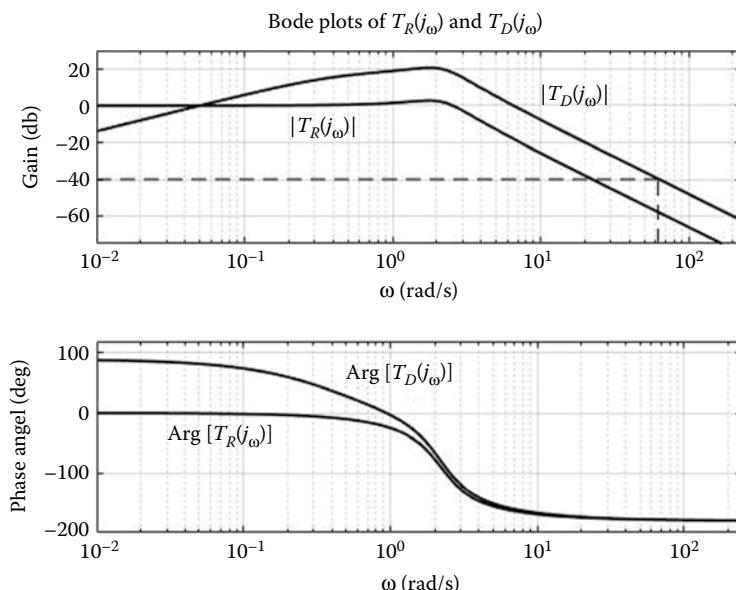


FIGURE 4.27 Bode plot for closed-loop frequency response functions $T_R(j\omega)$ and $T_D(j\omega)$.

A common measure of the frequency where “roll off” begins is ω_b and the interval $(0, \omega_b)$ is called the bandwidth of the system. The frequency ω_b satisfies

$$|G(j\omega_b)| = \frac{1}{2^{1/2}} |G(j0)| \Rightarrow 20\log|G(j\omega_b)| = 20\log|G(j0)| - 3\text{db} \quad (4.333)$$

Consequently, ω_b is the (lowest) frequency at which the gain (magnitude function measured in db) is 3 db below the DC gain of the system.

Consider the first-order system in [Figure 4.22](#) with magnitude function $|H(j\omega)|$ given in Equation 4.316. The frequency ω_b is obtained from

$$|H(j\omega_b)| = \frac{K}{[1 + (\omega_b\tau^2)^{1/2}]} = \frac{1}{2^{1/2}} \cdot |H(j0)| = \frac{1}{2^{1/2}} \cdot K \quad (4.334)$$

$$\Rightarrow 1 + (\omega_b\tau)^2 = 2 \quad (4.335)$$

$$\Rightarrow \omega_b = \frac{1}{\tau} \quad (4.336)$$

Equation 4.336 is important because it relates ω_b , a frequency domain parameter to the time constant τ , which characterizes the system’s transient response in the time domain. Furthermore, being inversely proportional to the system, time constant tells us that the bandwidth frequency ω_b is a measure of the speed of response of the first-order system. Hence, first-order systems like the one in [Figure 4.22](#) with a fast natural mode (τ small) exhibit larger bandwidths.

For a second-order system with transfer function

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.337)$$

increasing the natural frequency ω_n (with ζ constant) decreases the transient response time regardless of whether the system is underdamped, overdamped, or critically damped (see expressions for step response in [Section 2.3](#)). It is left as an exercise to show that the bandwidth frequency ω_b for the system with transfer function in Equation 4.337 is proportional to ω_n . Specifically,

$$\omega_b = \left[1 - 2\zeta^2 + (2 - 4\zeta^2 + 4\zeta^4)^{1/2} \right]^{1/2} \omega_n, \quad (K = 1) \quad (4.338)$$

and, therefore, ω_b is a measure of the speed of response for a second-order system as well.

A Bode plot for three second-order systems, all with $\omega_n = 1$ rad/s and damping ratios of $\zeta = 0.25, 1, 2$, is shown in [Figure 4.28](#). Also shown is an enlargement of the plots for the purpose of estimating the corresponding bandwidths. The calculated values of ω_b from Equation 4.338 are 1.4845 rad/s ($\zeta = 0.25$), 0.6436 rad/s ($\zeta = 1$), and 0.2666 rad/s ($\zeta = 2$) in agreement with the values estimated from [Figure 4.28](#).

[Figure 4.28](#) shows a peak in the gain (and magnitude function) for the underdamped system indicating the presence of a resonant frequency. The resonant frequency is $\omega_r = 0.935$ rad/s with $|G(j\omega_r)| = 2.0656$ (6.3 db). Not all underdamped second-order systems exhibit resonance (see [Exercise 4.32](#)).

The Bode plots and bandwidth calculations are handled in the MATLAB script file “[Ch4_Fig4_28.m](#).”

The step responses of the three second-order systems are shown in [Figure 4.29](#). The rise time is defined as $t_r = t_{0.9} - t_{0.1}$, where $t_{0.1}$ and $t_{0.9}$ are the times required for the step response to

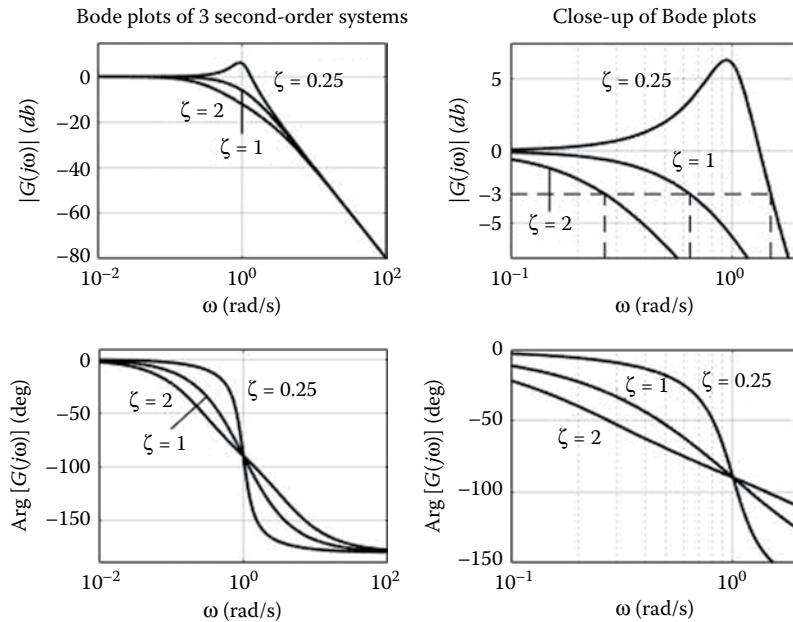


FIGURE 4.28 Bode plots for second-order systems ($\omega_n = 1$ rad/s) with $\zeta = 0.25, 1, 2$.

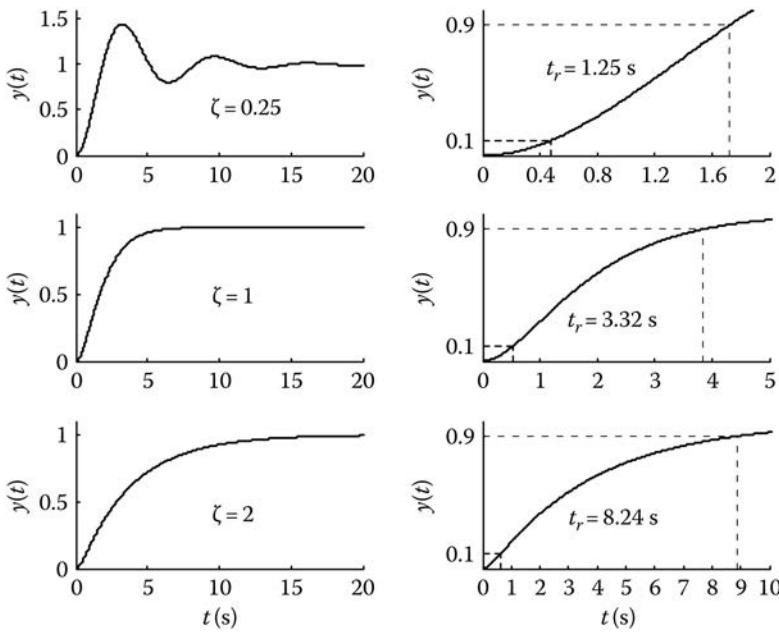


FIGURE 4.29 Step responses and rise times for three second-order systems.

reach 10% and 90% of its final value, respectively. The rise time is another measure of the system's speed of response. The times $t_{0.1}$ and $t_{0.9}$ and the approximate rise times are shown on the zoomed-in plots of the step responses. As expected, the lightly damped system ($\zeta = 0.25$) with the greatest bandwidth responds the quickest (shortest rise time) while the overdamped system ($\zeta = 2$) with the smallest bandwidth is the most sluggish and least responsive.

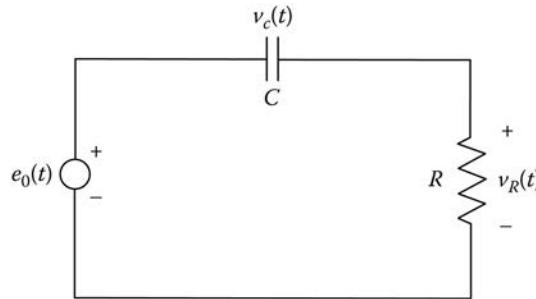


FIGURE 4.30 Circuit with high-pass filter transfer function.

The step responses are generated in the M-file “*Ch4_Fig4_29.m*.”

LTI systems modeled by transfer functions where the order of the numerator and that of the denominator polynomials are equal, that is, a direct connection exists from the input to the output, exhibit finite gain at frequencies approaching infinity. That is,

$$\lim_{\omega \rightarrow \infty} |H(j\omega)| = \lim_{\omega \rightarrow \infty} \left| \frac{\alpha_n s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0}{\beta_n s^n + \beta_{n-1} s^{n-1} + \dots + \beta_1 s + \beta_0} \right|_{s=j\omega} = \frac{\alpha_n}{\beta_n} \quad (4.339)$$

Since a real system cannot respond in a way suggested by Equation 4.339, the transfer function $H(s)$ with equal order polynomials in the numerator and denominator, or equivalently the same number of finite zeros and poles, is an ideal approximation that breaks down above a certain frequency. Nonetheless, it is a useful approximation to the transfer function of a system that readily passes high-frequency components present in its input(s), as in the case of a high-pass filter. Of course, when the high-frequency signals represent unwanted noise, which is invariably present in control systems, the closed-loop transfer function should be designed to attenuate the noise (see Exercise 4.34).

The simple *RC* circuit in Figure 4.30 with the voltage $v_R(t)$ as output is an example of a high-pass filter. The transfer function is

$$H(s) = \frac{V_R(s)}{E_0(s)} = \frac{RC_s}{RC_s + 1} \quad (4.340)$$

At high frequencies ($\omega \gg 1/RC$), the magnitude $|H(j\infty)| \approx 1$ (0 db). Note that the capacitor behaves like a short circuit at high frequencies.

4.5.1 STABILITY OF LINEAR FEEDBACK CONTROL SYSTEMS BASED ON FREQUENCY RESPONSE

Linear control systems are a class of LTI systems, and the basic premises of stability presented in the previous section are applicable. The following is a brief introduction to stability, as it applies to simple feedback control systems from the viewpoint of frequency response. For a more detailed discussion of the subject, the reader is encouraged to refer to any of the texts in linear feedback control systems listed in the References.

Figure 4.31 is a block diagram of a servo control system with transfer functions for the controller, actuator, plant, and sensor/transmitter.

Insight into the stability of the system can be ascertained by tracking the response to the error signal $e(t) = \mathcal{L}^{-1}\{E(s)\}$ as it propagates around the loop. Suppose the loop is broken immediately following the transmitter and a test signal $e(t) = \sin \omega t$ is inserted at the controller input. Each component along the open-loop path processes a sinusoidal input and delivers a sinusoidal output

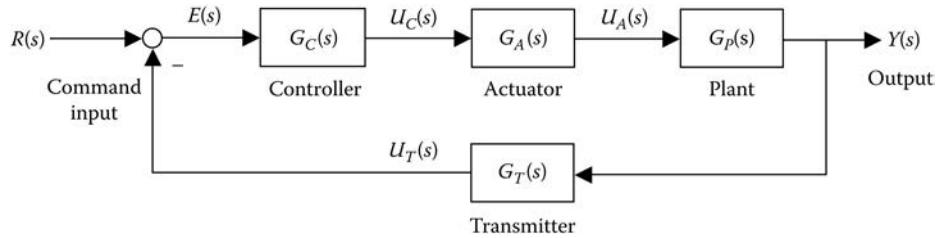


FIGURE 4.31 Block diagram of representative linear feedback control system.

(both at radian frequency ω) to the next component. Magnitude and phase shift of the individual sinusoids are determined by the frequency response functions of each component at radian frequency ω .

The closed-loop control system is unstable if $-u_T(t) = -\mathcal{L}^{-1}\{U_T(s)\}$ is ever in phase with $e(t)$ and its amplitude is greater than one. When this occurs, the error signal propagates around the loop and increases in magnitude while doing so. Conversely, when $e(t)$ and $-u_T(t)$ are in phase and $| -u_T(t) | = | u_T(t) | < 1$, a stable system results. Finally, a marginally stable system exists when $e(t)$ and $-u_T(t)$ are in phase and $| -u_T(t) | = | u_T(t) | = 1$.

Since the negative sign in $-u_T(t)$ is equivalent to -180° phase shift, $-u_T(t)$ will be in phase with $e(t)$ whenever $u_T(t)$ lags $e(t)$ by -180° , that is, there is a combined total of -180° phase lag in the open-loop system. The frequency at which this occurs is called the phase crossover frequency ω_{cp} . Hence, for a closed-loop, negative feedback control system to be marginally stable (or unstable), there must exist at least one frequency where the open-loop phase lag is equal to -180° .

The open-loop transfer function is

$$G_{OL}(s) = G_C(s)G_A(s)G_P(s)G_T(s) \quad (4.341)$$

For this example, assume the dynamics of each component are described by

$$G_C(s) = K_C, \quad G_A(s) = \frac{K_A}{\tau_A s + 1}, \quad G_P(s) = \frac{K_P}{s(\tau_P s + 1)}, \quad G_T(s) = \frac{K_T}{\tau_T s + 1} \quad (4.342)$$

where $K_C = 0.25$, $K_A = 2$, $\tau_A = 0.25$, $K_p = 8$, $\tau_p = 4$, $K_T = 0.1$, and $\tau_T = 0.003$.

The open-loop transfer function becomes

$$G_{OL}(s) = K_C \frac{K_A}{\tau_A s + 1} \cdot \frac{K_P}{s(\tau_P s + 1)} \cdot \frac{K_T}{\tau_T s + 1} \quad (4.343)$$

$$= (0.25) \frac{2}{0.25s + 1} \cdot \frac{8}{s(4s + 1)} \cdot \frac{0}{0.003s + 1} \quad (4.344)$$

$$= \frac{0.4}{s(0.25s + 1)(4s + 1)(0.003s + 1)} \quad (4.345)$$

A Bode plot of the open-loop transfer function is shown in [Figure 4.32](#).

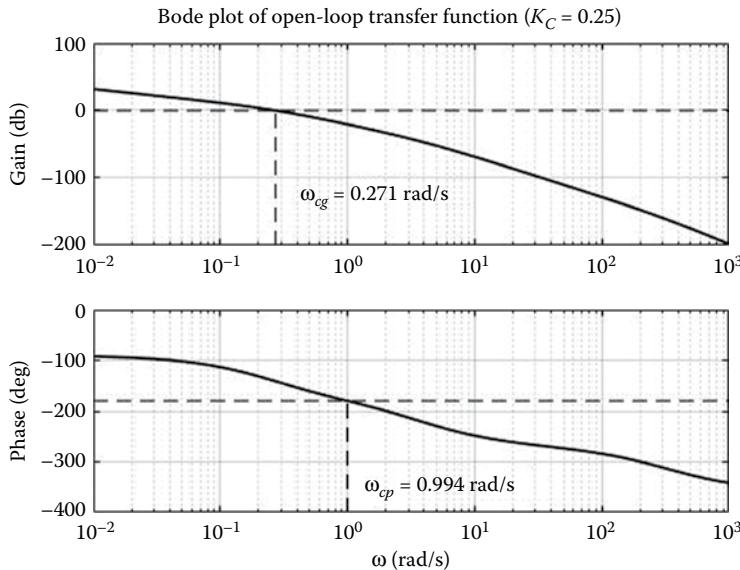


FIGURE 4.32 Bode plot of $G_{OL}(s)$ for stable system ($K_C = 0.25$).

Inspection of Equation 4.345 reveals the open-loop phase varies from -90° at $\omega = 0$ to -360° at $\omega \rightarrow \infty$ indicating the possibility of a marginally stable or unstable system.

The phase crossover frequency ω_{cp} , was determined by trial and error to be approximately 0.9936 rad/s. As a check,

$$\text{Arg}[G_{OL}(j0.9936)] \approx -180^\circ \quad (4.346)$$

The magnitude function evaluated at $\omega_{cp} \approx 0.9936$ rad/s is

$$|G_{OL}(j\omega_{cp})| = G_{OL}(j0.9936)| = 0.0953(-20.4 \text{ db}) \quad (4.347)$$

The system is stable since the magnitude function is less than one, or equivalently the gain is less than 0 db, at the phase crossover frequency. The gain of -20.4 db is a measure of stability. Control engineers would say the “gain margin” is 20.4 db.

Another indicator of stability, the “phase margin,” is the difference between the open-loop phase lag and -180° at the frequency where the gain is 0 db. This frequency, called the gain crossover frequency ω_{cg} , is approximately 0.271 rad/s for the stable system in Figure 4.32. Since $\text{Arg}[G_{OL}(j\omega_{cg})] = \text{Arg}[G_{OL}(j0.271)] = -141.2^\circ$, the phase margin is equal to $-142.1 - (-180) = 37.9^\circ$. Higher phase margins imply a greater measure of relative stability.

Increasing the controller gain K_C generally makes the system more responsive. Consider raising the gain K_C by an amount sufficient to make the system marginally stable, that is, $|G_{OL}(j\omega_{cp})| = 1 \Rightarrow 20 \log |G_{OL}(j\omega_{cp})| = 0 \text{ db}$. From Equation 4.347, it follows that if we multiply the current gain $K_C = 0.25$ by $1/|G_{OL}(j\omega_{cp})| = 1/0.0953$, the new open-loop gain will be equal to 0 db at ω_{cp} (which remains unchanged at 0.9936 rad/s). The Bode plot of the open-loop system transfer function when $K_C = 0.25(1/0.0953) = 2.62$ is shown in Figure 4.33.

The gain crossover frequency is identical to the phase crossover frequency, and the two stability margins have been reduced to zero. The control system is marginally stable, and there will be persistent oscillations at the crossover frequency 0.9936 rad/s in the natural response of the system.

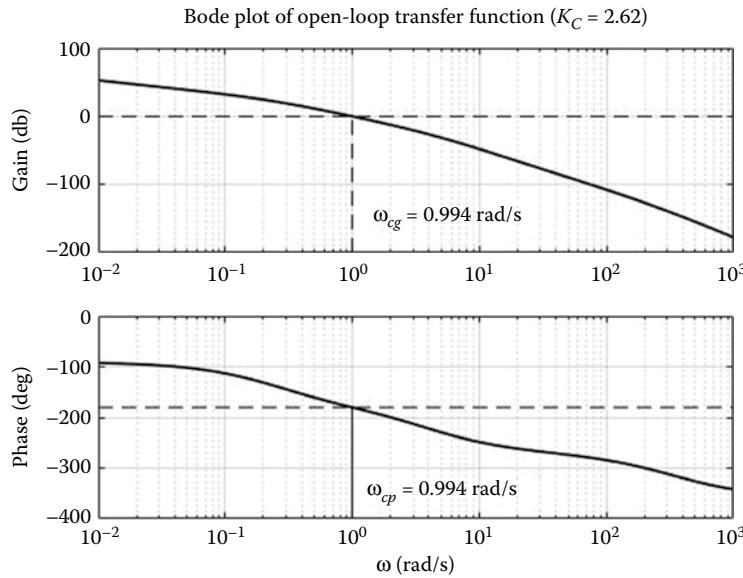


FIGURE 4.33 Bode plot of $G_{OL}(s)$ for marginally stable system.

The closed-loop transfer function is

$$G_{CL}(s) = \frac{G_C(s)G_A(s)G_P(s)}{1 + G_C(s)G_A(s)G_P(s)G_T(s)} \quad (4.348)$$

and the closed-loop system poles are the roots of

$$1 + G_C(s)G_A(s)G_P(s)G_T(s) = 1 + (2.6224) \frac{2}{0.25s+1} \cdot \frac{8}{s(4s+1)} \cdot \frac{0}{0.003s+1} = 0 \quad (4.349)$$

$$\Rightarrow (0.25s+1)s(4s+1)(0.003s+1) + (2.6224)(2)(8)(0.1) = 0 \quad (4.350)$$

$$\Rightarrow 0.003s^4 + 1.01275s^3 + 4.253s^2 + s + 4.1958 = 0 \quad (4.351)$$

Solving the characteristic equation above produces the four closed-loop system poles,

$$s_1 = -333.3, s_2 = -4.25, s_3 = j0.9936, s_4 = -j0.9936$$

demonstrating the marginal stability (poles on the imaginary axis) of the system as well as the frequency of sustained oscillations, namely, $\omega_{cp} = 0.9936$ rad/s.

Further increase in controller gain K_C produces an unstable system resulting in negative stability margins (gain and phase) as well as closed-loop system poles in the right-half plane. Superior performance requires a different type of controller, that is, one which provides sufficient phase lead in the vicinity of the gain crossover frequency for adequate stability and possibly phase lag at lower frequencies to improve steady-state response. Indeed, this is the essence of synthesizing controllers for feedback control systems using frequency response methods. Simulation is an indispensable tool for verifying control system design.

EXERCISES

- 4.28 Use Equations 4.302 and 4.303 to derive Equation 4.304.
- 4.29 The Fourier spectrum $|F(j\omega)|$ of a signal $f(t)$ can be used to find the energy in the signal in the frequency spectrum (ω_1, ω_2) according to

$$E_f(\omega_1, \omega_2) = \int_{\omega_1}^{\omega_2} |F(j\omega)|^2 d\omega$$

- a. Find the Fourier transform of the exponential $f(t) = \begin{cases} 0, & t < 0 \\ e^{-\alpha t}, & t \geq 0 \end{cases}$.
- b. Find and graph $|F(j\omega)|$.
- c. Find ω_0 such that $E_f(0, \omega_0) = 1/2 E_f(0, \infty)$.
- 4.30 For the third-order Butterworth filter in Equation 4.323 with $\omega_b = 2\pi$ rad/s, find
- The poles of $H(s)$.
 - The impulse response function $h(t)$.
 - The filter output at steady state when the input is $u(t) = \sin(0.5\omega_b t) + \sin(2\omega_b t)$.
- 4.31 Derive Equation 4.338 relating the bandwidth and natural frequency of a second-order system in standard form.
- 4.32 For a second-order system with natural frequency $\omega_n = 1$ rad/s, find
- The maximum value of ζ for which the system has a resonant frequency.
 - The resonant frequency if $\zeta = 0$.
 - The response when $\zeta = 0$ to a sinusoidal input at the resonant frequency.
- 4.33 The circuit shown in Figure E4.33 is designed to block 60 Hz noise in the input $v_i(t)$ from appearing in the output $v_o(t)$.

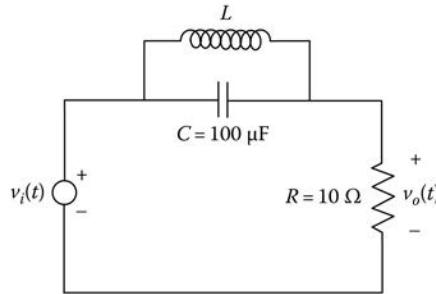
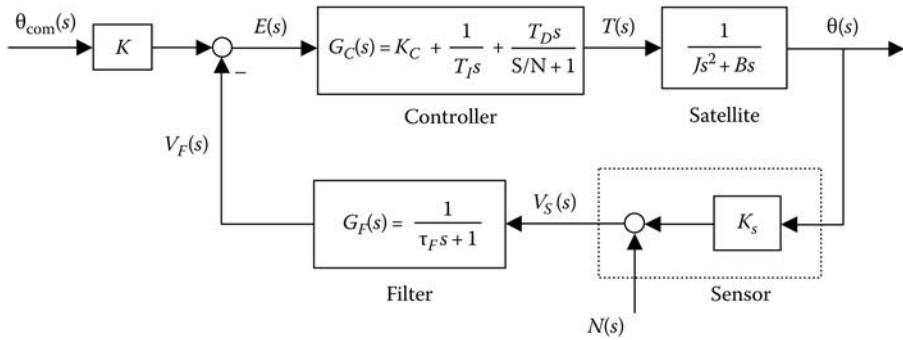


FIGURE E4.33

- a. Show that the transfer function $H(s) = V_o(s)/V_i(s) = (R(LCs^2 + 1))/(RLCs^2 + Ls + R)$.
- b. Find the frequency response function $H(j\omega)$.
- c. Find the inductance L for which $|H(j2\pi \cdot 60)| = 0$.
- d. Write an M-file to draw a Bode plot for $10^2 \leq \omega \leq 10^4$ rad/s.
- e. Find and graph $v_o(t)$ when $v_i(t) = \sin(2\pi \cdot 55)t + \sin(2\pi \cdot 60)t$.
- f. Find and graph $v_o(t)$ when $v_i(t) = \sin(2\pi \cdot 100)t + \sin(2\pi \cdot 60)t$.
- 4.34 A system for controlling the attitude of a rigid satellite is shown in Figure E4.34:

The controller determines the torque $T(t)$ developed by a pair of thrusters to control the satellite's attitude $\theta(t)$. The controller input is an error voltage signal $e(t)$, which is the difference between the commanded attitude $\theta_{com}(t)$ converted to a voltage and the filtered sensor output $v_F(t)$. The sensor output voltage $v_S(t)$ contains an additive noise component $n(t)$.

A low-pass filter is inserted between the comparator and the sensor output to attenuate the noise in the feedback signal. The gain K converts the commanded angle (deg) to a voltage for

**FIGURE E4.34**

comparison to the output voltage from the filter $v_F(t)$. The numerical value of K is the same as the sensor gain K_s .

The command and noise inputs are

$$\theta_{\text{com}}(t) = \begin{cases} 0, & t < 0 \\ At, & 0 \leq t < t_0 \\ At_0, & t \geq t_0 \end{cases}$$

$$n(t) = N_0 \sin \omega_0 t, \quad t \geq 0$$

Baseline parameter values are

$$K = K_s = 0.1 \text{ V/deg},$$

$$J = 150 \text{ ft lb/deg/s}^2, B = 15 \text{ ft lb/deg/s}$$

$$K_C = 10 \text{ ft lb/V}, T_l = 60 \text{ s}, T_D = 30 \text{ s}, N = 10$$

$$A = 1.2^\circ/\text{s}, t_0 = 2.5 \text{ s}$$

$$N_0 = 1 \text{ V}, \omega_0 = 50 \text{ Hz}$$

In parts (a) through (d), assume the filter is not present, that is, $V_S(s)$ is input to the summer.

- Find the transfer functions $H_{\text{com}}(s) = \theta(s)/\theta_{\text{com}}(s)$ and $H_N(s) = \theta(s)/N(s)$. Leave your answers in terms of the parameters $K, K_s, J, K_C, T_l, T_D, N$.
- Obtain Bode Plots for $H_{\text{com}}(j\omega)$ and $H_N(j\omega)$.
- Find and graph $\theta(t)$, $t \geq 0$.
- Find and graph the torque $T(t)$, $t \geq 0$.

In parts (e) through (i) the filter is present.

- Find the filter time constant τ_F if the filter gain is -40 db at the noise frequency.
- Find the transfer functions $H_{\text{com}}(s) = \theta(s)/\theta_{\text{com}}(s)$ and $H_N(s) = \theta(s)/N(s)$. Leave your answers in terms of $K, K_s, J, K_C, T_l, T_D, N, \tau_F$.
- Obtain Bode Plots for $H_{\text{com}}(j\omega)$ and $H_N(j\omega)$ using the value for τ_F .
- Find and graph $\theta(t)$, $t \geq 0$.
- Find the gain and phase margins of the closed-loop system.

4.35 For the control system shown in [Figure 4.31](#),

- a. Use the given baseline parameter values (except K_C), and fill in the missing values in [Table E4.35](#):

TABLE E4.35

	$K_C = 0.1$	$K_C = 0.25$	$K_C = 1$	$K_C = 2.5$
Phase margin				
Gain margin				
Band margin				

- b. Compare the step responses for each of the cases in [Table E4.35](#).

4.6 z-TRANSFORM

Difference equations result from approximation of continuous-time differential equation models. Inputs to the difference equations are commonly discrete-time signals resulting from sampling a continuous-time signal (sample data systems). Inherently discrete-time systems are modeled by difference equations relating inputs and outputs that change only at discrete points in time, as in the case of a numeric processor with a fixed cycle time or a loan balance with monthly payments to reduce the outstanding balance.

In the same way, we characterized continuous-time signals and continuous-time systems; discrete-time counterparts (signals and systems) can be analyzed with the help of a mathematical transformation. Instead of an integral transformation from a continuous-time signal $f(t)$, $t \geq 0$ to its Laplace transform $F(s)$, a different type of mapping is applied to a discrete-time function $f(k)$ or f_k , $k = \dots -3, -2, -1, 0, 1, 2, 3, \dots$. Similar to $F(s)$, the z -transform $F(z)$ is a complex-valued function, that is, s and z are both complex variables. Only causal signals, those that satisfy $f_k = 0$, $k = \dots -3, -2, -1$, will be considered.

The z -transform of a causal discrete-time signal f_k , $k = 0, 1, 2, 3, \dots$ denoted $F(z)$ or $z\{f_k\}$ is defined by the infinite series

$$F(z) = z\{f_k\} = \sum_{k=0}^{\infty} f_k z^{-k} \quad (4.352)$$

The region of convergence of $F(z)$ in the z -plane is all complex numbers greater than a certain distance from the origin, that is, $|z| > R$ where R depends on the particular sequence of numbers (discrete-time signal) f_k (Kuo 1980). As in the case of the Laplace transformation, the region of convergence of the z -transform for a particular discrete-time signal is of passing interest. The main consideration is that the sum in Equation 4.352 converges to a complex number somewhere in the z -plane. Several simple discrete-time signals and their z -transforms follow. The derivations follow directly from the definition in Equation 4.352.

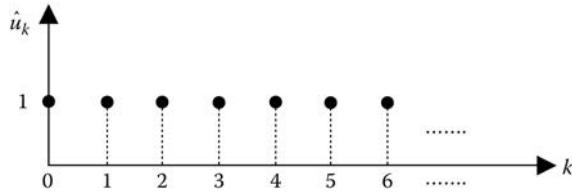
EXAMPLE 4.13

Find the z -transform of the unit step $\hat{u}_k = 1$, $k = 0, 1, 2, 3, \dots$ shown in [Figure 4.34](#).

$$U(z) = z\{\hat{u}_k\} = \sum_{k=0}^{\infty} 1 \cdot z^{-k} = 1 + z^{-1} + (z^{-1})^2 + (z^{-1})^3 + \dots \quad (4.353)$$

The infinite series converges to a sum, that is,

$$U(z) = \sum_{k=0}^{\infty} (z^{-1})^k = \frac{1}{1-z^{-1}} = \frac{z}{z-1} \quad (4.354)$$

**FIGURE 4.34** The discrete-time unit step.

provided $|z^{-1}| < 1$ or equivalently $|z| > 1$. Hence, the region of convergence is outside the Unit Circle, $|z| = 1$. A closed form for $U(z)$ is preferable to the infinite series and often easy to recognize when u_k is a simple expression.

EXAMPLE 4.14

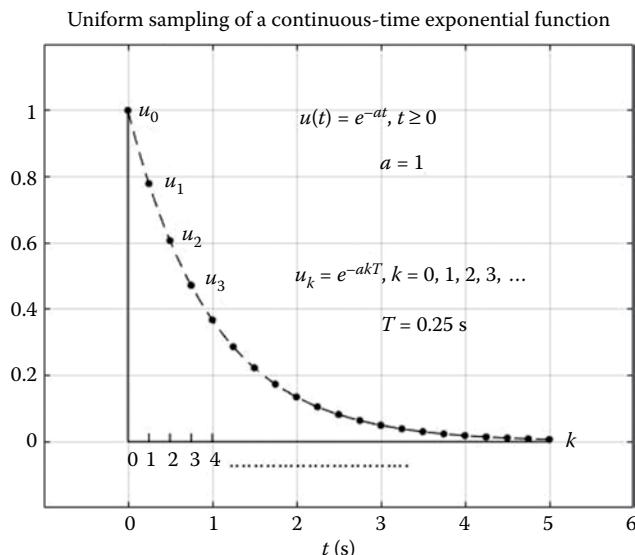
- Find the z-transform of the discrete-time signal u_k resulting from sampling the continuous-time function $u(t) = e^{-at}$, $t \geq 0$ every T s.
- Suppose $u(t)$ and u_k are as shown in Figure 4.35. Find $U(z)$.
- Sampling a continuous-time signal $u(t)$ every T s results in a discrete-time signal u_k where $u_k = u(t)|_{t=kT} = u(kT)$, $k = 0, 1, 2, 3, \dots$. Hence, from the definition of the z-transform and $u_k = e^{-akT}$, $k = 0, 1, 2, \dots$,

$$U(z) = \sum_{k=0}^{\infty} e^{-akT} z^{-k} \sum_{k=0}^{\infty} (e^{-aT} z^{-1})^k = \frac{1}{1 - e^{-aT} z^{-1}} = \frac{z}{z - e^{-aT}}, \quad |z| > e^{-aT} \quad (4.355)$$

Note the dependence of $U(z)$ on the sampling interval T .

- For $a = 1$, $T = 0.25$

$$U(z) = \frac{z}{z - e^{-0.25}}, \quad |z| > e^{-0.25} \quad (4.356)$$

**FIGURE 4.35** Uniform sampling of a continuous-time exponential function.

The next example looks at a discrete-time signal, which occurs frequently in the analysis of linear discrete-time systems, namely, the geometric sequence.

EXAMPLE 4.15

Find the z-transform of the discrete-time signal

$$u_k = a^k, k = 0, 1, 2, 3, \dots \quad (4.357)$$

Once again, our starting point is the definition of the z-transform in Equation 4.352.

$$U(z) = \sum_{k=0}^{\infty} a^k z^{-k} = \sum_{k=0}^{\infty} (az^{-1})^k = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \quad |z| > |a| \quad (4.358)$$

The result is easily checked by long division, that is, if the denominator in Equation 4.358 is divided into the numerator, the result is

$$\frac{z}{z - a} = 1 + az^{-1} + a^2 z^{-2} + a^3 z^{-3} + \dots + a^k z^{-k} + \dots \quad (4.359)$$

From the definition of $U(z)$ as an infinite series,

$$U(z) = \sum_{k=0}^{\infty} u_k z^{-k} = u_0 + u_1 z^{-1} + u_2 z^{-2} + u_3 z^{-3} + \dots + u_k z^{-k} + \dots \quad (4.360)$$

Comparing Equations 4.359 and 4.360, it follows that $u_0 = 1$, $u_1 = a$, $u_2 = a^2$, $u_3 = a^3$, ..., and, therefore, $u_k = a^k$, $k = 0, 1, 2, 3, \dots$. The long division method provides a quick check on $U(z)$ for a discrete-time signal u_k , $k = 0, 1, 2, 3, \dots$. Typically, the first several coefficients in the infinite series expression for $U(z)$ are compared to the corresponding values of the discrete-time signal u_k with an equivalence necessary (but not sufficient) for $U(z) = z[u_k]$.

Depending on the numerical value of the constant "a," the discrete-time signal u_k in Equation 4.357 can asymptotically approach zero in magnitude ($|a| < 1$), remain constant in magnitude ($|a| = 1$), or increase in magnitude without bound ($|a| > 1$). All six cases are shown in [Figure 4.36](#). Note that when $a = 1$, the discrete-time unit step ([Figure 4.34](#)) results and Equation 4.358 reduces to Equation 4.354.

The exponential sequence in Example 4.14 is also a geometric sequence. This is evident by expressing it in a slightly different way, that is,

$$u_k = e^{-akT} = (e^{-aT})^k = (b)^k, k = 0, 1, 2, 3, \dots \quad \text{where } b = e^{-aT} \quad (4.361)$$

The sequences resulting from uniform sampling of continuous-time sine and cosine functions are fundamental discrete-time signals with z-transforms that follow directly from the basic definition. The results are

$$\sin k\omega T \Leftrightarrow \frac{(\sin \omega T)z}{z^2 - (2\cos \omega T)z + 1} \quad (4.362)$$

$$\cos k\omega T \Leftrightarrow \frac{z(z - \cos \omega T)}{z^2 - (2\cos \omega T)z + 1} \quad (4.363)$$

where the symbol \Leftrightarrow denotes a z-transform pair, that is, a discrete-time signal and its z-transform.

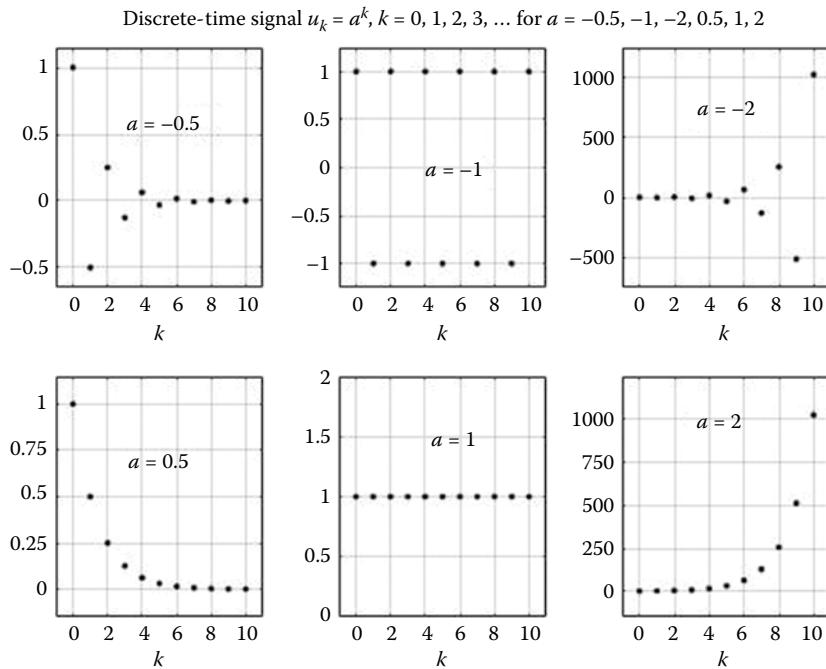


FIGURE 4.36 Discrete-time signal $u_k = a^k, k = 0, 1, 2, 3, \dots$ for $a = -0.5, -1, -2, 0.5, 1, 2$.

The discrete-time signals in Equations 4.362 and 4.363 produce interesting results when the sampling occurs at certain frequencies as shown in Example 4.16.

EXAMPLE 4.16

Find the z-transform of the discrete-time signal obtained from sampling

- a. $x(t) = \sin 3t, t \geq 0$ when $T = \pi/6$ s
- b. $x(t) = \sin 3t, t \geq 0$ when $T = \pi/3$ s
- c. $x(t) = \cos \omega t, t \geq 0$ when $T = 2\pi/\omega$ s

From Equations 4.362 and 4.363,

$$\text{a. } x_k = \sin 3kT \Leftrightarrow \frac{(\sin 3 \cdot \pi/6)z}{z^2 - (2 \cos 3 \cdot \pi/6)z + 1} = \frac{(\sin \pi/2)z}{z^2 - (2 \cos \pi/2)z + 1} = \frac{z}{z^2 + 1} \quad (4.364)$$

$$\text{b. } x_k = \sin 3kT \Leftrightarrow \frac{(\sin 3 \cdot \pi/3)z}{z^2 - (2 \cos 3 \cdot \pi/3)z + 1} = \frac{(\sin \pi)z}{z^2 - (2 \cos \pi)z + 1} = 0 \quad (4.365)$$

$$\begin{aligned} \text{c. } \cos k\omega T &\Leftrightarrow \frac{z(z - \cos \omega \cdot 2\pi/\omega)}{z^2 - (2 \cos \omega \cdot 2\pi/\omega)z + 1} = \frac{z(z - \cos 2\pi)}{z^2 - (2 \cos 2\pi)z + 1} \\ &= \frac{z(z - 1)}{z^2 - 2z + 1} = \frac{z}{z - 1} \end{aligned} \quad (4.366)$$

[Figure 4.37](#) shows the continuous-time signal $x(t) = \sin 3t, t \geq 0$ and the discrete-time signals $x_k = \sin 3kT, k = 0, 1, 2, 3, \dots$ resulting from sampling in parts (a) and (b).

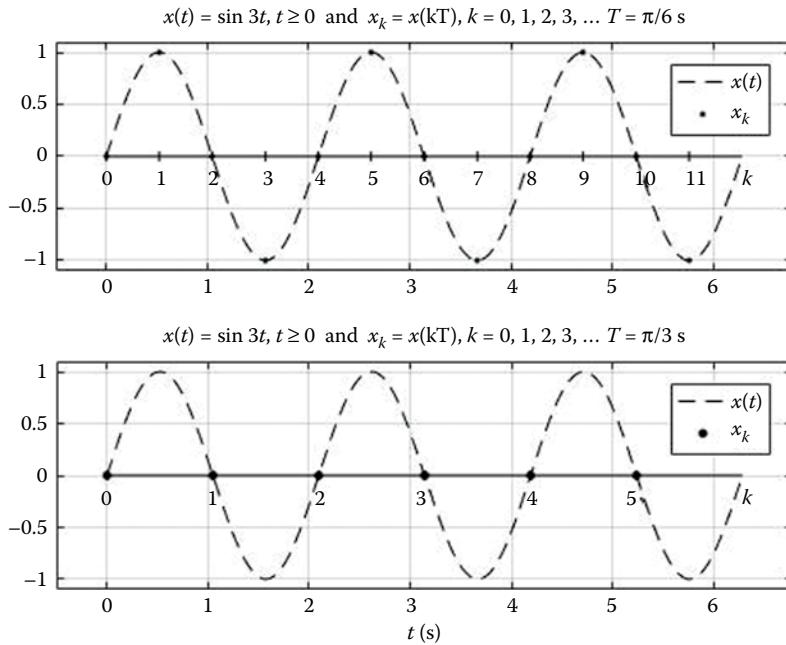


FIGURE 4.37 Uniform sampling of $x(t) = \sin 3t$ ($T = \pi/6$ s and $T = \pi/3$ s).

Note, in part (a), the frequency of sampling $\omega_s = 2\pi/T = 12$ rad/s is four times the frequency of the signal $x(t)$. The result given in Equation 4.364 is easily verified by long division of $z^2 + 1$ into z giving the infinite series

$$U(z) = \frac{z}{z^2 + 1} z^{-1} - z^{-3} + z^{-5} - z^{-7} + z^{-9} - z^{-11} + \dots \quad (4.367)$$

$$\Rightarrow u_k = \begin{cases} 0, & k = 0, 2, 4, 6, \dots \\ 1, & k = 1, 5, 9, \dots \\ -1, & k = 3, 7, 11, \dots \end{cases} = \begin{cases} 0, & k = 0, 2, 4, 6, \dots \\ (-1)^{(k+3)/2}, & k = 1, 3, 5, 7, \dots \end{cases} \quad (4.368)$$

At the slower sampling frequency of 6 rad/s in part (b), the discrete-time signal is identically zero for all k . In part (c), the cosine function is sampled once per cycle resulting in the discrete-time unit step function shown in Figure 4.38.

Table 4.4 is a brief listing of elementary continuous-time functions and their Laplace transforms along with the discrete-time signals resulting from uniform sampling of the continuous-time signals and the corresponding z-transforms (Jacquot 1981).

4.6.1 DISCRETE-TIME IMPULSE FUNCTION

We now introduce a discrete-time function, which plays a prominent role in analyzing the behavior of linear discrete-time systems. The unit strength discrete-time impulse occurring at discrete-time $k = 0$ is defined by

$$\delta_k = \begin{cases} 1, & k = 0 \\ 0, & k = 1, 2, 3, \dots \end{cases} \quad (4.369)$$

$$x(t) = \cos \omega t, t \geq 0 \text{ and } x_k = x(kT), k = 0, 1, 2, 3, \dots T = 2\pi/\omega \text{ s}$$

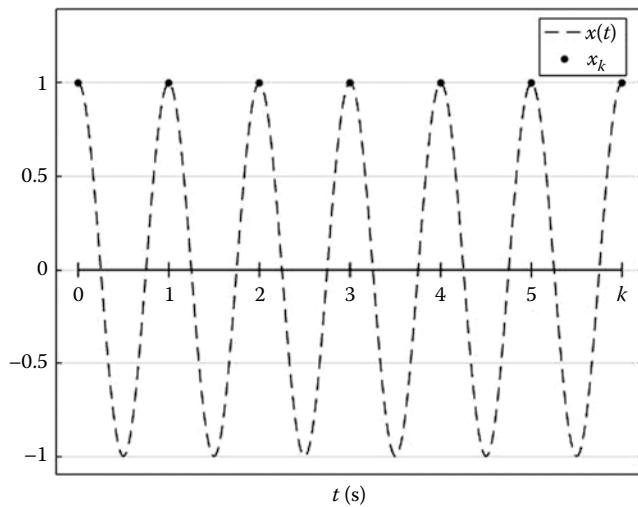


FIGURE 4.38 Uniform sampling of $\cos \omega t$ ($T = 2\pi/\omega$ s).

TABLE 4.4
Table of Laplace and z-Transforms

$f(t), t \geq 0$	$F(s) = L\{f(t)\}$	$f_k = f(kT), k = 0, 1, 2, \dots$	$F(z) = z\{f_k\}$
1	$\frac{1}{s}$	1	$\frac{z}{z-1}$
T	$\frac{1}{s^2}$	KT	$\frac{Tz}{(z-1)^2}$
e^{-at}	$\frac{1}{s+a}$	e^{-akT}	$\frac{z}{z-e^{-aT}}$
te^{-at}	$\frac{1}{(s+a)^2}$	kTe^{-akT}	$\frac{Te^{-aT}z}{(z-e^{-aT})^2}$
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$	$\sin k\omega T$	$\frac{(\sin \omega T)z}{z^2 - 2(\cos \omega T)z + 1}$
$\cos \omega t$	$\frac{s}{s^2 + \omega^2}$	$\cos k\omega T$	$\frac{z^2 - (\cos \omega T)z}{z^2 - 2(\cos \omega T)z + 1}$
$e^{-at} \sin \omega t$	$\frac{\omega}{(s+a)^2 + \omega^2}$	$e^{-akt} \sin k\omega T$	$\frac{(e^{-aT} \sin \omega T)z}{z^2 - 2(e^{-aT} \cos \omega T)z + e^{-2aT}}$
$e^{-at} \cos \omega t$	$\frac{s+a}{(s+a)^2 + \omega^2}$	$e^{-akt} \cos k\omega T$	$\frac{z^2 - (e^{-aT} \cos \omega T)z}{z^2 - 2(e^{-aT} \cos \omega T)z + e^{-2aT}}$

Delaying the discrete-time impulse by n units of discrete-time produces

$$\delta_{k-n} = \begin{cases} 1, & k = n \\ 0, & k = 0, 1, 2, \dots, n-1, n+1, \dots \end{cases} \quad (4.370)$$

It follows directly from the definition of the z -transform that

$$z\{\delta_k\} = 1 \quad \text{and} \quad z\{\delta_{k-n}\} = z^{-n} \quad (4.371)$$

An arbitrary discrete-time signal f_k , $k = 0, 1, 2, \dots$ can be expressed as a weighted sum of unit discrete-time impulses, that is,

$$f_k = \sum_{i=0}^{\infty} f_i \delta_{k-i} = f_0 \delta_k + f_1 \delta_{k-1} + f_2 \delta_{k-2} + f_3 \delta_{k-3} + \dots \quad (4.372)$$

The output of a linear discrete-time system subject to a unit discrete-time impulse is termed the unit impulse response. Just like in the case of continuous-time systems, the discrete-time impulse response reflects the natural dynamics of the system. This will be demonstrated after the z -domain transfer function is introduced.

EXAMPLE 4.17

Represent the discrete-time signal u_k , $k = 0, 1, 2, 3, \dots$ shown in Figure 4.39 in terms of discrete-time impulses and find $U(z)$.

$$u_k = \begin{cases} 0, & k = 0, 1, 2, 6, 7, \dots \\ 1, & k = 3, 5 \\ 2, & k = 4 \end{cases} \quad (4.373)$$

From Equation 4.372,

$$u_k = 1 \cdot \delta_{k-3} + 2 \cdot \delta_{k-4} + 1 \cdot \delta_{k-5} \quad (4.374)$$

$$U(z) = z\{u_k\} = z\{\delta_{k-3} + 2\delta_{k-4} + \delta_{k-5}\} = z^{-3} + 2z^{-4}z^{-5} = \frac{z^2 + 2z + 1}{z^5} \quad (4.375)$$

Note in Equation 4.375 we employed the linearity property of z -transforms, that is,

$$z\{\delta_{k-3} + 2\delta_{k-4} + \delta_{k-5}\} = z\{\delta_{k-3}\} + 2z\{\delta_{k-4}\} + z\{\delta_{k-5}\} \quad (4.376)$$

In the general case,

$$zz\{au_k + by_k\} = \sum_{i=0}^{\infty} \{au_k + by_k\} z^{-k} = a \sum_{k=0}^{\infty} u_k z^{-k} + b \sum_{k=0}^{\infty} y_k z^{-k} = aU(z) + bY(z) \quad (4.377)$$

Other useful properties (analogous to those of the Laplace transform) of the z -transform are included in Table 4.5.

The “delay” property is especially important. Suppose a discrete-time signal u_k for which $u_k = 0$ when $k < 0$ is delayed n units of discrete-time. The delayed signal, denoted u_{k-n} , is expressed in terms of u_k in Table 4.5. The case where $n = 1$ and 2 along with the general case is illustrated in Figure 4.40a through d.

The unit-delay operator, as the name suggests, delays its input by one unit of discrete-time. The symbol for a unit-delay operator is a block with z^{-1} inside. If the input to a unit-delay operator is the discrete-time signal u_k shown in Figure 4.40a, the output would be u_{k-1} in Figure 4.40b. A pair of unit-delay operators in series is shown in Figure 4.41.

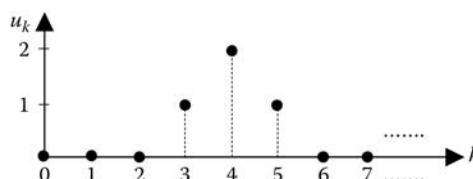


FIGURE 4.39 Graph of discrete-time signal u_k , $k = 0, 1, 2, 3, \dots$

TABLE 4.5
Useful Properties of the z-Transform

Description	Discrete-Time Signal	Property
Linearity	$u_k = ax_k + by_k$	$U(z) = aX(z) + bY(z)$
Delay (right shifting)	given $u_k, k = 0, 1, 2, 3, \dots$, where $u_k = 0$ for $k < 0$	$z\{u_{k-n}\} = z^{-n}U(z)$
	$u_{k-n} = \begin{cases} 0, & k = 0, 1, 2, \dots, n-1 \\ u_0, & k = n \\ u_1, & k = n+1 \\ u_1, & k = n+2 \\ \vdots & \text{etc.} \end{cases}$	
Summation	$y_k = \sum_{i=0}^k u_i$	$Y(z) = \frac{z}{z-1}U(z)$
Multiplication by geometric sequence	$y_k = a^k u_k$	$Y(z) = U\left(\frac{z}{a}\right)$
Multiplication by k property	$y_k = ku_k$	$Y(z) = -z \frac{d}{dz}U(z)$
Initial value property	$f_k = \sum_{i=0}^{\infty} f_i \delta_{k-i}, \quad k = 0, 1, 2, 3, \dots$	$f_0 = \lim_{ z \rightarrow \infty} F(z)$
Final value property	$f_k = \sum_{i=0}^{\infty} f_i \delta_{k-i}, \quad k = 0, 1, 2, 3, \dots$	$f_{\infty} = \lim_{ z \rightarrow 1} (z-1)F(z)$
Periodic signal	$f_k = f_{k+n}, \quad k = 0, 1, 2, 3, \dots$	$F(z) = \frac{z^n}{z^n - 1} \hat{F}(z)$ where $\hat{F}(z) = \sum_{k=0}^{n-1} f_k z^{-k}$

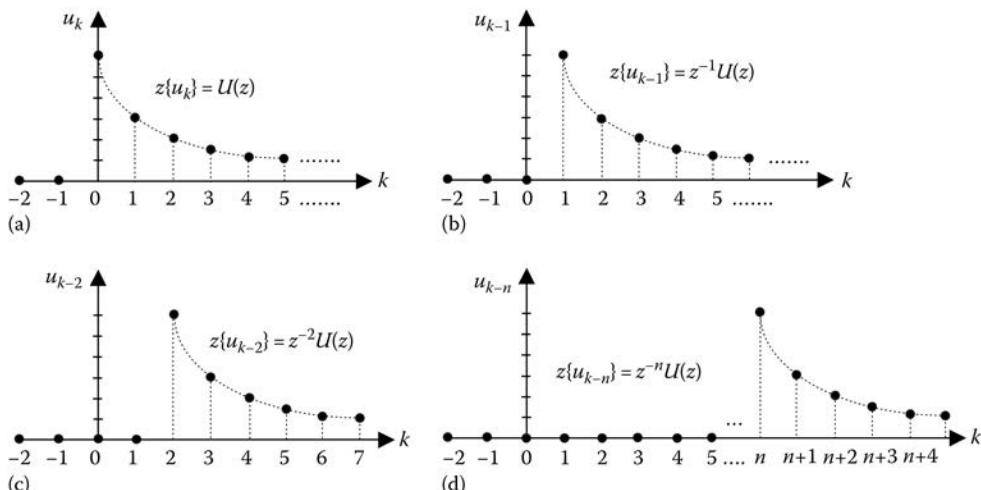


FIGURE 4.40 Illustration of the delay property in Table 4.5.

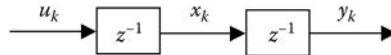


FIGURE 4.41 Unit-delay operators in series.

The outputs x_k and y_k are related to the input u_k by

$$x_k = u_{k-1} = \begin{cases} 0, & k = 0 \\ u_0, & k = 1 \\ u_1, & k = 2 \\ \dots & \dots \end{cases} \quad (4.378)$$

$$y_k = x_{k-1} = u_{k-2} = \begin{cases} 0, & k = 0, 1 \\ u_0, & k = 2 \\ u_1, & k = 3 \\ \dots & \dots \end{cases} \quad (4.379)$$

In a later section when we introduce simulation diagrams for discrete-time systems, it will be apparent that the unit delay is the counterpart to a continuous-time integrator in the simulation diagram of continuous-time systems.

Several examples illustrating the properties in Table 4.5.

EXAMPLE 4.18

A unit alternating sequence ($a = -1$ in Figure 4.36) is the input to a summer as shown in Figure 4.42.

- a. Find the output y_k , $k = 0, 1, 2, 3, \dots$
- b. Find $Y(z)$.
- a. Referring to the graphs of the geometric sequence in Figure 4.36 for the case when $a = -1$, it is apparent that the output of the summer is

$$y_k = \begin{cases} 1, & k = 0, 2, 4, \dots \\ 0, & k = 1, 3, 5, \dots \end{cases} \quad (4.380)$$

- b. From the definition of the z -transform as an infinite series in z^{-1} ,

$$y(z) = 1 + 1 \cdot z^{-2} + 1 \cdot z^{-4} + 1 \cdot z^{-6} + \dots \quad (4.381)$$

$$= 1 + (z^{-2}) + (z^{-2})^2 + (z^{-2})^3 + (z^{-2})^4 + \dots \quad (4.382)$$

$$= \frac{1}{1 - (z^{-2})} \quad (4.383)$$

$$= \frac{z^2}{z^2 - 1} \quad (4.384)$$



FIGURE 4.42 A summer with a unit alternating sequence input.

Alternatively, from the summation property in [Table 4.5](#) and knowing $z\{a^k\} = z/(z - a)$,

$$Y(z) = \frac{z}{z-1} U(z) = \frac{z}{z-1} \left(\frac{z}{z+1} \right) = \frac{z^2}{z^2-1} \quad (4.385)$$

EXAMPLE 4.19

Find the z -transform of the discrete-time signal resulting from sampling the output of a half-wave rectifier whose input is the continuous-time function $\sin \omega_0 t$. Sampling starts at $t = 0$ at a frequency of $8\omega_0$, where $\omega_0 = 2\pi$ rad/s.

The output of the half-wave rectifier is

$$v(t) = \begin{cases} \sin \omega_0 t, & k\pi/\omega_0 \leq t \leq (k+1)\pi/\omega_0 \\ 0, & (k+1)\pi/\omega_0 \leq t \leq (k+2)\pi/\omega_0 \end{cases} \quad \text{for } k = 0, 2, 4, \dots \quad (4.386)$$

Both $v(t)$ and $v_k = v(kT)$, $k = 0, 1, 2, 3, \dots$ are shown in [Figure 4.43](#). The discrete-time signal v_k is periodic, and the period is $n = 8$, that is, $v_{k+8} = v_k$, $k = 0, 1, 2, 3, \dots$

The z -transform of the first cycle of v_k is

$$\hat{V}(z) = \sum_{k=0}^7 v_k z^{-k} = \sum_{k=0}^3 \sin(2\pi k T) z^{-k} = \sum_{k=0}^7 0 \cdot z^{-k} = \sum_{k=0}^3 \sin\left(\frac{k\pi}{4}\right) z^{-k} \quad (4.387)$$

$$= 0 + \sin\left(\frac{\pi}{4}\right) z^{-1} + \sin\left(\frac{\pi}{2}\right) z^{-2} + \sin\left(\frac{3\pi}{4}\right) z^{-3} \quad (4.388)$$

$$= 0 + \frac{\sqrt{2}}{2} z^{-1} + z^{-2} + \frac{\sqrt{2}}{2} z^{-3} \quad (4.389)$$

Sampling the output of a half-wave rectifier with input $\sin 2\pi t$

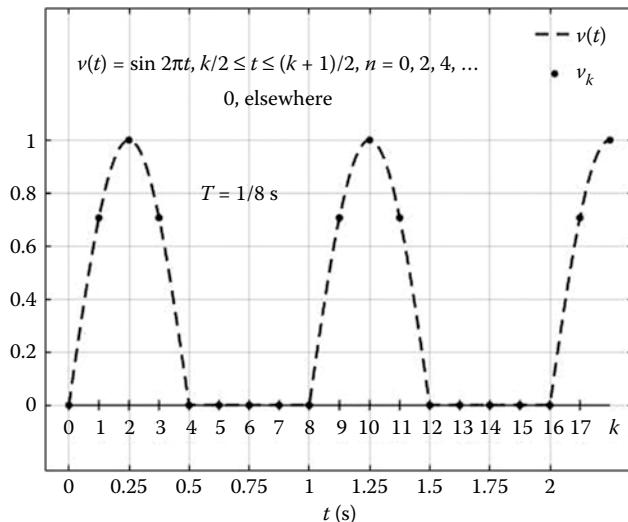


FIGURE 4.43 Sampling the continuous-time output of a half-wave rectifier with input $\sin 2\pi t$.

Applying the property in [Table 4.5](#) for periodic signals gives

$$V(z) = \frac{z^n}{z^n - 1} \hat{V}(z) = \frac{z^8}{z^8 - 1} \left(\frac{\sqrt{2}}{2} z^{-1} + z^{-2} + \frac{\sqrt{2}}{2} z^{-3} \right) \quad (4.390)$$

$$= \frac{z^5}{z^8 - 1} \left(\frac{\sqrt{2}}{2} z^2 + z + \frac{\sqrt{2}}{2} \right) \quad (4.391)$$

Long division of $z^8 - 1$ into $(\sqrt{2}/2)z^7 + z^6 (\sqrt{2}/2)z^5$ will generate a power series in z^{-1} with coefficients corresponding to the sampled values shown in [Figure 4.43](#).

4.6.2 INVERSE z-TRANSFORM

The analysis of discrete-time system dynamics requires the capability of inverting a z -transform $F(z)$ to find the discrete-time signal f_k , $k = 0, 1, 2, 3, \dots$. It is similar to the way in which the inverse Laplace transform was obtained, that is, by exploiting the basic properties of the z -transform, referring to tables of z -transform pairs, partial fraction expansion, and using one additional method not applicable to continuous-time systems, namely, long division. A simple example of finding the inverse z -transform based on some of the methods outlined above follows.

EXAMPLE 4.20

Find the inverse z -transform of

$$F(z) = \frac{z+1}{z(z+2)} \quad (4.392)$$

- a. Using properties of the z -transform along with the lookup table of z -transform pairs.
- b. By the method of long division.

$$a. \quad F(z) = \frac{z+1}{z(z+2)} = z^{-1} \left(\frac{z+1}{z+2} \right) \quad (4.393)$$

$$= z^{-1} \left(\frac{z}{z+2} \right) + z^{-2} \left(\frac{z}{z+2} \right) \quad (4.394)$$

From Equations 4.357 and 4.358, the term $(z/(z+2))$ is the z -transform of the discrete-time signal $g_k = (-2)^k$, $k = 0, 1, 2, 3, \dots$. From the delay property in [Table 4.5](#), f_k is the sum of g_k delayed one unit of time and g_k delayed two units of discrete-time. Denoting the delayed signals by $\tilde{g}_{k,1}$ and $\tilde{g}_{k,2}$, we can write

$$f_k = \tilde{g}_{k,1} + \tilde{g}_{k,2} \quad k = 0, 1, 2, 3, \dots \quad (4.395)$$

where

$$\tilde{g}_{k,1} = \begin{cases} 0 & k = 0 \\ (-2)^{k-1}, & k = 1, 2, 3, \dots \end{cases} \quad (4.396)$$

$$\tilde{g}_{k,2} = \begin{cases} 0 & k = 0, 1 \\ (-2)^{k-2}, & k = 2, 3, 4, \dots \end{cases} \quad (4.397)$$

Combining Equations 4.395 and 4.396, the inverse z -transform is

$$f_k = \begin{cases} 0, & k = 0 \\ 1, & k = 1 \\ (-2)^{k-1} + (-2)^{k-2}, & k = 2, 3, 4, \dots \end{cases} \quad (4.398)$$

Simplifying the expression in Equation 4.398 when $k = 2, 3, 4, \dots$ gives

$$f_k = \begin{cases} 0, & k = 0 \\ 1, & k = 1 \\ -(-2)^{k-2}, & k = 2, 3, 4, \dots \end{cases} \quad (4.399)$$

- c. Long division of the denominator in Equation 4.392 into the numerator results in an infinite series. The first few terms are

$$\frac{z+1}{z^2+2z} = z^{-1} - z^{-2} + 2z^{-3} - 4z^{-4} + 8z^{-5} - \dots \quad (4.400)$$

Looking at Equation 4.400, it is possible to recognize a pattern in the coefficients starting with the z^{-2} term. This pattern results in the expression in the third line of the general solution in Equation 4.399. The reader should verify that Equations 4.399 and 4.400 generate identical values for f_k , $k = 0, 1, 2, 3, \dots$ as they must.

4.6.3 PARTIAL FRACTION EXPANSION

Causal signals, that is, discrete-time signals f_k that are identically zero for negative values of discrete-time k , possess z -transforms of the form

$$F(z) = \frac{N(z)}{D(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n} \quad (n \geq m) \quad (4.401)$$

The partial fraction expansion of $F(z)$ depends on the nature of the roots of $D(z)$. Equation 4.401 is rewritten with the denominator $D(z)$ in factored form,

$$F(z) = \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{(z - p_1)(z - p_2) \cdots (z - p_n)} \quad (n \geq m) \quad (4.402)$$

where p_1, p_2, \dots, p_n are the poles of $F(z)$. Three cases are considered for finding the inverse z -transform of $F(z)$ by partial fractions.

Case I: Poles of $F(z)$ are real and distinct

When the poles p_1, p_2, \dots, p_n are real and unequal, $F(z)$ in partial fraction form is

$$F(z) = c_0 + c_1 \left(\frac{z}{z - p_1} \right) + c_2 \left(\frac{z}{z - p_2} \right) + \dots + c_n \left(\frac{z}{z - p_n} \right) \quad (4.403)$$

The constant c_0 is easily determined by substituting $z = 0$ in Equations 4.402 and 4.403.

$$c_0 = F(z)|_{z=0} \quad F(0) = \begin{cases} 0, & n > m \\ \frac{b_n}{(-p_1)(-p_2) \cdots (-p_n)}, & n = m \end{cases} \quad (4.404)$$

The remaining coefficients c_1, c_2, \dots, c_n are obtained from (Cadzow 1973)

$$c_i = \left(\frac{z - p_i}{z} \right) F(z) \Big|_{z=p_i}, \quad i = 1, 2, 3, \dots, n \quad (4.405)$$

From the z -transform pairs $\delta_k \Leftrightarrow 1$, $a^k \Leftrightarrow z/(z - a)$ and the linearity property of the z -transform, the inverse z -transform of $F(z)$ in Equation 4.403 is

$$f_k = c_0\delta_k + c_1p_1^k + c_2p_2^k + \dots + c_np_n^k, \quad k = 0, 1, 2, 3, \dots \quad (4.406)$$

EXAMPLE 4.21

Find the discrete-time signal with z -transform given by

$$F(z) = \frac{z^2 + z + 1}{z^2 - 4} \quad (4.407)$$

Factoring the denominator leads to the partial fraction expansion

$$\frac{z^2 + z + 1}{z^2 - 4} = \frac{z^2 + z + 1}{(z - 2)(z + 2)} = c_0 + c_1 \left(\frac{z}{z - 2} \right) + c_2 \left(\frac{z}{z + 2} \right) \quad (4.408)$$

where

$$c_0 = F(0) = \frac{z^2 + z + 1}{(z - 2)(z + 2)} \Big|_{z=0} \cdot = -\frac{1}{4} \quad (4.409)$$

$$c_1 = \left(\frac{z - 2}{z} \right) F(z) \Big|_{z=2} = \frac{z^2 + z + 1}{z(z + 2)} \Big|_{z=2} = \frac{7}{8} \quad (4.410)$$

$$c_2 = \left(\frac{z + 2}{z} \right) F(z) \Big|_{z=-2} = \frac{z^2 + z + 1}{z(z - 2)} \Big|_{z=-2} = \frac{3}{8} \quad (4.411)$$

$$\Rightarrow F(z) = -\frac{1}{4} + \frac{7}{8} \left(\frac{z}{z - 2} \right) + \frac{3}{8} \left(\frac{z}{z + 2} \right) \quad (4.412)$$

$$\Rightarrow f_k = -\frac{1}{4}\delta_k + \frac{7}{8}(2)^k + \frac{3}{8}(-2)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.413)$$

It is left as an exercise to show that the first several values of f_k , $k = 0, 1, 2, 3, \dots$ are in agreement with the values obtained by long division of $z^2 - 4$ into $z^2 + z + 1$.

If the denominator $D(z)$ in Equation 4.401 has a factor z^p , the inverse z -transform of $z^p F(z)$ should be determined first, followed by use of the delay property to obtain the final result. To illustrate, suppose $F(z)$ is given by

$$F(z) = \frac{z^2 + z + 1}{z^3(z^2 - 4)} \quad (4.414)$$

We start by inverting $z^3F(z)$,

$$z^3F(z) = \frac{z^2 + z + 1}{z^2 - 4} \quad (4.415)$$

From Example 4.21, we know

$$z^{-1} \left\{ \frac{z^2 + z + 1}{z^2 - 4} \right\} = -\frac{1}{4} \delta_k + \frac{7}{8}(2)^k + \frac{3}{8}(-2)^k \quad (4.416)$$

Hence, the inverse z-transform of $F(z)$ in Equation 4.414 is the discrete-time signal in Equation 4.416 delayed three units of discrete-time, that is,

$$f_k = \begin{cases} 0, & k = 0, 1, 2, \\ -\frac{1}{4}\delta_{k-3} + \frac{7}{8}(2)^{k-3} + \frac{3}{8}(-2)^{k-3}, & k = 3, 4, 5, \dots \end{cases} \quad (4.417)$$

Case II: Repeated real poles of $F(z)$

Suppose the pole p_1 has multiplicity m_1 . The partial fraction expansion contains the m_1 terms

$$c_1 \left(\frac{z}{z - p_1} \right) + \dots + c_{m_1-1} \left(\frac{z}{z - p_1} \right)^{m_1-1} + c_{m_1} \left(\frac{z}{z - p_1} \right)^{m_1} \quad (4.418)$$

associated with the factor $(z - p_1)^{m_1}$ in the denominator of $F(z)$. Simultaneous equations are developed for the constants $c_1, c_2, \dots, c_{m_1-1}$. An illustrative example follows.

EXAMPLE 4.22

Find f_k , $k = 0, 1, 2, 3, \dots$ when

$$F(z) = \frac{2z^2 + z}{(z - 1)^3(z + 1)} \quad (4.419)$$

The partial fraction expansion of $F(z)$ is

$$F(z) = \frac{2z^2 + z}{(z - 1)^3(z + 1)} = c_0 + c_1 \left(\frac{z}{z - 1} \right) + c_2 \left(\frac{z}{z - 1} \right)^2 + c_3 \left(\frac{z}{z - 1} \right)^3 + c_4 \left(\frac{z}{z + 1} \right) \quad (4.420)$$

The constants c_0 and c_4 are obtained as they would in Case I, that is,

$$c_0 = F(0) = 0 \quad (4.421)$$

$$c_4 = \left. \left(\frac{z+1}{z} \right) F(z) \right|_{z=-1} = \left. \frac{2z+1}{(z-1)^3} \right|_{z=-1} = \frac{1}{8} \quad (4.422)$$

The coefficient of the highest order term is evaluated directly from

$$c_3 = \left. \left(\frac{z-1}{z} \right)^3 F(z) \right|_{z=1} = \left. \frac{2z+1}{z^2(z+1)} \right|_{z=1} = \frac{3}{2} \quad (4.423)$$

Substituting the values for c_0 , c_3 , and c_4 into Equation 4.420 yields

$$F(z) = \frac{2z^2 + z}{(z-1)^3(z+1)} = c_1 \left(\frac{z}{z-1} \right) + c_2 \left(\frac{z}{z-1} \right)^2 + \frac{3}{2} \left(\frac{z}{z-1} \right)^3 + \frac{1}{8} \left(\frac{z}{z-1} \right) \quad (4.424)$$

Combining the terms on the right-hand side of Equation 4.424 into a single term with common denominator $(z-1)^3(z+1)$ and then equating the numerators give

$$2z^2 + z = c_1 z(z-1)^2(z+1) + c_2 z^2(z-1)(z+1) + \frac{3}{2} z^3(z+1) + \frac{1}{8} z(z-1)^3 \quad (4.425)$$

Expanding the right-hand side of Equation 4.425 and equating coefficients of like powers of z on both sides lead to

$$\begin{cases} z^4 : 0 = c_1 + c_2 + \frac{3}{2} + \frac{1}{8} \\ z^3 : 0 = -c_1 + \frac{3}{2} + \frac{3}{8} \\ z^2 : 2 = -c_1 - c_2 + \frac{3}{8} \\ z : 1 = c_1 - \frac{1}{8} \end{cases} \quad (4.426)$$

Selecting two of the above equations for simultaneous solution results in $c_1 = 9/8$ and $c_2 = -11/4$. Substituting the known values for c_1 and c_2 in Equation 4.424 gives

$$F(z) = \frac{9}{8} \left(\frac{z}{z-1} \right) - \frac{11}{4} \left(\frac{z}{z-1} \right)^2 + \frac{3}{2} \left(\frac{z}{z-1} \right)^3 + \frac{1}{8} \left(\frac{z}{z+1} \right) \quad (4.427)$$

Inverting $F(z)$ is accomplished using [Table 4.6](#) (Cadzow 1973)

$$f_k = \frac{9}{8} - \frac{11}{4}(k+1) + \frac{3}{2} \left[\frac{(k+1)(k+2)}{2} \right] + \frac{1}{8}(-1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.428)$$

TABLE 4.6
Table for Inverting z -Transforms of the Form
 $[z/(z-a)]^n, n = 1, 2, 3, \dots$

$F(z) \qquad f_k, k = 0, 1, 2, 3, \dots$

$\frac{z}{(z-a)}$	a^k
$\left[\frac{z}{(z-a)} \right]^2$	$(k+1)a^k$
$\left[\frac{z}{(z-a)} \right]^3$	$\frac{(k+1)(k+2)}{2}a^k$
$\left[\frac{z}{(z-a)} \right]^4$	$\frac{(k+1)(k+2)(k+3)}{3}a^k$
$\left[\frac{z}{(z-a)} \right]^5$	$\frac{(k+1)(k+2)(k+3)(k+4)}{4}a^k$

Evaluating the first several values of f_k gives

$$f_0 = \frac{9}{8} - \frac{11}{4} + \frac{3}{2} + \frac{1}{8} = 0, \quad f_1 = \frac{9}{8} - \frac{22}{4} + \frac{18}{2} + \frac{1}{8} = 0, \quad f_2 = \frac{9}{8} - \frac{33}{4} + \frac{36}{2} + \frac{1}{8} = 2$$

$$f_3 = \frac{9}{8} - \frac{44}{4} + \frac{60}{4} - \frac{1}{8} = 5, \quad f_4 = \frac{9}{8} - \frac{55}{4} + \frac{90}{2} + \frac{1}{8} = 10, \quad f_5 = \frac{9}{8} - \frac{66}{4} + \frac{126}{4} - \frac{1}{8} = 16$$

Checking the above by long division confirms the numerical values above.

$$\begin{array}{r} 2z^{-2} + 5z^{-3} + 10z^{-4} + 16z^{-5} \\ z^4 - 2z^3 + 0z^2 + 2z - 1 | 2z^2 + z \\ \underline{2z^2 - 4z + 0 + 4z^{-1} - 2z^{-2}} \\ 5z + 0 - 4z^{-1} + 2z^{-2} \\ \underline{5z - 10 + 0z^{-1} + 10z^{-2} - 5z^{-3}} \\ 10 - 4z^{-1} - 8z^{-2} + 5z^{-3} \\ \underline{10 - 20z^{-1} - 0z^{-1} + 20z^{-3} - 10z^{-4}} \\ 16z^{-1} - 8z^{-2} - 15z^{-3} + 10z^{-4} \end{array}$$

Case III: Complex poles of $F(z)$

When $F(z)$ possesses complex poles, the partial fraction expansion is dictated by the last two z -transform pairs in [Table 4.4](#). An example serves to illustrate the procedure.

$$F(z) = \frac{z^2 + z}{(z+1)(z^2 - 3z + 9)} \quad (4.429)$$

The first step is to decompose $F(z)$ in two parts,

$$F(z) = \frac{Az^2 + Bz}{(z^2 - 3z + 9)} + C\left(\frac{z}{z-1}\right) \quad (4.430)$$

The constant C is evaluated from

$$C = \left. \left(\frac{z-1}{z} \right) \frac{z^2 + z}{(z-1)(z^2 - 3z + 9)} \right|_{z=1} = \left. \frac{z+1}{z^2 - 3z + 9} \right|_{z=1} \frac{2}{7} \quad (4.431)$$

Constants A and B are obtained by combining the terms in Equation 4.430 into a single term with common denominator $(z-1)(z^2 - 3z + 9)$ and then equating the numerator to $z^2 + z$ the numerator in Equation 4.429. The resulting expression for $F(z)$ is

$$F(z) = \frac{-(2/7)z^2 + (11/7)z}{z^2 - 3z + 9} + \frac{2}{7} \left(\frac{z}{z-1} \right) \quad (4.432)$$

$$= -\frac{1}{7} \left(\frac{2z^2 - 11z}{z^2 - 3z + 9} \right) + \frac{2}{7} \left(\frac{z}{z-1} \right) \quad (4.433)$$

The quadratic factor in the denominator of Equation 4.433 implies that inverting $F(z)$ will require a linear combination of $e^{-akT} \sin k\omega T$ and $e^{-akT} \cos k\omega T$ (see [Table 4.4](#)). Comparing the

standard form of the denominator in the last row of [Table 4.4](#) and the quadratic denominator in Equation 4.433,

$$z^2 - 2(e^{-\alpha T} \cos \omega T)z + e^{-2\alpha T} = z^{-2} - 3z + 9 \quad (4.434)$$

Equating like powers of z and solving for $e^{-\alpha T}$ and ωT ,

$$e^{-2\alpha T} = 9 \Rightarrow e^{-\alpha T} = 3 \quad (4.435)$$

$$-2(e^{-\alpha T} \cos \omega T) = -3 \Rightarrow \cos(\omega T) = \frac{1}{2} \Rightarrow \omega T = \frac{\pi}{3} \quad (4.436)$$

The quadratic numerator in $F(z)$ in Equation 4.433 must be expressed as a linear combination of the standard numerator forms in the last two rows of [Table 4.4](#), that is,

$$2z^2 - 11z = c_1(e^{-\alpha T} \sin \omega T)z + c_2[z^2 - (e^{-\alpha T} \cos \omega T)z] \quad (4.437)$$

Solving for c_1 and c_2 in Equation 4.437 leads to $c_1 = -16\sqrt{3}/9$, $c_2 = 2$. $F(z)$ is now written in a form where [Table 4.4](#) can be used to find f_k .

$$F(z) = -\frac{1}{7} \left\{ \frac{c_1(e^{-\alpha T} \sin \omega T)z}{z^2 - 2(e^{-\alpha T} \cos \omega T)z + e^{-2\alpha T}} + \frac{c_2(z^2 - (e^{-\alpha T} \cos \omega T)z)}{z^2 - 2(e^{-\alpha T} \cos \omega T)z + e^{-2\alpha T}} \right\} + \frac{2}{7} \left(\frac{z}{z-1} \right) \quad (4.438)$$

$$f(k) = \frac{1}{7} [c_1 e^{-\alpha k T} \sin k \omega T + c_2 e^{-\alpha T} \cos k \omega T] + \frac{2}{7}, \quad k = 0, 1, 2, 3, \dots \quad (4.439)$$

$$= -\frac{1}{7} \left[\frac{-16\sqrt{3}}{9} (3)^k \sin \left(\frac{k\pi}{3} \right) + 2(3)^k \cos \left(\frac{k\pi}{3} \right) \right] + \frac{2}{7}, \quad k = 0, 1, 2, 3, \dots \quad (4.440)$$

By observation of $F(z)$ in Equation 4.429, it follows that $f_0 = 0$ and $f_1 = 1$. The reader can readily verify these values from Equation 4.440 with $k = 0, 1$.

An alternative approach when $F(z)$ contains complex poles is to proceed the same way as in Case 1 where all the poles were real and distinct. The key is appropriate conversion between rectangular and polar representations of the complex roots of $F(z)$ and the complex coefficients arising from partial fraction expansion.

Suppose $F(z)$ is of the form

$$F(z) = \frac{N(z)}{D(z)} = \frac{N(z)}{(z - p_1)(z - p_2)} \quad (4.441)$$

where the complex poles expressed in polar form are $p_1 = Re^{j\theta}$, $p_2 = Re^{-j\theta}$.

Expanding $F(z)$ as we did in Case 1 (real and distinct poles),

$$F(z) = A_1 \left(\frac{z}{z - p_1} \right) + A_2 \left(\frac{z}{z - p_2} \right) \quad (4.442)$$

where

$$A_1 = \frac{(z - p_1)}{z} \left[\frac{N(z)}{(z - p_1)(z - p_2)} \right]_{z=p_1} = \frac{N(p_1)}{p_1(p_1 - p_2)} \quad (4.443)$$

and A_2 is the conjugate of A_1 . In polar form, $A_1 = Ce^{j\phi}$, $A_2 = Ce^{-j\phi}$ Equation 4.442 becomes

$$F(z) = Ce^{j\phi} \left(\frac{z}{z - Re^{j\theta}} \right) + Ce^{-j\phi} \left(\frac{z}{z - Re^{-j\theta}} \right) \quad (4.444)$$

The inverse z -transform of $F(z)$ in Equation 4.444 is

$$f_k = Ce^{j\phi} (Re^{j\theta})^k + Ce^{-j\phi} (Re^{-j\theta})^k \quad (4.445)$$

$$= CR^k [e^{j\phi} (e^{j\theta})^k + e^{-j\phi} (e^{-j\theta})^k] \quad (4.446)$$

$$= 2CR^k \left[\frac{e^{j(k\theta+\phi)} + e^{j(k\theta-\phi)}}{2} \right] \quad (4.447)$$

$$= 2CR^k \cos(k\theta + \phi), \quad k = 0, 1, 2, \dots \quad (4.448)$$

Thus, $F(z)$ in Equation 4.441 can be inverted simply by finding polar coordinates of the poles p_1 , p_2 , and complex coefficients A_1 , A_2 .

EXAMPLE 4.23

Find the inverse z -transform of $F(z) = z^2 - z/z^2 - 0.6z + 0.25$.

Factoring the denominator to find the poles p_1 and p_2 ,

$$F(z) = \frac{z^2 - z}{z^2 - 0.6z + 0.25} = \frac{z^2 - z}{(z - p_1)(z - p_2)}, \quad p_{1,2} = 0.3 \pm j0.4 \quad (4.449)$$

Converting the complex poles to polar form gives

$$p_{1,2} = 0.3 \pm j0.4 = Re^{\pm j\theta} \quad (4.450)$$

where

$$R = [(0.3)^2 + (0.4)^2]^{1/2} = 0.5$$

$$\theta = \tan^{-1} \left(\frac{4}{3} \right) = 0.9273 \text{ rad}$$

From Equation 4.443, the constant A_1 in the partial fraction expansion of $F(z)$ is

$$\begin{aligned} A_1 &= \frac{N(p_1)}{p_1(p_1 - p_2)} = \frac{p_1^2 - p_1}{p_1(p_1 - p_2)} = \frac{p_1 - 1}{p_1 - p_2} \\ &= \frac{0.3 + j0.4 - 1}{0.3 + j0.4 - (0.3 - j0.4)} \\ &= \frac{1}{2} + j\frac{7}{8} \end{aligned} \quad (4.451)$$

Converting A_1 to polar form,

$$C = |A_1| = \left| \frac{1}{2} + j \frac{7}{8} \right| = \frac{\sqrt{65}}{8} \quad (4.452)$$

$$\phi = \text{Arg}(A_1) = \text{Arg}\left(\frac{1}{2} + j \frac{7}{8}\right) = \tan^{-1}\left(\frac{7}{4}\right) = 1.0517 \text{ rad} \quad (4.453)$$

From Equation 4.448, the discrete-time signal f_k is

$$f_k = 2 \frac{\sqrt{65}}{8} (0.5)^k \cos(0.9273k + 1.0517), \quad k = 0, 1, 2, \dots \quad (4.454)$$

$$= 2.0156(0.5)^k \cos(0.9273k + 1.0517), \quad k = 0, 1, 2, \dots \quad (4.455)$$

The reader should check that the first several values of f_k obtained from Equation 4.455 agree with the numerical values obtained by long division of the denominator $z^2 - 0.6z + 0.25$ of $F(z)$ into the numerator $z^2 - z$.

EXERCISES

4.36 Find the z -transforms of the following causal sequences f_k , $k = 0, 1, 2, 3, \dots$. Use long division to check the first two nonzero values of f_k .

$$(a) ka^k \quad (b) k^2(-1)^k \quad (c) \delta_k + (0.5)^k \quad (d) \sin k\pi \quad (e) (-1)^k \cos(2k\pi/3)$$

$$(f) (k+1)\delta_{k-1} \quad (g) f_k = \begin{cases} 0, & k = 0, 2, 4, 6, \dots \\ 1, & k = 1, 3, 5, \dots \end{cases} \quad (h) f_k = \begin{cases} 0, & k = 0, 1, 2, 3, \dots \\ k, & k = 4, 5, 6, \dots \end{cases}$$

$$(i) f_k = \begin{cases} 1, & k = 0 \\ 0, & k = 0, 1, 3, 5, 7, \dots \\ \frac{1}{2}(-2)^{k/2}, & k = 2, 4, 6, \dots \end{cases}$$

4.37 Find the inverse z -transform of the expressions below. Use long division to check the first two nonzero values of f_k .

$$(a) \frac{z+a}{z+b} \quad (b) \frac{z^2+1}{z^2(z^2-1)} \quad (c) \frac{z^2}{(z-3)^3} \quad (d) \frac{z^2+1}{(z+1)^2}$$

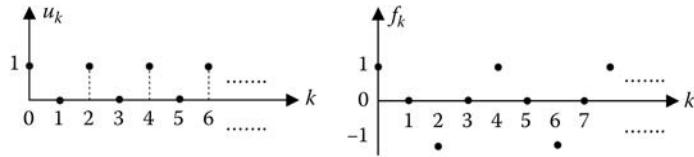
$$(e) \frac{z^3+z}{(z^2-1)^2} \quad (f) \frac{z^2+1}{z^3+z^2} \quad (g) \frac{z+2}{z^2-z+4} \quad (h) \frac{z(z-2)}{z^2-z+(3/4)}$$

$$(i) \frac{z^4}{z^4-1} \quad (j) \frac{z^2+1}{z^2+2} \quad (k) \frac{z+1}{z(z^2+z+2)}$$

4.38 Find the z -transforms of the discrete-time signals resulting from uniform sampling of the continuous-time functions below. All functions are zero for $t < 0$.

- (a) $1 + 2t$
- (b) te^{-2t}
- (c) $e^{-at} - e^{-bt}$
- (d) $t^2 \sin 2t$
- (e) $e^{-2t} \cos t$
- (f) $1/2^t$

4.39 a. Find the z -transforms $U(z)$ and $F(z)$ of the discrete-time signals pictured in Figure E4.39.

**FIGURE E4.39**

- b. Express the signal u_k in [below figure](#) as $u_k = \alpha + \beta(-1)^k, k = 0, 1, 2, 3, \dots$ and determine α and β . Use the linearity property and the z -transforms of the unit step and unit alternating sequence, that is, $z\{\hat{u}k\} = z/(z - 1)$ and $z\{(-1)^k\} = z/(z + 1)$, to find $U(z)$.
- 4.40 Graph the discrete-time signals $f_k, k = 0, 1, 2, 3, \dots$ below and find $F(z)$.

a. $f_k = \sum_{i=0}^{\infty} \delta_{k-i}$

b. $f_k = \sum_{i=0}^k 1$

c. $f_k = \sum_{i=0}^k i$

d. $f_k = \sum_{i=0}^k (-1)^i$

- 4.41 Use the two methods discussed for inverting z -transforms with complex poles to find the discrete-time signal $f_k, k = 0, 1, 2, \dots$ with z -transform

$$F(z) = \frac{3z^2 + z}{(z - 1)(z^2 + 2z + 2)}$$

- 4.42 Write a MATLAB function to invert

$$F(z) = \frac{b_3z^2 + b_2z^2 + b_1z + b_0}{(z - p_1)(z^2 + a_1z + a_0)}$$

where the quadratic term $z^2 + a_1z + a_0 = (z - p_2)(z - p_3), p_2, p_3 = \alpha \pm j\beta$.

The inverse z -transform is given by

$$f_k = F_0\delta_k + A_1(p_1)^k + 2CR^k \cos(k\theta + \phi), \quad k = 0, 1, 2, 3, \dots$$

The function input parameters are $p_1, a_1, a_0, b_3, b_2, b_1$, and b_0 and the outputs are A_1, C, R, θ , and F_0 . The function declaration line is

```
[A1, C, R, theta, phi, F0] = invert(p1, a1, a0, b3, b2, b1, b0)
```

Check the function by running it for

i. $F(z) = \frac{3z^2 + z}{(z - 1)(z^2 + 2z + 2)}$

ii. $F(z) = \frac{2z^3 + z^2 + 4z + 5}{(z - 3)(z^2 + 2z + 4)}$

and comparing the first several values of $f_k, k = 0, 1, 2, \dots$ with the values obtained by long division of the cubic denominator into the quadratic numerator.

4.7 Z-DOMAIN TRANSFER FUNCTION

We have seen how the transfer function of a linear continuous-time system is used to find the system's response to elementary inputs. Stability and frequency response characteristics of the system can be inferred from the transfer function as well. A discrete-time system transfer function does the same for linear discrete-time systems. We begin with the n th-order, linear, constant coefficient difference equation

$$y_k + a_1 y_{k-1} + \cdots + a_n y_{k-n} = b_0 u_{k-1} + \cdots + b_m u_{k-m}, \quad n \geq m \quad (4.456)$$

z -Transforming both sides and applying the linearity property gives

$$z\{y_k\} + a_1 z\{y_{k-1}\} + \cdots + a_n z\{y_{k-n}\} = b_0 z\{u_k\} + b_1 z\{u_{k-1}\} + \cdots + b_m z\{u_{k-m}\} \quad (4.457)$$

Assuming the input is applied at $k = 0$ and the initial values $y_{-1}, y_{-2}, \dots, y_{-n}$ are zero, we can use the delay property in [Table 4.5](#) in the previous section to arrive at

$$Y(z) + a_1 z^{-1} Y(z) + \cdots + a_n z^{-n} Y(z) = b_0 U(z) + b_1 z^{-1} U(z) + \cdots + b_m z^{-m} U(z) \quad (4.458)$$

The z -domain transfer function is defined as the ratio of $Y(z)$ to $U(z)$. Thus,

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n}}, \quad n \geq m \quad (4.459)$$

$$= \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n}, \quad n \geq m \quad (4.460)$$

Depending on the application, one of the two forms given in Equations 4.459 and 4.460 for the transfer function, also called the pulse transfer function, is usually preferable. A good example to illustrate how to find the z -domain transfer function of a discrete-time system is an Euler integrator. Recall from Section 3.3 that the difference equation for approximating a continuous-time integrator using explicit Euler integration is

$$x_A(n+1) = x_A(n) + T u(n), \quad n = 0, 1, 2, \dots \quad (4.461)$$

where $u(n)$ and $x_A(n)$ are the discrete-time input and outputs and $x_A(0) = 0$. Employing the notation of this chapter, the difference equation is written

$$x_k - x_{k-1} = T u_{k-1}, \quad k = 1, 2, 3, \dots \quad (4.462)$$

where $u_k = u(kT)$, $k = 0, 1, 2, \dots$ are sampled values of the input signal and x_k , $k = 0, 1, 2, \dots$ is the discrete-time output intended to approximate the continuous-time integrator output $x(t)$ at the end of each integration step. The initial condition is $x_0 = x(0) = 0$ and the first computed value is x_1 .

z -transforming Equation 4.462,

$$z\{x_k\} - z\{x_{k-1}\} = T z\{u_{k-1}\} \quad (4.463)$$

Since x_k and u_k are both zero for $k < 0$, the delay property in [Table 4.5](#) applies.

$$X(z) - z^{-1}X(z) = Tz^{-1}U(z) \quad (4.464)$$

$$H(z) = \frac{X(z)}{U(z)} = \frac{Tz^{-1}}{1 - z^{-1}} = \frac{T}{z - 1} \quad (4.465)$$

EXAMPLE 4.24

The input to a continuous-time integrator is $u(t) = \sin \pi t$.

- a. Approximate the output $x(t)$ using Euler integration with step size $T = 0.1$ s.
- b. Find the exact solution $x(t)$ and plot on the same graph with x_k .
- a. Solving for $X(z)$ in [Equation 4.465](#) and looking up $U(z) = z\{u_k\} = z\{\sin k\omega T\}$ from [Table 4.4](#) give

$$X(z) = H(z)U(z) = \frac{T}{z - 1} \left[\frac{(\sin \omega T)z}{z^2 - 2(\cos \omega T)z + 1} \right]_{T=0.1, \omega=\pi} \quad (4.466)$$

$$= \frac{0.1}{z - 1} \left[\frac{(\sin 0.1\pi)z}{z^2 - 2(\cos 0.1\pi)z + 1} \right] \quad (4.467)$$

Using the method of partial fraction expansion presented in [Section 4.6.3](#), the inverse z -transform of $X(z)$ is (details are left as an exercise)

$$x_k = 0.05 \left[\frac{\sin 0.1\pi}{1 - \cos 0.1\pi} (1 - \cos 0.1k\pi) - \sin 0.1k\pi \right], \quad k = 0, 1, 2, 3, \dots \quad (4.468)$$

- b. The continuous-time integrator output is obtained by integration of the input $u(t)$,

$$x(t) = \int_0^t u(\lambda) d\lambda = \int_0^t \sin \pi \lambda d\lambda = \frac{1}{\pi} (1 - \cos \pi t) \quad (4.469)$$

The discrete-time signal x_k and the continuous-time integrator output $x(t)$ are plotted in [Figure 4.44](#) for one cycle of the input.

4.7.1 NONZERO INITIAL CONDITIONS

Using the z -transform to solve a difference equation with nonzero initial conditions requires additional terms to account for the nonzero values. Suppose y_k is a discrete-time signal for which $y_{-1} \neq 0$ like the one shown in [Figure 4.45](#). Also shown is y_{k-1} .

$z\{y_{k-1}\}$ is obtained from the basic definition of the z -transform, that is,

$$z\{y_{k-1}\} = \sum_{k=0}^{\infty} y_{k-1} z^{-k} = y_{-1} + y_0 z^{-1} + y_1 z^{-2} + y_2 z^{-3} + \dots \quad (4.470)$$

$$= y_{-1} + z^{-1}[y_0 + y_1 z^{-1} + y_2 z^{-2} + \dots] \quad (4.471)$$

$$= y_{-1} + z^{-1}Y(z) \quad (4.472)$$

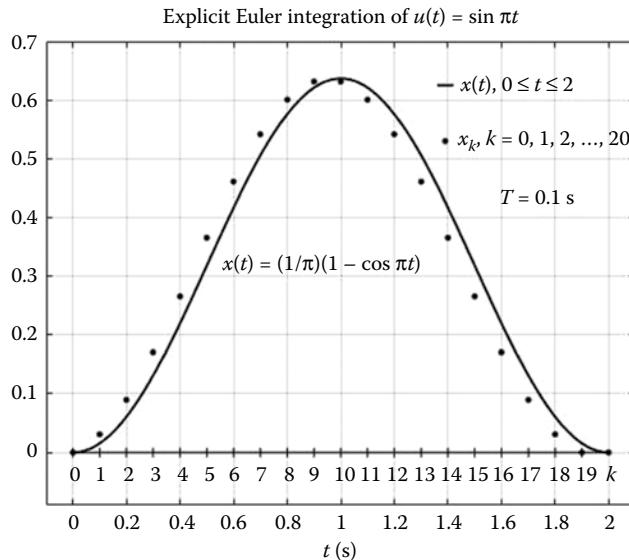


FIGURE 4.44 Continuous- and discrete-time (explicit Euler) integrator outputs.

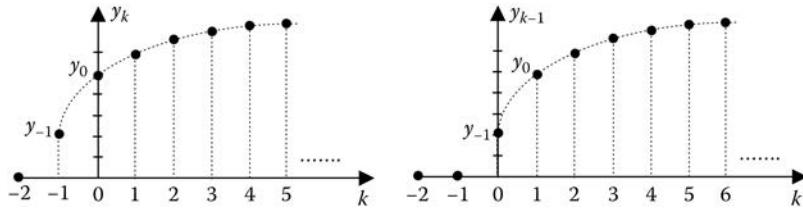


FIGURE 4.45 Discrete-time signals y_k and y_{k-1} ($y_{-1} \neq 0$).

Consider the first-order difference equation

$$y_k = \beta u_k + \alpha y_{k-1}, \quad k = 0, 1, 2, 3, \dots \quad (4.473)$$

with input u_k applied at $k = 0$ and nonzero initial condition y_{-1} . It will be shown later that Equation 4.473 is the difference equation of a low-pass digital filter. z -transforming Equation 4.473 and using the result in Equation 4.472 give

$$Y(z) = \beta U(z) + \alpha [y_{-1} + z^{-1} Y(z)] \quad (4.474)$$

Multiplying Equation 4.474 by z and solving for $Y(z)$ give

$$Y(z) = \frac{\beta z}{z - \alpha} U(z) + \alpha y_{-1} \left(\frac{z}{z - \alpha} \right) \quad (4.475)$$

The first term on the right-hand side of Equation 4.475 is $H(z) U(z)$. The additional term results from the nonzero initial condition. A similar procedure is employed for higher order difference equations with several nonzero initial conditions.

EXAMPLE 4.25

For the discrete-time system described by Equation 4.473,

- Find the response to a unit step when the initial condition $y_{-1} \neq 0$.
- Find the response to a unit alternating input when $y_{-1} \neq 0$.
- For $\alpha = 0.9$ and $\beta = 0.1$, graph the responses in parts (a) and (b) when $y_{-1} = 2$.

- a. $u_k = 1, k = 0, 1, 2, 3, \dots$, and $U(z) = z/(z - 1)$.

$$Y(z) = \frac{\beta z}{z - \alpha} \left(\frac{z}{z - 1} \right) + \alpha y_{-1} \left(\frac{z}{z - \alpha} \right) \quad (4.476)$$

$$= \frac{\beta}{1 - \alpha} \left[\frac{z}{z - 1} - \alpha \frac{z}{z - \alpha} \right] + \alpha y_{-1} \left(\frac{z}{z - \alpha} \right) \quad (4.477)$$

$$y_k = \frac{\beta}{1 - \alpha} (1 - \alpha^{k+1}) + y_{-1} \alpha^{k+1}, \quad k = 0, 1, 2, \dots \quad (4.478)$$

- b. $u_k = (-1)^k, k = 0, 1, 2, 3, \dots$, and $U(z) = z/(z + 1)$.

$$Y(z) = \frac{\beta z}{z - \alpha} \left(\frac{z}{z + 1} \right) + \alpha y_{-1} \left(\frac{z}{z - \alpha} \right) \quad (4.479)$$

$$= \frac{\beta}{1 - \alpha} \left[\frac{z}{z + 1} - \alpha \frac{z}{z - \alpha} \right] + \alpha y_{-1} \left(\frac{z}{z - \alpha} \right) \quad (4.480)$$

$$y_k = \frac{\beta}{1 + \alpha} [(-1)^k + \alpha^{k+1}] + y_{-1} \alpha^{k+1}, \quad k = 0, 1, 2, \dots \quad (4.481)$$

Note that the solutions in Equations 4.478 and 4.481 reduce to the given initial condition y_{-1} for $k = -1$.

- c. Graphs of $y_k, k = -1, 0, 1, 2, \dots$ in Equations 4.478 and 4.481 are shown in [Figure 4.46](#).

Note how the system passes the low-frequency unit step and effectively blocks the higher frequency unit alternating sequence once the transient component α^{k+1} dies out. Setting $\beta = 1 - \alpha$, the normalized unit step and unit alternating steady-state responses are

$$(yk)_{ss} = 1 \quad \text{for } u_k = 1, \quad k = 0, 1, 2, 3, \dots \quad (4.482)$$

$$(yk)_{ss} = \frac{1 - \alpha}{1 + \alpha} (-1)^k = \frac{1 - 0.9}{1 - 0.9} (-1)^k = \frac{1}{19} (-1)^k \quad \text{for } u_k = (-1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.483)$$

4.7.2 APPROXIMATING CONTINUOUS-TIME SYSTEM TRANSFER FUNCTIONS

It is common practice to start with a block diagram representation of a continuous-time system and transform it to a block diagram of a discrete-time system with comparable dynamics. The discrete-time signals are intended to approximate the corresponding signals in the continuous-time system

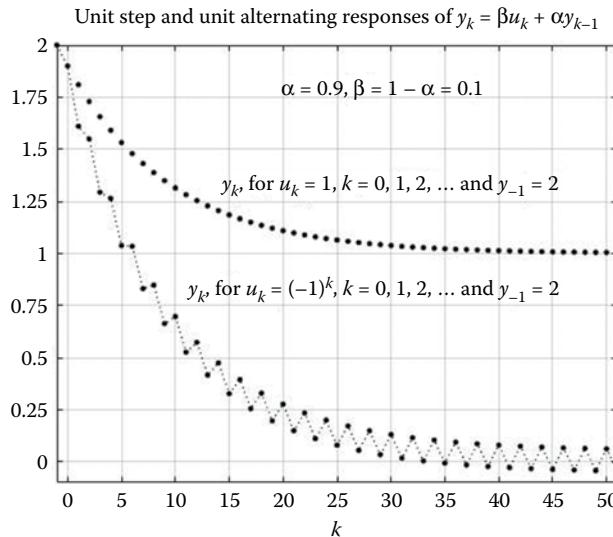


FIGURE 4.46 Responses of discrete-time system with nonzero initial condition.

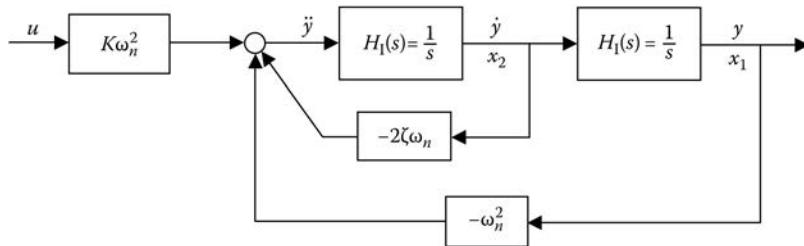


FIGURE 4.47 Simulation diagram for second-order system in Equation 4.484.

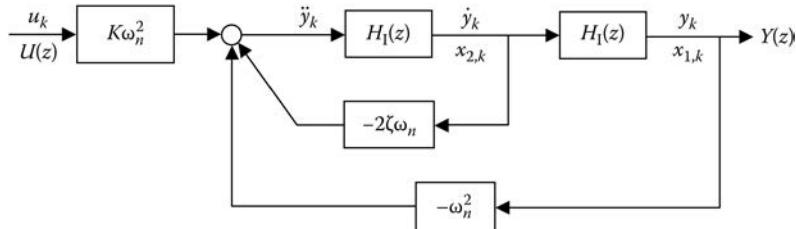


FIGURE 4.48 Discrete-time system with numerical integrator z -domain transfer functions.

at discrete points in time. To illustrate, suppose we have a need to approximate the behavior of a second-order system

$$H(s) = \frac{Y(s)}{U(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.484)$$

A simulation diagram is shown in [Figure 4.47](#).

The continuous-time integrator blocks with transfer function $H_l(s) = 1/s$ are replaced by discrete-time (numerical) integrators with z -domain transfer functions $H_l(z)$, and the signals become discrete-time in nature (see [Figure 4.48](#)).

Block diagram reduction or any other suitable method, for example, Mason's Gain Formula (Dorf and Bishop 2005), using signal flow graphs or solution of simultaneous equations results in the pulse transfer function of the discrete-time system given in Equation 4.485.

$$H(z) = \frac{Y(z)}{U(z)} = \frac{K\omega_n^2 H_I^2(z)}{\omega_n^2 H_I^2(z) + 2\zeta\omega_n H_I(z) + 1} \quad (4.485)$$

Choosing $H_I(z)$ as the z -domain transfer function for an explicit Euler integrator (see Equation 4.465) gives

$$H(z) = \frac{Y(z)}{U(z)} = \frac{K\omega_n^2(T/(z-1))^2}{\omega_n^2(T/(z-1))^2 + 2\zeta\omega_n(T/(z-1)) + 1} \quad (4.486)$$

Simplifying the above expression yields

$$H(z) = \frac{Y(z)}{U(z)} = \frac{K(\omega_n T)^2}{z^2 - 2(1 - \zeta\omega_n T)z + 1 - 2\zeta\omega_n T + (\omega_n T)^2} \quad (4.487)$$

The difference equation for the discrete-time system is obtained directly from the z -domain transfer function expressed in terms of negative power of z .

$$\frac{Y(z)}{U(z)} = \frac{K(\omega_n T)^2 z^{-2}}{1 - 2(1 - \zeta\omega_n T)z^{-1} + [1 - 2\zeta\omega_n T + (\omega_n T)^2]z^{-2}} \quad (4.488)$$

$$\Rightarrow Y(z) - 2(1 - \zeta\omega_n T)z^{-1}Y(z) + [1 - 2\zeta\omega_n T + (\omega_n T)^2]z^{-2}Y(z) = K(\omega_n T)^2 z^{-2}U(z) \quad (4.489)$$

$$\Rightarrow Y_k - 2(1 - \zeta\omega_n T)y_{k-1} + [1 - 2\zeta\omega_n T + (\omega_n T)^2]y_{k-2} = K(\omega_n T)^2 u_{k-2} \quad (4.490)$$

Assuming the initial conditions are y_{-1} and y_0 , the discrete-time variable k in Equation 4.490 assumes the values $k = 1, 2, 3, \dots$. The first computed value is y_1 .

Initial conditions in the discrete-time system model are based on the initial conditions for the continuous-time system, $y(0)$ and $\dot{y}(0)$. Figure 4.49 illustrates a derivation for y_{-1} using backward extrapolation from the point $y(0)$ along the line with slope $\dot{y}(0)$. Note the dependence on T in the result for y_{-1} . A similar approach is used to extrapolate y_1 when the initial conditions are y_0 and y_1 . The first computed value is y_2 . What is the starting value for k in Equation 4.490?

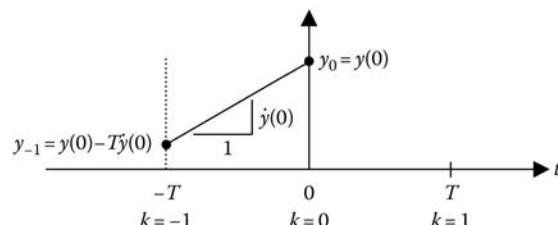


FIGURE 4.49 Initial conditions y_0, y_{-1} obtained from $y(0)$ and $\dot{y}(0)$.

EXAMPLE 4.26

Consider a second-order system with parameters $K = 1$, $\omega_n = 2$ rad/s, and $\zeta = 0.5$.

- Using explicit Euler integration with step size $T = 0.025$ s, find a difference equation that can be solved recursively to approximate the unit step response of the continuous-time system.
- Find the analytical solution for the step response of the continuous-time system.
- Plot the continuous- and discrete-time responses on the same graph.
- A recursive solution for y_k , $k = 1, 2, 3, \dots$ is obtained from Equation 4.490 as follows.

$$y_k = 2(1 - \zeta\omega_n T)y_{k-1} - [1 - 2\zeta\omega_n T + (\omega_n T)^2]y_{k-2} + K(\omega_n T)^2 u_{k-2}, \quad k = 1, 2, 3, \dots \quad (4.491)$$

$$\Rightarrow y_k = 1.95y_{k-1} - 0.9525y_{k-2} + 0.0025u_{k-2}, \quad k = 1, 2, 3, \dots \quad (4.492)$$

- The continuous-time step response can be obtained from the transfer function of the second-order system by inverse Laplace transformation of $Y(s) = H(s)U(s)$. Alternatively, we can use Equation 2.23 or 2.24 for the step response of an underdamped second-order system. Adopting the latter approach,

$$y(t) = K \left[1 - e^{-\zeta\omega_n t} \left(\cos \omega_d t + \frac{\zeta\omega_n}{\omega_d} \sin \omega_d t \right) \right], \quad t \geq 0 \quad \left(\omega_d = \sqrt{1 - \zeta^2} \omega_n \right) \quad (4.493)$$

$$\Rightarrow y(t) = 1 - e^{-t} \left(\cos \sqrt{3}t + \frac{1}{\sqrt{3}} \sin \sqrt{3}t \right) \quad (4.494)$$

- Graphs of the solution to Equations 4.492 (every other point) and 4.494 are plotted in Figure 4.50, and selected values are presented in Table 4.7 for comparison (see MATLAB M-file "Ch4_Ex4_26.m").

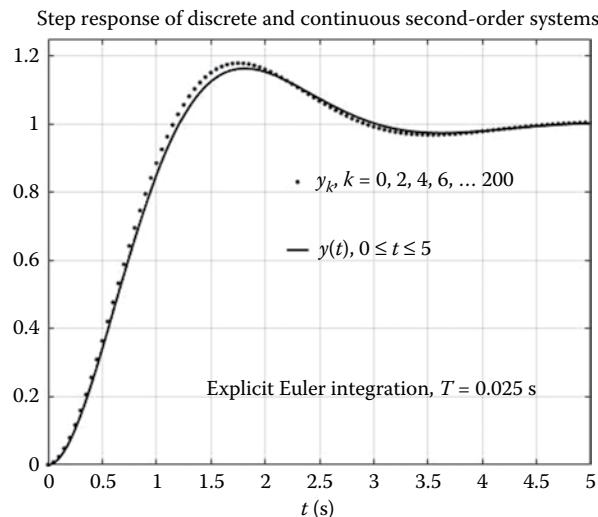


FIGURE 4.50 Continuous- and discrete-time (Euler integration) second-order system step responses ($K = 1$, $\omega_n = 2$ rad/s, $\zeta = 0.5$).

TABLE 4.7
Continuous- and Discrete-Time
(Euler Integration) Responses

<i>K</i>	<i>y_k</i>	<i>t_k</i>	<i>y(t_k)</i>
0	0	0	0
10	0.1170	0.25	0.1044
20	0.3643	0.5	0.3403
30	0.6425	0.75	0.6105
40	0.8845	1.0	0.8494
50	1.0562	1.25	1.0234
60	1.1506	1.5	1.1244
70	1.1787	1.75	1.1616
80	1.1605	2.0	1.1531
90	1.1174	2.25	1.1184
100	1.0677	2.5	1.0746

From the numerical values in [Table 4.7](#), it appears that the discrete- and continuous-time transient responses are in agreement to one place after the decimal point. Greater accuracy requires we reduce the step size or consider a more accurate numerical integrator like the ones discussed in [Chapter 3](#).

A general approach to deriving the *z*-domain transfer function $H(z)$ for a discrete-time system intended to approximate a linear continuous-time system with transfer function $H(s)$ is now given. Starting with a simulation diagram of the continuous-time system, each integrator block with transfer function $H_i(s) = 1/s$ is replaced by a discrete-time transfer function block $H_i(z)$ corresponding to a specific numerical integrator. For example, replacing $H_i(s)$ by $H_i(z)$ for explicit Euler integration,

$$\frac{1}{s} \leftarrow H_i(z) = \frac{T}{z-1} \Rightarrow s \leftarrow \frac{1}{H_i(z)} = \frac{z-1}{T} \quad (4.495)$$

Hence, when explicit Euler integration is used to approximate the continuous-time integrators in an LTI system with transfer function $H(s)$, the *z*-domain transfer function of the discrete-time system is obtained by replacing s in $H(s)$ with $(z-1)/T$. That is,

$$H(z) = H(s) \Big|_{s \leftarrow (z-1)/T} \quad (4.496)$$

For the continuous-time second-order system of [Equation 4.484](#),

$$H(z) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \Big|_{s \leftarrow (z-1)/T} = \frac{K\omega_n^2}{((z-1)/T)^2 + 2\zeta\omega_n((z-1)/T) + \omega_n^2} \quad (4.497)$$

Simplifying [Equation 4.497](#) results in [Equation 4.487](#).

EXAMPLE 4.27

Use trapezoidal integration in place of explicit Euler to approximate the unit step response of the second-order system in [Example 4.26](#).

Approximating a continuous-time integrator with input $u(t)$ and output $y(t)$ using trapezoidal integration results in (see [Equation 3.40](#))

$$y_k = y_{k-1} + \frac{T}{2}[u_{k-1} + u_k] \quad (4.498)$$

z-transforming Equation 4.498,

$$Y(z) - z^{-1} + Y(z) = \frac{T}{2}[z^{-1}U(z) + U(z)] \quad (4.499)$$

and solving for $H_I(z)$ give

$$H_I(z) \frac{Y(z)}{U(z)} = \frac{T}{2} \left[\frac{1+z^{-1}}{1-z^{-1}} \right] \quad (4.500)$$

$$= \frac{T}{2} \left[\frac{z+1}{z-1} \right] \quad (4.501)$$

The z-domain transfer function of the discrete-time system is therefore

$$H(z) = H(s) \Big|_{s \leftarrow \frac{1}{H_I(z)}} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \Big|_{s \leftarrow (2/T)((z-1)/(z+1))} \quad (4.502)$$

Replacing s by $(2/T)((z-1)/(z+1))$ in Equation 4.502 and simplifying result in

$$H(z) = \frac{K(\omega_n T)^2(z^2 + 2z + 1)}{[4(1 + \zeta\omega_n T) + (\omega_n T)^2]z^2 + 2[(\omega_n T)^2 - 4]z + 4(1 - \zeta\omega_n T) + (\omega_n T)^2} \quad (4.503)$$

Multiplying the numerator and denominator of $H(z)$ in Equation 4.503 by z^{-2} leads to the difference equation of the discrete-time system,

$$\begin{aligned} & [4(1 + \zeta\omega_n T) + (\omega_n T)^2]y_k + 2[\omega_n T]^2 - 4]y_{k-1} + [4(1 - \zeta\omega_n T) + (\omega_n T)^2]y_{k-2} \\ & = K(\omega_n T)^2(u_k + 2u_{k-1} + u_{k-2}) \end{aligned} \quad (4.504)$$

Substituting the given values for K , ζ , and ω_n gives

$$y_k = \frac{1}{4.1025} [7.995y_{k-1} - 3.9025y_{k-2} + 0.0025(u_k + 2u_{k-1} + u_{k-2})], \quad k = 1, 2, 3, \dots \quad (4.505)$$

where $u_k = 1$, $k = 0, 1, 2, 3, \dots$ (zero otherwise) and $y_{-1} = y_0 = 0$.

The unit step responses of the continuous- and discrete-time system approximation in Equation 4.505 are calculated in the M-file “Ch4_Ex4_27.m.” The results are graphed in Figure 4.51 and tabulated in Table 4.8. As expected, the trapezoidal integrator is more accurate than the explicit Euler.

4.7.3 SIMULATION DIAGRAMS AND STATE VARIABLES

When a discrete-time system is modeled by one or more difference equations, a simulation diagram represents a more visual description of the system’s dynamics. Furthermore, a simulation diagram leads directly to an equivalent discrete-time state-space model, in much the same way a continuous-time state variable model was developed from a simulation diagram of the continuous-time system. As in the continuous-time case, the simulation diagram and state-space models of a discrete-time system are not unique.

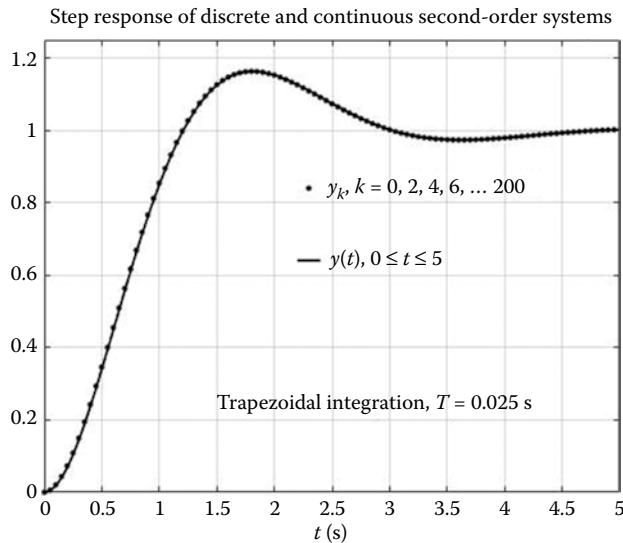


FIGURE 4.51 Continuous- and discrete-time (trapezoidal integration) second-order system step responses ($K = 1$, $\omega_n = 2$ rad/s, $\zeta = 0.5$).

TABLE 4.8
Continuous- and Discrete-Time
(Trapezoidal Integration) Responses

K	y_k	t_k	$y(t_k)$
0	0	0	0
10	0.1090	0.25	0.1044
20	0.3468	0.5	0.3403
30	0.6169	0.75	0.6105
40	0.8546	1.0	0.8494
50	1.0268	1.25	1.0234
60	1.1261	1.5	1.1244
70	1.1620	1.75	1.1616
80	1.1526	2.0	1.1531
90	1.11175	2.25	1.1184
100	1.0736	2.5	1.0746

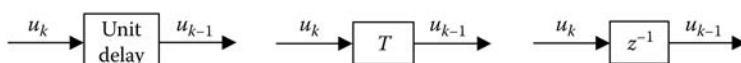


FIGURE 4.52 Graphical representation of the delay property.

The dynamic block in a simulation diagram representation of a continuous-time system is the integrator or $1/s$ block. In a discrete-time system, delaying y_k for one time step results in y_{k-1} . If $y_{-1} = 0$, the delay property states $z\{y_{k-1}\} = z^{-1}z\{y_k\}$. For a discrete-time system, the unit-delay block is the counterpart to the integrator block. Figure 4.52 shows several common ways of representing a unit-delay block.

A block diagram implementation of the n th-order difference Equation 4.456 is shown in Figure 4.53.

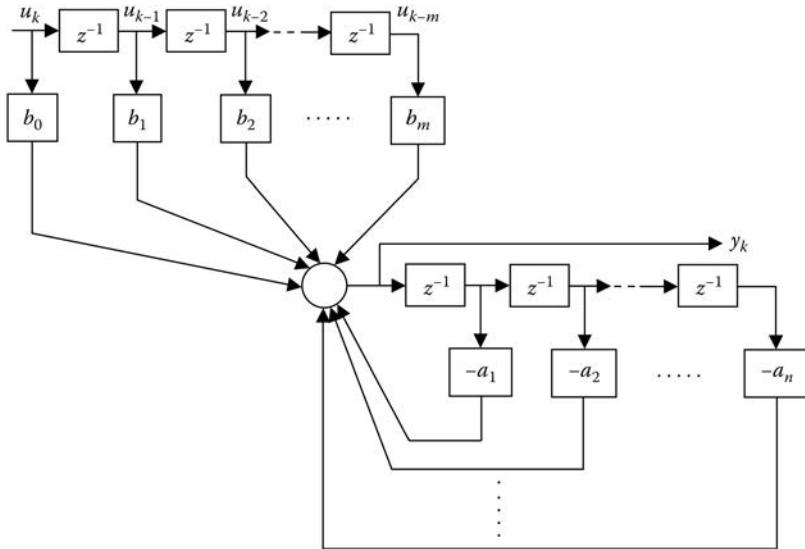


FIGURE 4.53 Block diagram for n th-order discrete-time system in Equation 4.456.

The block diagram shown in Figure 4.53 contains $n + m$ unit delays to implement the n th-order discrete-time system governed by the difference equation in Equation 4.456. Only block diagrams with the minimum number of n delays are classified as simulation diagrams. A simulation diagram serves as a convenient way of identifying the discrete-time states in much the same way continuous-time simulation diagrams were used to define the continuous-time states. The discrete-time states $x_{1,k}, x_{2,k}, \dots, x_{n,k}$ are chosen as the outputs of the n unit delays. As in the case of continuous-time systems, the simulation diagram and, hence, the states are not unique.

When the past input terms $u_{k-1}, u_{k-2}, \dots, u_{k-m}$ are not present in Equation 4.456, the constants $b_1 = b_2 = \dots = b_m = 0$ and the block diagram in Figure 4.53 reduces to a simulation diagram. When one or more past input terms appear in the difference equation, a simulation diagram can be constructed by starting with the z -domain transfer function in Equation 4.459 expressed as

$$\frac{Y(z)}{U(z)} = \frac{Y(z)}{W(z)} \frac{W(z)}{U(z)} \quad (4.506)$$

where

$$\frac{W(z)}{U(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} \quad (4.507)$$

$$\frac{Y(z)}{W(z)} = b_0 + b_1 z^{-1} + \dots + b_m z^{-m} \quad (4.508)$$

Difference equations corresponding to Equations 4.507 and 4.508 are

$$w_k = u_k - a_1 w_{k-1} - a_2 w_{k-2} - \dots - a_n w_{k-n} \quad (4.509)$$

$$y_k = b_0 w_k + b_1 w_{k-1} + \dots + b_m w_{k-m} \quad (4.510)$$

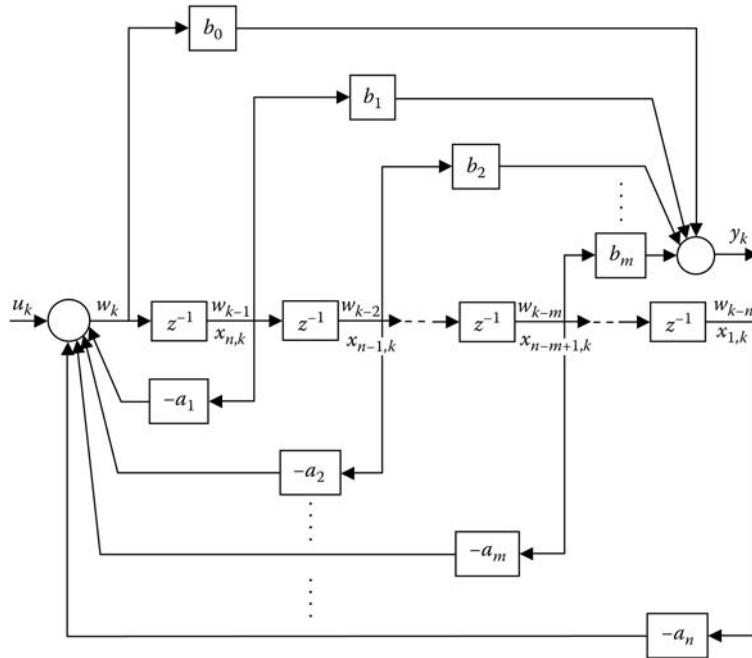


FIGURE 4.54 Simulation diagram for n th-order system showing states $x_{1,k}, x_{2,k}, \dots, x_{n,k}$.

Implementation of Equations 4.509 and 4.510 results in the simulation diagram shown in Figure 4.54. Discrete-time state equations relate the state vector at time $k + 1$ to the discrete-time state vector and input vector at time k . In the single input case with the states as shown in Figure 4.54, the result is

$$\begin{cases} x_{1,k+1} = x_{2,k} \\ x_{2,k+1} = x_{3,k} \\ \vdots \\ x_{n-1,k+1} = x_{n,k} \\ x_{n,k+1} = w_k = -a_n x_{1,k} - \cdots - a_2 x_{n-1,k} - a_1 x_{n,k} + u_k \end{cases} \quad (4.511)$$

The output y_k is expressed in terms of the state and input according to

$$y_k = b_m x_{n-m+1,k} + \cdots + b_2 x_{n-1,k} + b_1 x_{n,k} + b_0 w_k \quad (4.512)$$

$$= b_m x_{n-m+1,k} + \cdots + b_2 x_{n-1,k} + b_1 x_{n,k} + b_0 [u_k - a_n x_{1,k} - \cdots - a_2 x_{n-1,k} - a_1 x_{n,k}] \quad (4.513)$$

$$y_k = \begin{cases} -a_n b_0 x_{1,k} - a_{n-1} b_0 x_{2,k} - \cdots - a_1 b_0 x_{n,k} + b_0 u_k, & m = 0 \\ -a_n b_0 x_{1,k} - a_{n-1} b_0 x_{2,k} - \cdots - a_{m+1} b_0 x_{n-m,k} \\ + (b_m - a_m b_0) x_{n-m+1,k} + \cdots + (b_1 - a_1 b_0) x_{n,k} + b_0 u_k, & m = 1, \dots, n-1 \\ (b_n - a_n b_0) x_{1,k} + (b_{n-1} - a_{n-1} b_0) x_{2,k} + \cdots + (b_1 - a_1 b_0) x_{n,k} + b_0 u_k, & m = n \end{cases} \quad (4.514)$$

In the general case of a linear discrete-time system with r inputs and p outputs, the discrete-time state equations are of the form

$$\underline{x}_{k+1} = A\underline{x}_k + B\underline{u}_k, \quad \underline{y}_k = C\underline{x}_k + D\underline{u}_k \quad (4.515)$$

where the system matrix A is $n \times n$, the input matrix B is $n \times r$, the output matrix C is $p \times n$, and the direct coupling matrix D is $p \times r$. For the discrete-time system described by Equations 4.511 and 4.514, the system matrix A and input matrix B are

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_2 & -a_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (4.516)$$

and the output matrix C and direct transmission matrix D are

$$C = \begin{cases} [-a_n b_0 & -a_{n-1} b_0 & \cdots & -a_1 b_0], & m = 0 \\ [-a_n b_0 & -a_{n-1} b_0 & \cdots & -a_{m+1} b_0 & b_m - a_m b_0 & \cdots & b_1 - a_1 b_0], & m = 1, \dots, n-1 \\ [b_n - a_n b_0 & b_{n-1} - a_{n-1} b_0 & \cdots & b_1 - a_1 b_0], & m = n \end{cases} \quad (4.517)$$

$$D = [b_0] \quad (4.518)$$

The simulation diagram is redrawn for the case where $m = n$ in Figure 4.55.

To illustrate the use of the state equations, consider the discrete-time approximation to a second-order continuous-time system using trapezoidal integration. From Equation 4.504, the difference equation is

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} = b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} \quad (m = n = 2) \quad (4.519)$$

$$a_1 = \frac{2[(\omega_n T)^2 - 4]}{4(1 + \zeta \omega_n T) + (\omega_n T)^2}, \quad a_2 = \frac{4(1 - \zeta \omega_n T) + (\omega_n T)^2}{4(1 + \zeta \omega_n T) + (\omega_n T)^2} \quad (4.520)$$

$$b_0 = \frac{K(\omega_n T)^2}{4(1 + \zeta \omega_n T) + (\omega_n T)^2}, \quad b_1 = \frac{2K(\omega_n T)^2}{4(1 + \zeta \omega_n T) + (\omega_n T)^2}, \quad b_2 = \frac{K(\omega_n T)^2}{4(1 + \zeta \omega_n T) + (\omega_n T)^2} \quad (4.521)$$

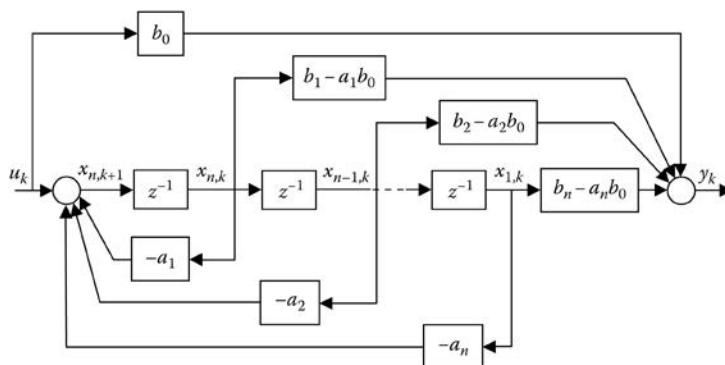


FIGURE 4.55 Simulation diagram for n th-order discrete-time system in Equation 4.456 ($m = n$).

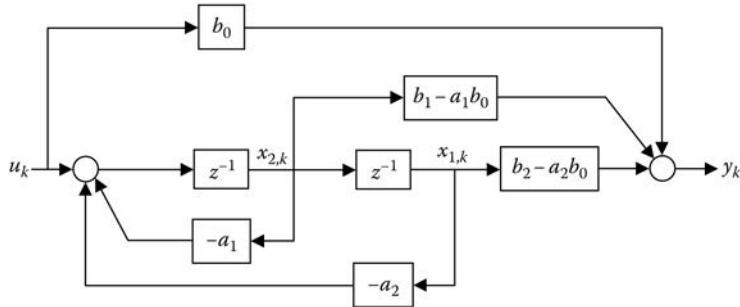


FIGURE 4.56 Simulation diagram for trapezoidal integration of second-order system.

The simulation diagram is shown in [Figure 4.56](#).

The state equations follow directly from [Figure 4.56](#).

$$\begin{cases} x_{1,k+1} = x_{2,k} \\ x_{2,k+1} = -a_2 x_{1,k} - a_1 x_{2,k} + u_k \end{cases} \quad (4.522)$$

$$y_k - (b_2 - a_2 b_0) x_{1,k} + (b_1 - a_1 b_0) x_{2,k} + b_0 u_k \quad (4.523)$$

and the matrices A , B , C , and D in Equation 4.515 are

$$A = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{4(1+\zeta\omega_n T) + (\omega_n T)^2}{4(1+\zeta\omega_n T) + (\omega_n T)^2} & -\frac{2[(\omega_n T)^2 - 4]}{4(1+\zeta\omega_n T) + (\omega_n T)^2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.524)$$

$$C = [b_2 - a_2 b_0 \quad b_1 - a_1 b_0] = 8K \left(\frac{\omega_n T}{4(1+\zeta\omega_n T) + (\omega_n T)^2} \right)^2 [\zeta\omega_n T \quad 2 + \zeta\omega_n T] \quad (4.525)$$

$$D = [b_0] = \left[\frac{K(\omega_n T)^2}{4(1+\zeta\omega_n T) + (\omega_n T)^2} \right] \quad (4.526)$$

Using the same second-order system parameter values as in Examples 4.26 and 4.27, recursive solution of Equations 4.522 and 4.523 produces identical results to those shown in [Figure 4.51](#) and [Table 4.8](#) (see M-file “Ch4_trapezoidal_state.m”).

Discrete-time approximations to the step responses of two additional continuous-time second-order systems, one with light damping ($K = 1$, $\zeta = 0.1$, $\omega_n = 1$ rad/s) and the other heavily damped ($K = 1$, $\zeta = 2.5$, $\omega_n = 1$ rad/s) are shown in [Figure 4.57](#). Results are based on recursive solution of the state equations for trapezoidal integration (see M-file “Ch4_Fig4_57.m”). Agreement between the exact and approximate solutions for both systems appears to be acceptable. More detailed comparisons require numerical outputs from the continuous- and discrete-time systems.

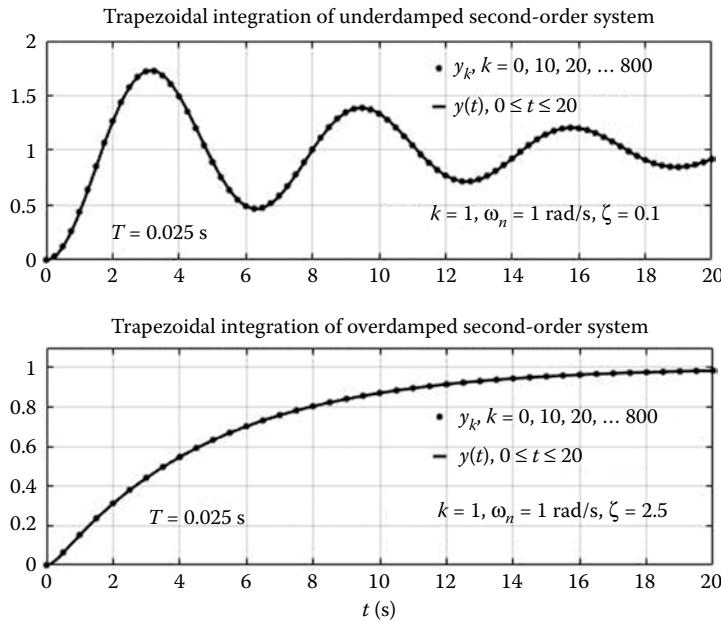


FIGURE 4.57 Trapezoidal integration of (a) light and (b) heavily damped second-order systems.

4.7.4 SOLUTION OF LINEAR DISCRETE-TIME STATE EQUATIONS

A general solution to the discrete-time state equations gives the state \underline{x}_k for any value of discrete-time k without resorting to a recursive (sequential) solution. Solving for the first several values of \underline{x}_k in Equation 4.515 leads to the observation

$$\underline{x}_k = A^k \underline{x}_0 + A^{k-1} B \underline{u}_0 + A^{k-2} B \underline{u}_1 + \cdots + A B \underline{u}_{k-2} + B \underline{u}_{k-1}, \quad k = 0, 1, 2, 3, \dots \quad (4.527)$$

$$= A^k \underline{x}_0 + \sum_{i=0}^{k-1} A^{k-i-1} B \underline{u}_i, \quad k = 0, 1, 2, 3, \dots \quad (4.528)$$

Equation 4.528 for the state \underline{x}_k is substituted in Equation 4.515 to obtain the general solution for the output y_k , $k = 0, 1, 2, 3, \dots$. The result is

$$\underline{y}_k = C A^k \underline{x}_0 + C \left(\sum_{i=0}^{k-1} A^{k-i-1} B \underline{u}_i \right) + D \underline{u}_k, \quad k = 0, 1, 2, 3, \dots \quad (4.529)$$

The discrete-time state transition matrix Φ_k is defined as

$$\Phi_k = A^k, \quad k = 0, 1, 2, 3, \dots \quad (4.530)$$

Solutions for \underline{x}_k and \underline{y}_k , in terms of Φ_k , are

$$\underline{x}_k = \Phi_k \underline{x}_0 + \sum_{i=0}^{k-1} \Phi_{k-i-1} B \underline{u}_i, \quad k = 0, 1, 2, 3, \dots \quad (4.531)$$

$$\underline{y}_k = C\Phi_k \underline{x}_0 + C \sum_{i=0}^{k-1} (\Phi_{k-i-1} B \underline{u}_i) + D \underline{u}_k, \quad k = 0, 1, 2, 3, \dots \quad (4.532)$$

Observe that an unforced system ($\underline{u}_k = 0, k = 0, 1, 2, 3, \dots$) transitions from its initial state \underline{x}_0 to a new state \underline{x}_k at time k according to $\underline{x}_k = \Phi_k \underline{x}_0$. The discrete-time state equations and solutions are analogous to the results for continuous-time systems.

An expression for evaluating the discrete-time transition matrix can be obtained by z -transforming the first equation in Equation 4.515 resulting in

$$z\{\underline{x}_{k+1}\} = z\{A\underline{x}_k + B\underline{u}_k\} = A\underline{X}(z) + B\underline{U}(z) \quad (4.533)$$

It is left as an exercise to show that

$$z\{\underline{x}_{k+1}\} = z[X(z) - x_0] \quad (4.534)$$

Combining Equations 4.533 and 4.534 gives

$$z[\underline{X}(z) - \underline{x}_0] = A\underline{X}(z) + B\underline{U}(z) \quad (4.535)$$

$$(zI - A)\underline{X}(z) = z\underline{x}_0 + B\underline{U}(z) \quad (4.536)$$

$$\underline{X}(z) = (zI - A)^{-1}[z\underline{x}_0 + B\underline{U}(z)] \quad (4.537)$$

$$\underline{x}_k = z^{-1}\{zI - A\}^{-1}(z\underline{x}_0) + z^{-1}\{(zI - A)^{-1}B\underline{U}(z)\} \quad (4.538)$$

Comparison of Equations 4.531 and 4.538 with $u_k = 0, k = 0, 1, 2, \dots$ implies

$$\Phi_k = z^{-1}\{\Phi(z)\} = z^{-1}\{z(zI - A)^{-1}\} \quad (4.539)$$

An example using the discrete-time state equations follows.

EXAMPLE 4.28

The yearly increase in a monetary fund is a weighted sum of the increases over the prior 2 years plus an end-of-year (EOY) deposit. The fund starts with an initial amount P_0 .

- Write the difference equation for $y_k, k = 0, 1, 2, 3, \dots$ the fund balance at the end of the k th year. Let $u_k, k = 0, 1, 2, 3, \dots$ be the EOY deposit in the fund. The weights are α (previous year increase) and β (increase 2 years ago).
- Draw a simulation diagram and convert the difference equation to state variable form.
- Given $\alpha = 0.5$, $\beta = 0.25$, and $P_0 = \$100$, and all EOY deposits are zero, find the components of the discrete-time state transition matrix needed to solve for $y_k, k = 0, 1, 2, 3, \dots$
- Find and plot the fund balance $y_k, k = 0, 1, 2, 3, \dots$
- The time line in Figure 4.58 shows the relationship between the discrete-time variable k and the EOY marker. Note that the initial fund amount is y_{-1} .

The difference equation for $y_k, k = 0, 1, 2, 3, \dots$ is

$$y_k - y_{k-1} = \alpha(y_{k-1} - y_{k-2}) + \beta(y_{k-2}y_{k-3}) + u_k, \quad k = 0, 1, 2, 3, \dots \quad (4.540)$$

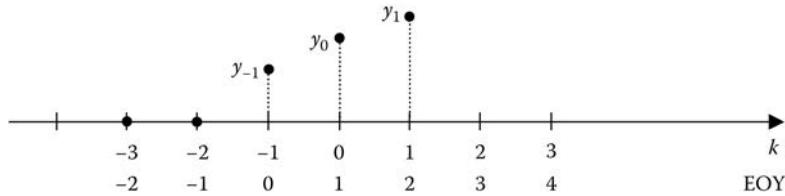


FIGURE 4.58 Relationship between discrete-time variable k and end of year.

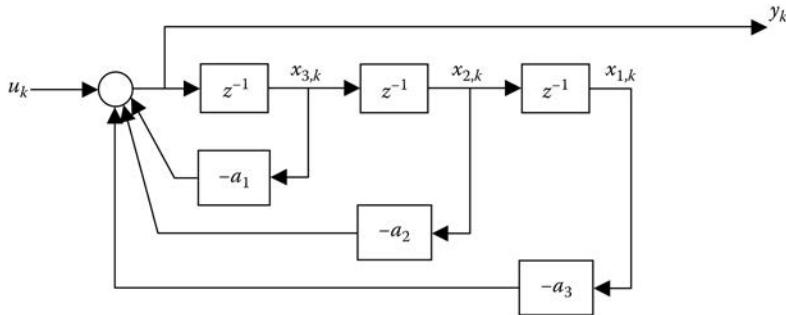


FIGURE 4.59 Simulation diagram for monetary fund example.

The initial conditions are $y_{-1} = P_0$, $y_{-2} = y_{-3} = 0$.

Rewriting Equation 4.540 in the standard form introduced in Equation 4.456

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + a_3 y_{k-3} = b_0 u_k, \quad k = 0, 1, 2, 3, \dots \quad (4.541)$$

where $a_1 = -(1 + \alpha)$, $a_2 = \alpha - \beta$, $a_3 = \beta$, and $b_0 = 1$.

- b. Referring to Figure 4.53 or 4.54 with $n = 3$, $m = 0$, and $b_0 = 1$, the simulation diagram reduces to Figure 4.59.

The state equations follow from the simulation diagram.

$$\begin{cases} x_{1,k+1} = x_{2,k} \\ x_{2,k+1} = x_{3,k} \\ x_{3,k+1} = -a_3 x_{1,k} - a_2 x_{2,k} - a_1 x_{3,k} + u_k \end{cases} \quad (4.542)$$

$$y_k = -a_3 x_{1,k} - a_2 x_{2,k} - a_1 x_{3,k} + u_k \quad (4.543)$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\beta & \beta - \alpha & 1 + \alpha \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.544)$$

$$C = [-a_3 - a_2 - a_1] = [-\beta \quad \beta - \alpha \quad 1 + \alpha], \quad D = [1] \quad (4.545)$$

- c. $a_1 = -(1 + \alpha) = -(1 + 0.5) = -1.5$, $a_2 = \alpha - \beta = 0.5 - 0.25 = 0.25$, $a_3 = \beta = 0.25$

$$zI - A = z \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \quad (4.546)$$

$$= \begin{bmatrix} z & -1 & 0 \\ 0 & z & -1 \\ a_3 & a_2 & z + a_1 \end{bmatrix} = \begin{bmatrix} z & -1 & 0 \\ 0 & z & -1 \\ 0.25 & 0.25 & z - 1.5 \end{bmatrix} \quad (4.547)$$

$$\Phi(z) = z(zI - A)^{-1} \quad (4.548)$$

Inverting $(zI - A)$ followed by multiplication by z results in

$$\Phi(z) = \frac{z}{z^3 - 1.5z^2 + 0.25z + 0.25} \begin{bmatrix} z^2 - 1.5z + 0.25 & z - 1.5 & 1 \\ -0.25 & z(z - 1.5) & z \\ -0.25z & -0.25(z + 1) & z^2 \end{bmatrix} \quad (4.549)$$

From Equation 4.532, with $\underline{y}_k = \underline{0}$, $k = 0, 1, 2, \dots$ the solution for y_k is

$$y_k = C\Phi_k \underline{x}_0 \quad (4.550)$$

where the initial state

$$\underline{x}_0 = \begin{bmatrix} y_{-3} \\ y_{-2} \\ y_{-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ P_0 \end{bmatrix}$$

The transition matrix Φ_k is obtained by inverse z -transforming $\Phi(z)$ in Equation 4.549. The last column of Φ_k is all that is necessary to determine y_k as a result of the zeros in the first and second rows of \underline{x}_0 . The last column of Φ_k comprises

$$(\Phi_k)_{1,3} = z^{-1}\{\Phi_{1,1}(z)\} = z^{-1}\left\{\frac{z}{z^3 - 1.5z^2 + 0.25z + 0.25}\right\} \quad (4.551)$$

$$(\Phi_k)_{2,3} = z^{-1}\{\Phi_{2,1}(z)\} = z^{-1}\left\{\frac{z^2}{z^3 - 1.5z^2 + 0.25z + 0.25}\right\} \quad (4.552)$$

$$(\Phi_k)_{3,3} = z^{-1}\{\Phi_{3,1}(z)\} = z^{-1}\left\{\frac{z^3}{z^3 - 1.5z^2 + 0.25z + 0.25}\right\} \quad (4.553)$$

The roots of $z^3 - 1.5z^2 + 0.25z + 0.25 = 0$ are $p_1 = 1$, $p_2 = 0.8090$, $p_3 = -0.3090$. $(\Phi_k)_{1,3}$, $(\Phi_k)_{2,3}$, $(\Phi_k)_{3,3}$ are linear combinations of the geometric sequences $(p_1)^k$, $(p_2)^k$, $(p_3)^k$, that is,

$$(\Phi_k)_{1,3} = A_1(p_1)^k + A_2(p_2)^k + A_3(p_3)^k \quad (4.554)$$

$$(\Phi_k)_{2,3} = B_1(p_1)^k + B_2(p_2)^k + B_3(p_3)^k \quad (4.555)$$

$$(\Phi_k)_{3,3} = C_1(p_1)^k + C_2(p_2)^k + C_3(p_3)^k \quad (4.556)$$

The partial fraction expansion coefficients are evaluated in M-file "Ch4_Ex4_28.m." The results are

$$\begin{aligned} A_1 &= 4, & A_2 &= -4.6833, & A_3 &= 0.6833 \\ B_1 &= 4, & B_2 &= -3.7889, & B_3 &= -0.2111 \\ C_1 &= 4, & C_2 &= -3.065, & C_3 &= 0.0652 \end{aligned}$$

d. From Equations 4.545 and 4.550, the fund balance is

$$y_k = [-a_3 - a_2 - a_1] \begin{bmatrix} (\Phi_k)_{1,3} \\ (\Phi_k)_{2,3} \\ (\Phi_k)_{3,3} \end{bmatrix} P_0 \quad (4.557)$$

$$= [-\beta \beta - \alpha 1 + \alpha] \begin{bmatrix} A_1(p_1)^k + A_2(p_2)^k + A_3(p_3)^k \\ B_1(p_1)^k + B_2(p_2)^k + B_3(p_3)^k \\ C_1(p_1)^k + C_2(p_2)^k + C_3(p_3)^k \end{bmatrix} P_0 \quad (4.558)$$

$$\begin{aligned} &= \{-\beta[A_1(p_1)^k + A_2(p_2)^k + A_3(p_3)^k] + (\beta - \alpha)[B_1(p_1)^k + B_2(p_2)^k + B_3(p_3)^k] \\ &\quad + (1 + \alpha)[C_1(p_1)^k + C_2(p_2)^k + C_3(p_3)^k]\}P_0 \end{aligned} \quad (4.559)$$

A graph of y_k , $k = -3, -2, -1, 0, 1, 2, \dots$ is shown in Figure 4.60.

The limiting value, $y_\infty = \$400$ from the root $p_1 = 1$. Since the magnitude of roots p_2 and p_3 are less than 1, it follows from Equation 4.559 at steady state that the output y_∞ is given by

$$\begin{aligned} y_\infty &= \lim_{k \rightarrow \infty} y_k = [-\beta A_1 + (\beta - \alpha) B_1 + (1 + \alpha) C_1] P_0 \\ &= [-(0.25)(4) + (0.25 - 0.5)(4) + (1 + 0.5)(4)] 100 = 400 \end{aligned} \quad (4.560)$$

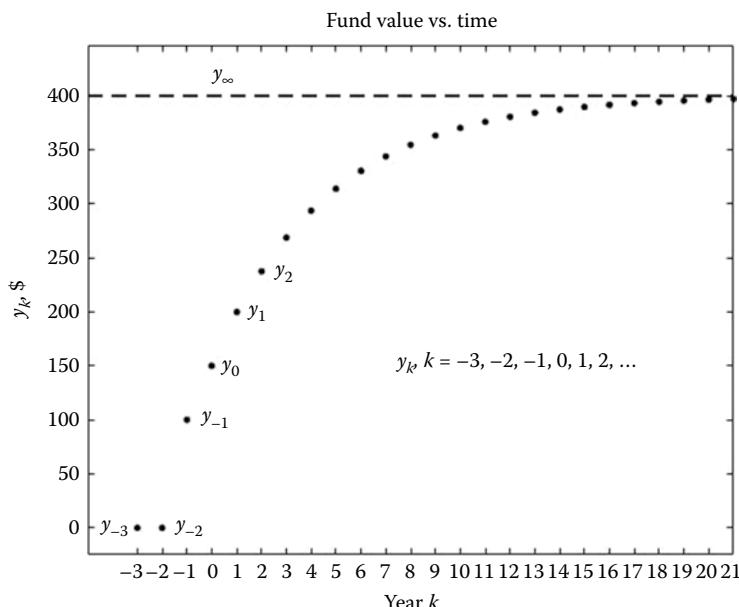


FIGURE 4.60 Discrete-time system output in Example 4.28.

We will have a lot more to say about the location of these roots in the next section on stability.

The complete transition matrix is left as an exercise problem. However, a suitable check on the correctness of Φ_k is that it satisfies $\Phi_0 = I$, where I is the $n \times n$ identity matrix. This follows from Equation 4.530 with $k = 0$ as well as Equation 4.531 with zero input and $k = 0$. A quick glance at $\Phi_0 = I$ in Equation 4.549 should be enough to convince you that $\Phi_0 = I$ (*Hint:* only the diagonal terms of $\Phi(z)$ contain cubic polynomials in z in the numerator). Keep in mind that $\Phi_0 = I$ is necessary but not sufficient for Φ_k to be correct.

4.7.5 WEIGHTING SEQUENCE (IMPULSE RESPONSE FUNCTION)

A difference equation and a z -domain transfer function are but two of several different ways of characterizing a discrete-time system. A third approach is based on the system's impulse response function, similar to the case of continuous-time systems. Recall from our discussion of linear continuous-time systems that the response to an arbitrary input $u(t)$, $t \geq 0$ is expressible in the form of a convolution integral, that is,

$$y(t) = \int_0^t h(\tau)u(t-\tau)d\tau \quad (4.561)$$

where $h(t)$, $t \geq 0$ is the impulse response function. It is related to the continuous-time system transfer function $H(s)$ according to $h(t) = \mathcal{L}^{-1}\{H(s)\}$.

We now demonstrate the existence of a sequence, h_k , $k = 0, 1, 2, 3, \dots$ which allows us to find the forced response of a linear discrete-time system to an arbitrary input u_k , $k = 0, 1, 2, 3, \dots$ similar to the convolution integral in Equation 4.561 for linear continuous-time systems. The only restriction is that the initial conditions prior to application of the input, namely, $y_{-1}, y_{-2}, \dots, y_{-n}$, are zero for an n th-order linear discrete-time system.

Consider the first-order system

$$Y_k + a_1 y_{k-1} = b_0 u_k + b_1 u_{k-1} \quad (4.562)$$

where $y_{-1} = 0$ and the input $u_k = 0$, $k < 0$. Evaluating the first several values of y_k ,

$$k = 0 : y_0 = b_0 u_0 \quad (4.563)$$

$$k = 1 : y_1 + a_1 y_0 = b_0 u_1 + b_1 u_0 \quad (4.564)$$

$$y_1 = b_0 u_1 + (b_1 - a_1 b_0) u_0 \quad (4.565)$$

$$k = 2 : y_2 + = a_1 y_1 = b_0 u_2 + b_1 u_1 \quad (4.566)$$

$$y_2 = b_0 u_2 + (b_1 - a_1 b_0) u_1 - a_1 (b_1 - a_1 b_0) u_0 \quad (4.567)$$

$$k = 3 : y_3 + a_0 y_2 = b_1 u_3 + b_0 u_2 \quad (4.568)$$

$$y_3 = b_0 u_3 + (b_1 - a_1 b_0) u_2 - a_1 (b_1 - a_1 b_0) u_1 + a_1^2 (b_1 - a_1 b_0) u_0 \quad (4.569)$$

By induction, a general solution for y_k , $k = 0, 1, 2, 3, \dots$ is

$$y_k = \sum_{i=0}^k h_i u_{k-i}, \quad k = 0, 1, 2, 3, \dots \quad (4.570)$$

where

$$h_i = \begin{cases} b_0, & i = 0 \\ (b_1 - a_1 b_0)(-a_1)^{i-1}, & i = 1, 2, 3, \dots \end{cases} \quad (4.571)$$

The discrete-time variable in Equation 4.571 is written as “ i ” instead of “ k ” to avoid confusion; however, it is helpful to think of the sequence as h_k , $k = 0, 1, 2, 3, \dots$. Equation 4.570 reveals that the current output y_k is a linear combination of the current and past inputs, that is, writing out the terms in the sum

$$y_k = h_0 u_k + h_1 u_{k-1} + h_2 u_{k-2} + \dots + h_k u_0, \quad k = 0, 1, 2, 3, \dots \quad (4.572)$$

The weights are in fact the numerical values of the sequence h_k , $k = 0, 1, 2, 3, \dots$ with the current input u_k weighted by h_0 , the previous input u_{k-1} weighted by h_1 up to the oldest input u_0 with a weight of h_k . The sequence h_k , $k = 0, 1, 2, 3, \dots$ in Equations 4.570 and 4.572 is called the weighting sequence of the discrete-time system.

The sum in Equation 4.570 is called the convolution sum, the counterpart to the convolution integral for continuous-time systems in Equation 4.561. The weighting sequence and convolution sum representation are not restricted to the simple first-order discrete-time system in Equation 4.562. They are applicable to n th-order LTI discrete-time systems. Fortunately, a more efficient technique for determining the weighting sequence than was previously illustrated exists. The method is deferred until after the following example.

EXAMPLE 4.29

The low-pass filter in Equation 4.473 is a first-order discrete-time system.

- a. Find the weighting sequence h_k , $k = 0, 1, 2, 3, \dots$.
- b. Graph the weighting sequence for $\alpha = 0.9$ and $\beta = 0.1$.
- c. Find the unit step response by convolution, and compare the result with the response in Equation 4.478 with $y_{-1} = 0$.
- a. For the discrete-time system in Equation 4.473, $a_1 = -\alpha$, $b_0 = \beta$, and $b_1 = 0$. The weighting sequence given in Equation 4.571 reduces to

$$h_k = \begin{cases} \beta, & k = 0 \\ (\alpha\beta)(\alpha)^{k-1}, & k = 1, 2, 3, \dots \end{cases} \quad (4.573)$$

$$= \beta(\alpha)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.574)$$

- b. The weighting sequence with $\alpha = 0.9$ and $\beta = 0.1$ is graphed in Figure 4.61.
- c. From Equation 4.570 with $u_k = 1$, $k = 0, 1, 2, 3, \dots$, the unit step response is

$$y_k = \sum_{i=0}^k h_i u_{k-i} = \sum_{i=0}^k h_i = \sum_{i=0}^k \beta\alpha^i \quad (4.575)$$

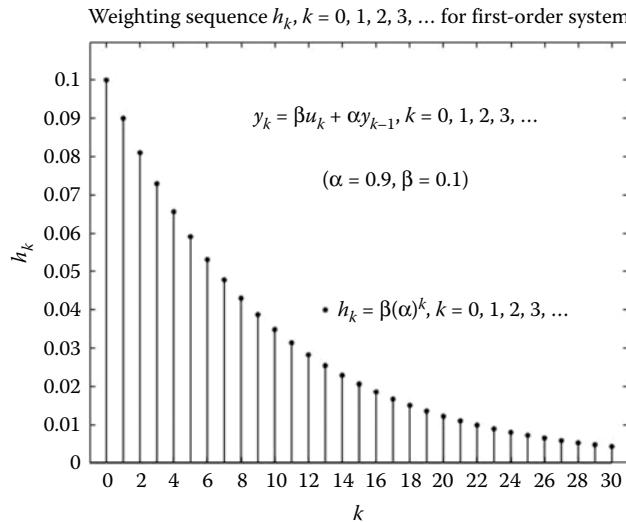


FIGURE 4.61 Weighting sequence $h_k, k = 0, 1, 2, 3, \dots$ for first-order system in Equation 4.473.

$$= \beta \left(\frac{1 - \alpha^{k+1}}{1 - \alpha} \right) \quad (4.576)$$

$$= 1 - (0.9)^{k+1}, \quad k = 0, 1, 2, 3, \dots \quad (4.577)$$

in agreement with the unit step response obtained from Equation 4.478 with $y_{-1} = 0$.

The memory and transient response of a stable linear discrete-time system are reflected in its weighting sequence. Loosely speaking, the memory in a discrete-time system depends on how far back past inputs affect the current output in a significant way, that is, if the current output is predominantly influenced by only the last several inputs, then the system is said to exhibit a relatively short memory. Conversely, if distant inputs are influential in determining the current output, the system possesses a longer memory.

From the convolution sum representation for the current output y_k in Equation 4.572, it is readily apparent that the amount of memory in the system is directly related to how fast the weighting sequence approaches zero. (Discrete-time systems with weighting sequences that do not approach zero as k approaches infinity are considered in the next section dealing with stability.) Transient and steady-state response will also be considered at the same time; however, it should be clear even now that a fast responding system, that is, one with a short transient response must have a weighting sequence that approaches zero quickly and is, therefore, characterized as a system with a short memory.

For the first-order system considered in Example 4.29, the rate of decay to zero in the weighting sequence depends solely on the parameter α . Figure 4.62 shows the unit step responses of three first-order systems with different values of α and $\beta = 1 - \alpha$.

One is a fast responding system ($\alpha = 0.3$), one with moderate speed ($\alpha = 0.7$), and the last one is seen to have a sluggish response ($\alpha = 0.9$).

The response of an LTI discrete-time system to an impulse δ_k is quite significant. From the convolution sum in Equation 4.570, the unit impulse response is

$$(y_k)_{\text{impulse response}} = \sum_{i=0}^k h_i \delta_{k-i} = h_k, \quad k = 0, 1, 2, 3, \dots \quad (4.578)$$

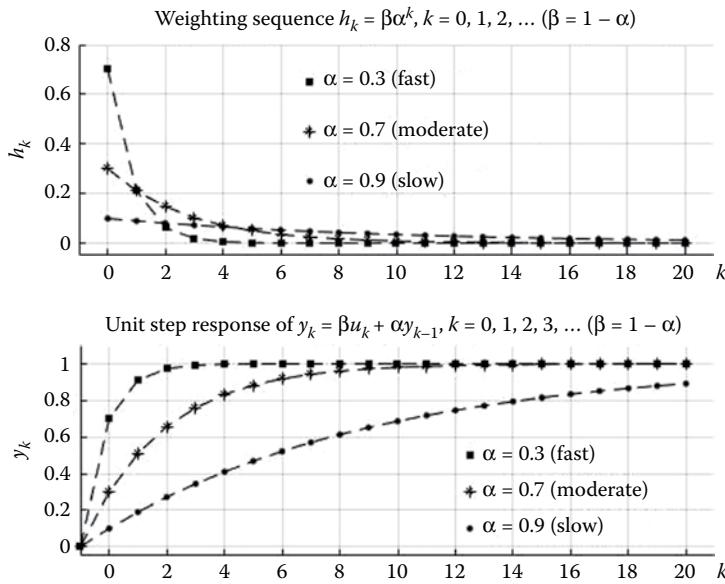


FIGURE 4.62 Weighting sequences and unit step responses of three first-order discrete-time systems governed by $y_k = (1 - \alpha)u_k + \alpha y_{k-1}$, $k = 0, 1, 2, \dots$

In other words, the impulse response is identical to the weighting sequence. Furthermore, for a system with z -domain transfer function $H(z)$, the z -transform of the impulse response is given by

$$Y_{\text{impulse response}}(z) = H(z)z\{\delta_k\} = H(z) \cdot 1 = H(z) \quad (4.579)$$

Invert z -transforming Equation 4.579,

$$(y_k)_{\text{impulse response}} = h_k = z^{-1}\{H(z)\} \quad (4.580)$$

Equation 4.580 tells us the impulse response of an LTI discrete-time system is equal to the inverse z -transform of the z -domain transfer function of the system. Henceforth, the impulse response sequence will be denoted h_k , $k = 0, 1, 2, \dots$. This most important property of discrete-time systems is illustrated in Figure 4.63.

The z -domain transfer function of the first-order system in Example 4.29 is

$$H(z) = \frac{Y(z)}{U(z)} = \frac{\beta z}{z - \alpha} \quad (4.581)$$

$$h_k = z^{-1}\{H(z)\} \quad (4.582)$$

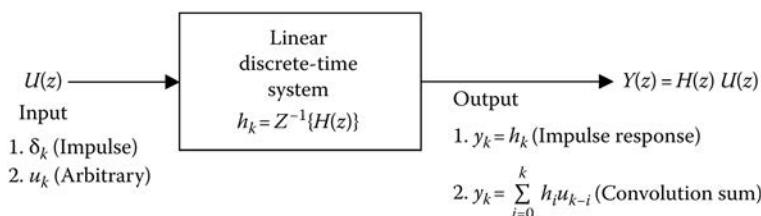


FIGURE 4.63 Relationship of impulse response to z -domain transfer function.

$$= z^{-1} \left\{ \frac{\beta z}{z - \alpha} \right\} \quad (4.583)$$

$$= \beta \alpha^k, \quad k = 0, 1, 2, 3, \dots \quad (4.584)$$

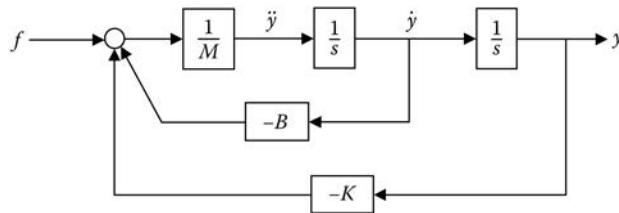
The impulse response (weighting sequence) is therefore the same as in Equation 4.574.

The impulse response is fundamental to the design of digital filters implemented by linear difference equations. The two major categories of such filters are FIR and IIR, which stand for “finite impulse response” and “infinite impulse response,” respectively (Orfanidis 1996).

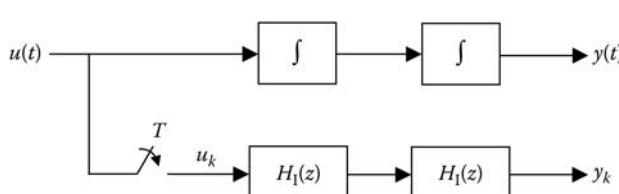
EXERCISES

- 4.43 Find the z -domain transfer function of the discrete-time system, which results from an approximation to a continuous-time integrator using
- Implicit Euler integration
 - Improved Euler integration
- 4.44 Find the z -domain transfer function $H(z)$ of the discrete-time system resulting from approximation of the first-order system $\tau \dot{y}(t) + y(t) = ku(t)$ using the following numerical integrators:
- Explicit Euler
 - Implicit Euler
 - Trapezoidal
- 4.45 Let $u_k, k = 0, 1, 2, 3, \dots$ be uniformly spaced samples of an input $u(t)$ and $y_k, k = 0, 1, 2, 3, \dots$ be an approximation to $y(t) = \int_0^t u(t)dt$ based on trapezoidal integration.
- Find a difference equation relating u_k and y_k .
 - Solve the difference equation recursively using an appropriate step size to approximate the area under
 - $u(t) = te^{-t/2}, 1 \leq t \leq 2$
 - $u(t) = (1/\sqrt{2\pi})e^{-t^2/2}, 0 \leq t \leq 5$
- 4.46 Prove Equation 4.534 for the scalar case, that is, show that $z[x_{k+1}] = z[X(z) - x_0]$, where x_0 is the value of x_k at $k = 0$.
- 4.47 In Example 4.28, find the complete transition matrix and verify that $\Phi_0 = I$.
- 4.48 In Example 4.28,
- Find $Y(z)$, the z -transform of the response, by z -transforming the difference equation of the system with appropriate initial conditions.
 - Find y_∞ by applying the final value property (see Table 4.5).
 - Find y_0 by applying the initial value property (see Table 4.5).
 - Find $y|_{k=0} = z - 1\{Y(z)\}$.
- 4.49 In Example 4.28, assume the initial conditions $y_{-3} = y_{-2} = y_{-1} = 0$.
- Find $H(z) = Y(z)/U(z)$, the z -domain transfer function of the system.
 - The input is $u_k = A_0 \delta_k + A_1 \delta_{k-1} + A_2 \delta_{k-2}$ and $y_{-1} = y_{-2} = y_{-3} = 0$. Find A_0, A_1, A_2 if the response is identical to the case when $u_k = 0, k = 0, 1, 2, \dots$ and $y_{-1} = P_0, y_{-2} = y_{-3} = 0$.
- 4.50 The unit step response of a discrete-time system is $y_k = -1 + 3^{k+1}, k = 0, 1, 2, 3, \dots$.
- Find the difference equation relating u_k and y_k .
 - Find the impulse response, $h_k, k = 0, 1, 2, 3, \dots$.
- 4.51 The discrete-time signal $u_k = 1 + k, k = 0, 1, 2, 3, \dots$ is delayed one unit of discrete-time and then input to a discrete-time system with z -domain transfer function $H(z) = Y(z)/U(z) = z^2/(z + 1)^2$. Find the output y_k at $k = 3$ and $k = 6$.
- 4.52 A discrete-time system with input u_k and output y_k is governed by the difference equation $y_k = \alpha_1 y_{k-1} + \beta_1 u_{k-1} + \beta_0 u_k, k = 0, 1, 2, 3, \dots$
- Find the z -domain transfer function of the system

- b. Find the impulse response sequence $h_k, k = 0, 1, 2, 3, \dots$
 - i. By inverse z -transformation of $H(z)$
 - ii. By recursive solution of the difference equation with $u_k = \delta_k$
 - c. Find the final value of the unit step response in terms of α_1, β_0 , and β_1 .
 - i. By letting $k \rightarrow \infty$ in the unit step response
 - ii. By applying the final value property
 - iii. By setting $u_k = 1, k = 0, 1, 2, 3, \dots$ and solving for $y_\infty = \lim_{k \rightarrow \infty} y_k = \lim_{k \rightarrow \infty} y_{k-1}$ in the difference equation
- 4.53 Use the same approach for finding $z\{y_{k-1}\}$ when $y_{-1} \neq 0$ resulting in Equation 4.472 to find
- a. $z\{y_{k-2}\}, y_{-1}, y_{-2} \neq 0$
 - b. $z\{y_{k-n}\}, y_{-1}, y_{-2}, \dots, y_{-n} \neq 0$
- 4.54 A discrete-time system is described by $y_k + a_1 y_{k-1} + a_2 y_{k-2} = 0, k = 0, 1, 2, 3, \dots$
- a. Find $Y(z)$ for the case when $y_{-1} = 0$ and $y_{-2} = 0$.
 - b. Find $Y(z)$ for the case when the right-hand side is $b_0 \delta_k + b_1 \delta_{k-1}$ and $y_{-1} = y_{-2} = 0$.
 - c. Find expressions for the weights b_0 and b_1 in terms of a_1, a_2, y_{-1} , and y_{-2} , so that the response $y_k, k = 0, 1, 2, 3, \dots$ is the same in parts (a) and (b). Comment on the implication of replacing initial conditions with impulse forcing functions.
- 4.55 A simulation diagram for an $M-B-K$ mechanical system governed by the second-order differential equation $M\ddot{y}(t) + B\dot{y}(t) + Ky(t) = f(t)$ is shown in [Figure E4.55](#):

**FIGURE E4.55**

- a. Find a difference equation relating y_k and f_k based on the use of explicit Euler integration. Convert the difference equation to state variable form.
 - b. Find a difference equation relating y_k and f_k based on the use of implicit Euler integration. Convert the difference equation to state variable form.
 - c. Find a difference equation relating y_k and f_k based on the use of trapezoidal Euler integration. Convert the difference equation to state variable form.
 - d. Find a difference equation relating y_k and f_k based on the use of explicit Euler integration for the first integrator (\dot{y}) and implicit Euler integration for the second integrator (y). Convert the difference equation to state variable form.
 - e. Approximate the unit step response of the system for parts (a) through (d) when $M = 1$, $B = 2$, and $K = 1$, and compare each with the continuous-time response.
- 4.56 Consider the double integrator shown in [Figure E4.56](#):

**FIGURE E4.56**

- a. Write the differential equation relating $y(t)$ and $u(t)$.
 - b. Find the difference equation relating y_k and u_k if both numerical integrators are based on explicit Euler integration.
 - c. Find dy/dt and $y(t)$ when the initial conditions are $y(0) = 0$, $\dot{y}(0) = 1$ and the input $u(t) = 10 - e^{-t/2}$, $t \geq 0$.
 - d. Find the z -domain transfer function and impulse response of the discrete-time system.
 - e. Find the output y_k , $k = 1, 2, 3, \dots$ when the integration step size $T = 0.1$ s.
 - f. Plot the continuous- and discrete-time outputs on the same graph, and comment on the results.
- 4.57 A discrete-time system is described by $y_k + a_1 y_{k-1} + a_2 y_{k-2} = 0$, $k = 0, 1, 2, 3, \dots$
- a. Find $Y(z)$ for the case when $y_{-1} \neq 0$, $y_{-2} \neq 0$.
 - b. Find $Y(z)$ for the case when the right-hand side is $b_0 \delta_k + b_1 \delta_{k-1}$ and $y_{-1} = y_{-2} = 0$.
 - c. Find expressions for the weights b_o and b_i in terms of a_1 , a_2 , y_{-1} , y_{-2} so that the response y_k , $k = 0, 1, 2, 3, \dots$ is the same in parts (a) and (b). Comment on the implication of replacing initial conditions with impulse forcing functions.
- 4.58 Show that the unit step response of a discrete-time system with z -domain transfer function $H(z)$ is given by

$$y_k = z^{-1} \left\{ \frac{z}{z-1} H(z) \right\}, \quad k = 0, 1, 2, \dots$$

4.8 STABILITY OF LTI DISCRETE-TIME SYSTEMS

One way of characterizing the stability of a discrete-time system is by the way it responds to a bounded input. When the response remains bounded, the system is said to exhibit BIBO stability. The implications of BIBO stability on the system's z -domain transfer function, impulse response (weighting sequence), and natural response will be explored.

Consider an n th-order LTI discrete-time system described by Equation 4.456 in the previous section. The z -domain transfer function is

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n}, \quad n \geq m \quad (4.585)$$

Suppose the poles of $H(z)$ are real and distinct. Then

$$Y(z) = H(z)U(z) = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_m z^{n-m}}{(z - p_1)(z - p_2) \cdots (z - p_n)} U(z) \quad (4.586)$$

In the case where the poles of $U(z)$ are different from p_1, p_2, \dots, p_n ,

$$\begin{aligned} Y(z) &= A_0 \left\{ A_1 \frac{z}{z - p_1} + A_2 \frac{z}{z - p_2} + \cdots + A_n \frac{z}{z - p_n} \right\} \\ &\quad + \text{terms due to poles of } U(z)z^{-1}\{U(z)\} \end{aligned} \quad (4.587)$$

The response y_k , $k = 0, 1, 2, 3, \dots$ is therefore

$$y_k = A_0 \delta_k + \{A_1 p_1^k + A_2 p_2^k + \cdots + A_n p_n^k\} + \text{terms generated from } z^{-1}\{U(z)\} \quad (4.588)$$

The bracketed expression is the natural response, that is, a linear combination of the natural modes $p_1^k, p_2^k, \dots, p_n^k$, while the terms arising from the inverse z -transformation of $U(z)$ are similar in nature to the input and comprise the forced component of the overall response. Since the natural

response is excited by the presence of an input, it must obviously be a bounded sequence for a BIBO stable system.

The impulse response $h_k = z^{-1}\{H(z)\}$ is also a linear combination of the system's natural modes $p_1^k, p_2^k, \dots, p_n^k$, (plus in some cases, a weighted impulse at the origin). Imagine a discrete-time system with impulse response $h_k, k = 0, 1, 2, \dots$ subject to a unit step input $u_k = 1, k = 0, 1, 2, \dots$. Using the convolution sum form of the output,

$$|y_k| = \left| \sum_{i=0}^k h_i u_{k-i} \right| = \left| \sum_{i=0}^k h_i \right| < \sum_{i=0}^k |h_i|, \quad k = 0, 1, 2, \dots \quad (4.589)$$

From Equation 4.589, the step response at discrete-time k remains finite provided the sum of the first $k + 1$ values of the magnitude of the impulse response satisfies

$$\sum_{i=0}^k |h_i| < \infty, \quad k = 0, 1, 2, 3, \dots \quad (4.590)$$

It follows that the entire response $y_k, k = 0, 1, 2, 3, \dots$ is bounded whenever the impulse response sequence satisfies

$$\sum_{k=0}^{\infty} |h_k| < \infty \quad (4.591)$$

While Equation 4.591 was derived for the case where the input is a unit step, it applies to any bounded input. Equation 4.591 is a necessary and sufficient condition for the output of an LTI discrete-time system to remain bounded in response to any bounded input. A consequence of Equation 4.591 is that the weighting sequence of a BIBO stable system must decay to zero as $k \rightarrow \infty$.

From Equation 4.588, an n th-order LTI discrete-time system with z -domain transfer function having real and distinct poles is BIBO stable when the poles satisfy

$$-1 < p_i < 1, \quad i = 1, 2, 3, \dots, n \quad (4.592)$$

The expression for the output y_k in Equation 4.588 assumed that the poles of $H(z)$ were real and distinct. A real pole p with multiplicity m generates a weighted sum of the natural modes $p^k, kp^k, \dots, k^{m-1} p^k$ in the output; however, Equation 4.592 still applies for BIBO stability.

When a pair of complex poles of $H(z)$ is present, y_k contains trigonometric terms like $R^k (k\theta + \phi)$ where R is the magnitude of the complex poles. In order to include the possibility of complex poles of $H(z)$, Equation 4.592 is appropriately expressed as

$$|p_i| < 1, \quad i = 1, 2, \dots, n \quad (4.593)$$

Consequently, a sufficient condition for BIBO stability of LTI discrete-time systems is that all of its z -domain transfer function poles have a magnitude less than 1, that is, all poles are located inside the Unit Circle in the complex plane.

In Example 4.30, we look at a second-order system with real and distinct poles subject to a bounded input. The effect of moving one of the poles is investigated. Following that, we consider the ramifications of various locations of the z -domain transfer function's poles in the complex plane.

EXAMPLE 4.30

A discrete-time system is described by the difference equation

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} = b_0 u_k, \quad k = 0, 1, 2, 3, \dots \quad (4.594)$$

Initial conditions $y_{-1} = y_{-2} = 0$. The input sequence is given by

$$u_k = 1 + (0.1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.595)$$

Find the z-domain transfer function $H(z)$ and its poles, the impulse response h_k , $k = 0, 1, 2, 3, \dots$, the total response y_k , $k = 0, 1, 2, 3, \dots$, and the natural and forced components of the total response, and comment on stability for the following cases:

- (a) $a_1 = 0, \quad a_2 = -0.25, \quad b_0 = 1$
- (b) $a_1 = -0.5, \quad a_2 = -0.5, \quad b_0 = 1$
- (c) $a_1 = -1.5, \quad a_2 = -1, \quad b_0 = 1$

a. z-transforming the difference equation $y_k - 0.25y_{k-2} = u_k$, $k = 0, 1, 2, 3, \dots$ yields

$$H(z) = \frac{Y(z)}{U(z)} = \frac{z^2}{z^2 - 0.25} = \frac{z^2}{(z - 0.5)(z + 0.5)} \quad (4.596)$$

with poles $p_1 = -0.5, p_2 = 0.5$. The impulse response is obtained from

$$h_k = z^{-1}\{H(z)\} = z^{-1} \left\{ \frac{z^2}{(z + 0.5)(z - 0.5)} \right\} \quad (4.597)$$

$$= z^{-1} \left\{ \frac{0.5z}{z + 0.5} + \frac{0.5z}{z - 0.5} \right\} \quad (4.598)$$

$$= 0.5[(-0.5)^k + (0.5)^k], \quad k = 0, 1, 2, 3, \dots \quad (4.599)$$

$$= (0.5)^{k+1}[(-1)^k + 1], \quad k = 0, 1, 2, 3, \dots \quad (4.600)$$

$$= (0.5)^k, \quad k = 0, 2, 4, 6, \dots \quad (4.601)$$

The complete response y_k , $k = 0, 1, 2, \dots$ is determined by inverse z-transformation of

$$Y(z) = \frac{z^2}{(z^2 + 0.25)} \left[\frac{z}{z-1} + \frac{z}{z-0.1} \right] \quad (4.602)$$

$$= \frac{z^3(2z - 1.1)}{(z + 0.5)(z - 0.5)(z - 1)(z - 0.1)} \quad (4.603)$$

$$= \frac{7}{12} \left(\frac{z}{z + 0.5} \right) + \frac{1}{8} \left(\frac{z}{z - 0.5} \right) + \frac{4}{3} \left(\frac{z}{z - 1} \right) - \frac{1}{24} \left(\frac{z}{z - 0.1} \right) \quad (4.604)$$

$$\Rightarrow y_k = \frac{7}{12}(-0.5)^k + \frac{1}{8}(-0.5)^k + \frac{4}{3} - \frac{1}{24}(0.1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.605)$$

From Equation 4.605, the natural (free) response and forced response are

$$(y_k)_{\text{natural}} = \frac{7}{12}(-0.5)^k + \frac{1}{8}(0.5)^k, \quad k = 0, 1, 2, \dots \quad (4.606)$$

$$(y_k)_{\text{forced}} = \frac{4}{3} - \frac{1}{24}(0.1)^k, \quad k = 0, 1, 2, \dots \quad (4.607)$$

The system is stable as evidenced by the natural response decaying to zero as $k \rightarrow \infty$. This was expected since the two poles of $H(z)$ are located between -1 and $+1$. Can you show that Equation 4.591 is satisfied as well? Note the similarity between the natural response in Equation 4.606 and the impulse response in Equation 4.599.

- b. The difference equation becomes $y_k - 0.5y_{k-1} - 0.5y_{k-2} = u_k, k = 0, 1, 2, 3, \dots$. The results for this system are obtained in an analogous fashion to part (a).

$$H(z) = \frac{z^2}{z^2 - 0.5z - 0.5} = \frac{z^2}{(z + 0.5)(z - 1)} \quad (p_1 = -0.5, p_2 = 1) \quad (4.608)$$

$$h_k = \frac{1}{3}[(-0.5)^k + 2], \quad k = 0, 1, 2, 3, \dots \quad (4.609)$$

$$y_k = \frac{7}{18}(-0.5)^k + \frac{44}{27} + \frac{2}{3}k - \frac{1}{54}(0.1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.610)$$

$$(y_k)_{\text{nat}} = \frac{7}{18}(-0.5)^k + \frac{44}{27}, \quad k = 0, 1, 2, 3, \dots \quad (4.611)$$

$$(y_k)_{\text{forced}} = \frac{2}{3}k - \frac{1}{54}(0.1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.612)$$

The forced response also contains a constant component resulting from the pole of $U(z)$ at $z = 1$. This constant is combined with the constant in the natural response, and the sum of $44/27$ is shown entirely in the natural response in Equation 4.611.

The second pole of $H(z)$, namely, $p_2 = 1$, does not satisfy Equation 4.592, and the system is not BIBO stable. In this case, a bounded input produced an unbounded output. The impulse response in Equation 4.609 does not asymptotically decay to zero.

- c. The difference equation is $y_k - 1.5y_{k-1} - y_{k-2} = u_k, k = 0, 1, 2, 3, \dots$.

$$H(z) = \frac{z^2}{z^2 - 1.5z - 1} = \frac{z^2}{(z + 0.5)(z - 2)}, \quad (p_1 = -0.5, p_2 = 2) \quad (4.613)$$

$$h_k = \frac{1}{5}(-0.5)^k + \frac{4}{5}(2)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.614)$$

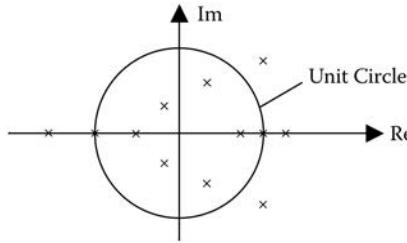


FIGURE 4.64 The Unit Circle and various locations of real and complex poles.

$$y_k = \frac{7}{30}(-0.5)^k + \frac{232}{95}(2)^k - \frac{2}{3} - \frac{1}{114}(0.1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.615)$$

$$(y_k)_{\text{natural}} = \frac{7}{30}(-0.5)^k + \frac{232}{95}(2)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.616)$$

$$(y_k)_{\text{forced}} = -\frac{2}{3} - \frac{1}{114}(0.1)^k, \quad k = 0, 1, 2, 3, \dots \quad (4.617)$$

Once again, the system is unstable. The natural response and, by implication, the impulse response are unbounded as $k \rightarrow \infty$.

The real poles of an nth-order LTI discrete-time system transfer function are located on the real axis in the complex plane. Figure 4.64 shows real poles located at (from right to left) 1.25, 1, 0.75, -0.5, -1, and -1.5 along the real axis.

There are six distinct regions for location of real poles along the real axis, each with a different type of natural mode. According to Equation 4.592, only the poles at 0.75 and -0.5 located inside the Unit Circle correspond to stable natural modes. The impulse response h_k , $k = 0, 1, 2, \dots$ approaches zero as $k \rightarrow \infty$ in both cases. When the poles are located on the Unit Circle at +1 and 1, the impulse response sequence remains finite as $k \rightarrow \infty$; however, a linear discrete-time system with a pole at either location is not BIBO stable.

The remaining two cases correspond to real poles located outside the Unit Circle, either in the region $p > 1$ or $p < -1$. The natural response of an LTI discrete-time system with poles located in either region is unbounded, and, hence, the system is not BIBO stable.

Figure 4.65 illustrates the natural modes corresponding to each of the real poles.

4.8.1 COMPLEX POLES OF $H(z)$

Three pairs of complex poles are also shown in Figure 4.64. The z -domain transfer function $H(z)$ possesses a pair of complex poles if there is a quadratic factor in its denominator with complex roots. Figure 4.66 illustrates the case where $H(z)$ has complex poles at $z = a \pm jb$. The transformation to polar form $z = re^{\pm j\theta}$ is shown as well.

In terms of polar coordinates, the quadratic factor is

$$Q(z) = (z - re^{j\theta})(z - re^{-j\theta}) = z^2 - (2r\cos\theta)z + r^2 \quad (4.618)$$

Consider a second-order discrete-time system with z -domain transfer function

$$H(z) = \frac{Az^2 + Bz}{z^2 - (2r\cos\theta)z + r^2} \quad (4.619)$$

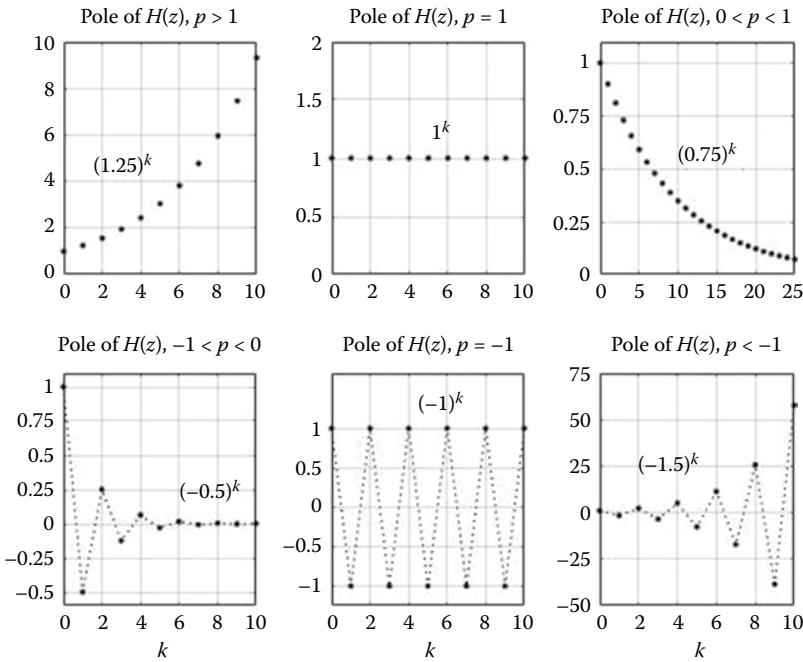


FIGURE 4.65 Natural modes corresponding to real poles of $H(z)$.

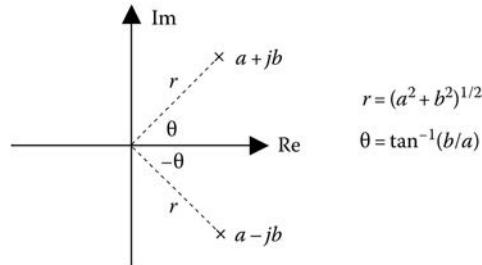


FIGURE 4.66 Complex poles of discrete-time system transfer function $H(z)$.

For reasons that will soon become apparent, $H(z)$ is expressed as

$$H(z) = \frac{c_1(r \sin \theta)z + c_2[z^2 - (r \cos \theta)z]}{z^2 - (2r \cos \theta)z + r^2} \quad (4.620)$$

where c_1 and c_2 are obtained by equating like powers of z in the numerators of Equations 4.619 and 4.620. The result is

$$c_1 = \frac{Ar \cos \theta + B}{r \sin \theta}, \quad c_2 = A \quad (4.621)$$

$H(z)$ in Equation 4.620 is expressed as

$$H(z) = c_1 \left[\frac{(r \sin \theta)z}{z^2 - (2r \cos \theta)z + r^2} \right] + c_2 \left[\frac{z^2 - (r \sin \theta)z}{z^2 - (2r \cos \theta)z + r^2} \right] \quad (4.622)$$

Referring to [Table 4.4](#) with $e^{-aT} = r$ and $\omega T = \theta$ suggests the impulse response $h_k = z^{-1} \{H(z)\}$ is

$$h_k = c_1 r^k \sin k\theta + c_2 r^k \cos k\theta = r^k (c_1 \sin k\theta + c_2 \cos k\theta), \quad k = 0, 1, 2, 3, \dots \quad (4.623)$$

There are three cases to consider, which are illustrated in [Figure 4.64](#). The three cases correspond to the region inside the Unit Circle ($r < 1$), all points on the Unit Circle ($r = 1$), and the exterior of the Unit Circle ($r > 1$). It follows from Equation 4.623 that the impulse response satisfies the necessary condition for BIBO stability in Equation 4.591 only in the first case, $r < 1$, that is, when the poles are located inside the Unit Circle. The natural response, being of similar form to the impulse response, decays to zero as $k \rightarrow \infty$. Hence, the system is BIBO stable.

When the poles are either on the Unit Circle or outside, Equation 4.591 is not satisfied, and the system is therefore not BIBO stable. The natural response consists of sustained oscillations when $r = 1$ and oscillations of increasing magnitude when $r > 1$.

EXAMPLE 4.31

A second-order discrete-time system has a z-domain transfer function given by

$$H(z) = \frac{z^2 + 3z}{Q(z)} \quad (4.624)$$

where $Q(z)$ is a quadratic with complex roots located in the three different regions like the ones shown in [Figure 4.64](#). Suppose the roots are

$$(a) -0.25 \pm j0.5 \quad (b) 0.5(1 \pm j\sqrt{3}) \quad (c) 1 \pm j$$

- a. Find the z-domain transfer function $H(z)$ for each case.
- b. Find the impulse response h_k , $k = 0, 1, 2, 3, \dots$ for each case.
- c. Graph the impulse response for each case.

a.

- i. $(a = -0.25, b = 0.5)$. The polar coordinates of the transfer function poles are

$$r = [(-0.25)^2 + (0.5)^2]^{1/2} = 0.5990, \quad \theta = \tan^{-1}\left(\frac{0.5}{-0.25}\right) = 2.0344 \text{ rad}$$

$$Q(z) = z^2 - (2r \cos \theta)z + r^2$$

$$= z^2 - [2(0.5990) \cos(2.0344)]z + (0.5990)^2$$

$$= z^2 + 0.5z + 0.3125$$

$$\Rightarrow H(z) = \frac{z^2 + 3z}{Q(z)} = \frac{z^2 + 3z}{z^2 + 0.5z + 0.3125} \quad (4.625)$$

$$\text{ii. } (a = 0.5, b = 0.5\sqrt{3}) \Rightarrow r = 1, \quad \theta = 1.0472 \text{ rad} \quad H(z) = \frac{z^2 + 3z}{z^2 - z + 1} \quad (4.626)$$

$$\text{iii. } (a = 1, b = 1) \Rightarrow r = \sqrt{2}, \quad \theta = \frac{\pi}{4} \text{ rad}, \quad H(z) = \frac{z^2 + 3z}{z^2 - 2z + 2} \quad (4.627)$$

$$\text{b. } c_1 = \frac{Ar\cos\theta + B}{r\sin\theta} = \frac{1(0.5990)\cos(2.0344) + 3}{0.5990\sin(2.0344)} = 5.5, \quad c_2 = A = 1$$

The constants c_1 and c_2 for (ii) and (iii) are determined in similar fashion. From Equation 4.623, the impulse responses are

$$\text{i. } h_k = (0.5990)^k [5.5 \sin(2.0344k) + \cos(2.0344k)], \quad k = 0, 1, 2, 3, \dots \quad (4.628)$$

$$\text{ii. } h_k = 4.0415 \sin(1.0472k) + \cos(1.0472k), \quad k = 0, 1, 2, 3, \dots \quad (4.629)$$

$$\text{iii. } h_k = (\sqrt{2})^k \left[4 \sin\left(\frac{k\pi}{4}\right) + \cos\left(\frac{k\pi}{4}\right) \right], \quad k = 0, 1, 2, 3, \dots \quad (4.630)$$

c. Graphs of the impulse responses in Equations 4.628 through 4.630 are shown in Figure 4.67.

The discrete-time system with poles located inside the Unit Circle is BIBO stable. The impulse response given in Equation 4.628 satisfies the necessary and sufficient condition for BIBO stability in Equation 4.591. Poles of the transfer functions in Equations 4.626 and 4.627 are situated on the Unit Circle and outside it, respectively. Neither system is BIBO stable.

Consider a system with a pair of complex poles of $H(z)$ on the Unit Circle at $e^{\pm j\theta}$. Its response to the bounded input $u_k = \sin k\theta, k = 0, 1, 2, 3, \dots$ is obtained from

$$Y(z) = H(z)U(z) = \frac{N(z)}{(z - e^{j\theta})(z - e^{-j\theta})} \cdot \frac{\sin \theta \cdot z}{z^2 - (2 \cos \theta)z + 1} \quad (4.631)$$

$$= \frac{N(z)}{(z - e^{j\theta})(z - e^{-j\theta})} \cdot \frac{\sin \theta \cdot z}{(z - e^{j\theta})(z - e^{-j\theta})} \quad (4.632)$$

$$= \frac{\sin \theta \cdot z N(z)}{(z - e^{j\theta})(z - e^{-j\theta})^2} \quad (4.633)$$

It is left as an exercise to show that y_k contains a linear combination of the terms, $\cos k\theta, \sin k\theta, k \cos k\theta$, and $k \sin k\theta$. Consequently, the response is unbounded, and the system is not BIBO stable.

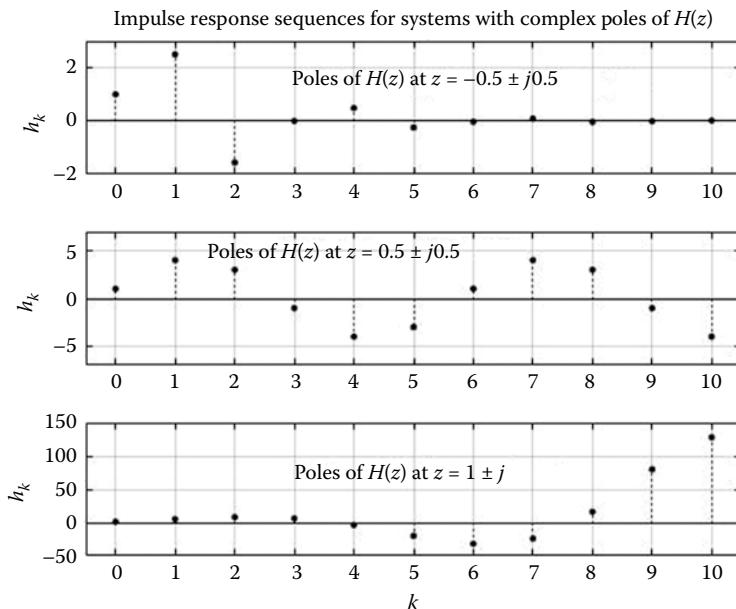


FIGURE 4.67 Impulse responses for discrete-time systems with different complex poles of $H(z)$.

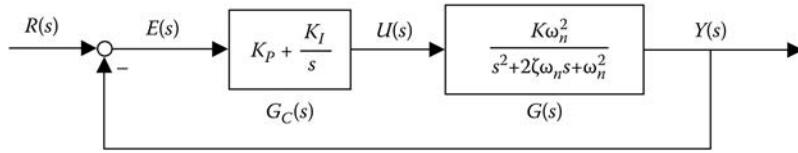


FIGURE 4.68 P-I control of a second-order continuous-time process.

When a real pole of $H(z)$ is located on the Unit Circle at $z = -1$ or $z = +1$, and the input is $u_k = (-1)^k$, $k = 0, 1, 2, 3$, or the unit step $u_k = 1$, $k = 0, 1, 2, 3, \dots$, respectively, the response is unbounded due to the presence of $(z + 1)^2$ or $(z - 1)^2$ in the denominator of the output $Y(z)$. The first case results in the term $k(-1)^k$ (multiplied by a constant) appearing in the output. In the second case, y_k contains a term proportional to $k(1)^k = k$ (see Example 4.30, part [b]).

We conclude this section with a simulation of the continuous-time control system in Figure 4.68. The analog P-I (Proportional–Integral) controller $G_c(s)$ is approximated by a discrete-time controller with transfer function $G_c(z)$ based on the use of trapezoidal integration. It was shown in Example 4.27 in Section 4.7.2 that the z-domain transfer function of a trapezoidal integrator is

$$H_I(z) = \frac{T}{2} \left[\frac{z+1}{z-1} \right] \quad (4.634)$$

And, therefore, $G_c(z)$ is obtained by replacing s with $1/H_I(z)$, that is,

$$G_c(z) = K_p + \frac{K_I}{s} \Big|_{s \leftarrow (2/T)((z-1)/(z+1))} = \frac{(2K_p + K_IT)z - 2K_p + K_IT}{2(z-1)} \quad (4.635)$$

Several discrete-time approximations to the second-order system in Figure 4.68 were developed in Section 4.7.2. Explicit Euler approximation resulted in

$$G(z) = \frac{Y(z)}{U(z)} = \frac{K(\omega_n T)^2}{z^2 - 2(1 - \zeta\omega_n T)z + 1 - 2\zeta\omega_n T + (\omega_n T)^2} \quad (4.636)$$

The block diagram of the discrete-time system intended to simulate the continuous-time control system is shown in Figure 4.69.

The closed-loop transfer function is

$$H(z) \frac{Y(z)}{R(z)} = \frac{G_c(z)G(z)}{1 + G_c(z)G(z)} \quad (4.637)$$

and the poles of $H(z)$ are the roots of the characteristic equation

$$\Delta(z) = 1 + G_c(z)G(z) = 0 \quad (4.638)$$

Substituting Equations 4.635 and 4.636 into Equation 4.638 yields

$$\Delta(z) = z^3 + \alpha_1 z^2 + \alpha_2 z + \alpha_3 = 0 \quad (4.639)$$

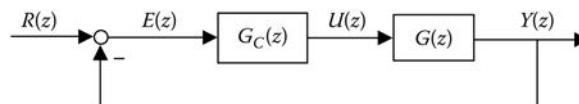


FIGURE 4.69 Block diagram of discrete-time system.

TABLE 4.9
Continuous- and Discrete-Time Control System Poles

System Parameters and Integration Step Size	Poles of $H(s)$	Poles of $H(z)$	Mag of $H(z)$ Complex Poles
$K = 1, \omega_n = 10, \zeta = 1.0$			
$K_p = 0.5, K_I = 2$	$-9.1616 \pm j5.9448$	$0.5398 \pm j0.3201$	0.628
$T = 0.05$	-1.6768	0.9205	
$K = 1, \omega_n = 5, \zeta = 0.5124$			
$K_p = 1, K_I = 3$	$-1.7133 \pm j6.4225$	$0.8674 \pm j0.4979$	1.000
$T = 0.075$	-1.6975	0.8808	
$K = 1, \omega_n = 20, \zeta = 0.15$			
$K_p = 1, K_I = 3$	$-2.2436 \pm j28.0745$	$0.9436 \pm j0.7085$	1.180
$T = 0.025$	-1.5128	0.9629	

where

$$\left. \begin{aligned} \alpha_1 &= -3 + 2\zeta\omega_n T \\ \alpha_2 &= 3 - 4\zeta\omega_n T + (\omega_n T)^2 [1 + K(K_p + 0.5K_I T)] \\ \alpha_3 &= -1 + 2\zeta\omega_n T + (\omega_n T)^2 [-1 + K(-K_p + 0.5K_I T)] \end{aligned} \right\} \quad (4.640)$$

Table 4.9 summarizes the results for different combinations of continuous-time second-order systems, controllers, and integration step size.

The continuous-time system and discrete-time poles are shown in [Figure 4.70](#). All three continuous-time control systems are stable since the poles are located in the left-half plane. The discrete-time systems for simulating them, however, are not all BIBO stable. In fact, the discrete-time

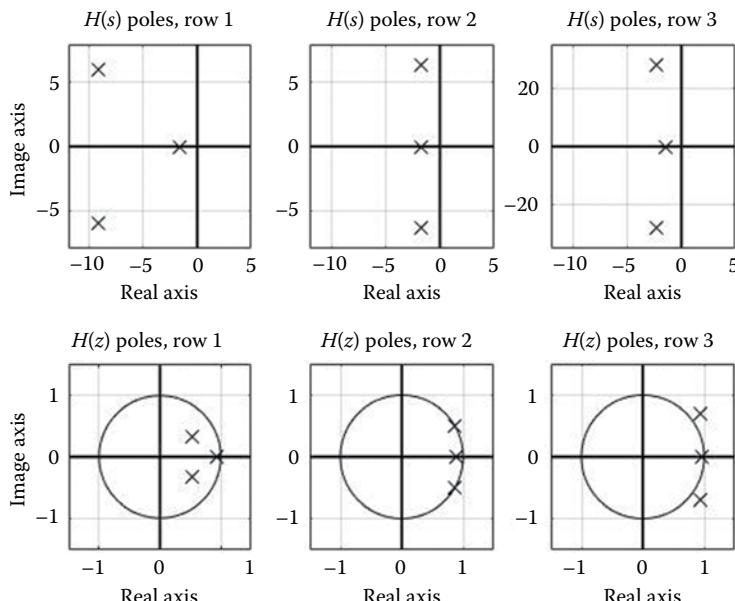


FIGURE 4.70 Continuous- and discrete-time system poles for rows 1, 2, 3 in [Table 4.9](#).

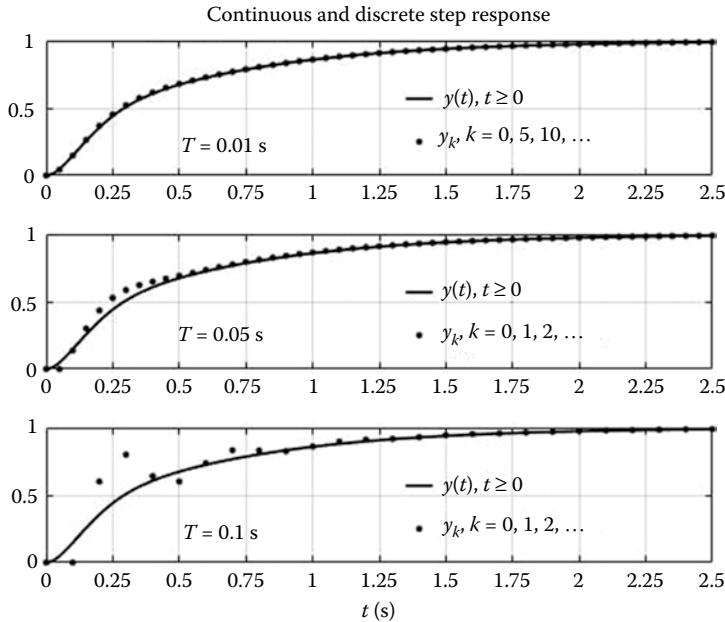


FIGURE 4.71 Unit step response of continuous-time system (Figure 4.68) and discrete-time system (Figure 4.69) with $T = 0.01, 0.05, 0.1 \text{ s}$.

systems in Rows 2 and 3 in Table 4.9 possess a pair of complex poles located on the Unit Circle and outside it, respectively.

It is important to keep in mind that while the discrete-time system approximation in Row 1 of Table 4.9 is stable, its accuracy in approximating the continuous-time system response to various inputs is another matter. Suppose the input to the control system in Figure 4.68 is $r(t), t \geq 0$ and the output is $y(t), t \geq 0$. If the discrete-time system response to $r_k = r(kT), k = 0, 1, 2, \dots$ is $y_k, k = 0, 1, 2, \dots$, an accurate simulation requires that $y_k \approx y(kT), k = 0, 1, 2, \dots$

The locations of the continuous- and discrete-time poles corresponding to Row 1 in Figure 4.70 imply that the natural responses consist of a monotonically decaying and damped oscillatory modes. The question still remains whether the time constants and damped natural frequencies are comparable. A thorough examination of this point is deferred to a later chapter; however, we can gain insight by looking at the step responses of each system.

EXAMPLE 4.32

Find and graph the unit step response of the continuous-time system in Figure 4.68 ($K = 1, \omega_n = 10, \zeta = 1.0, K_p = 0.5, K_I = 2$) and the discrete-time system shown in Figure 4.69 with integration step size $T = 0.01, 0.05, 0.1 \text{ s}$.

The step responses are computed in M-file "Ch4_Ex4_32.m" and shown in Figure 4.71. The top graph is a plot of every fourth point of the discrete-time system step response.

The discrete-time system is stable for all three values of integration step size T and the steady-state values are identical to the continuous-time steady-state value. However, the transient response of the discrete-time system when $T = 0.1 \text{ s}$ varies considerably from the continuous-time system transient response.

EXERCISES

- 4.59 Find the poles of the z -domain transfer functions $H(z)$ below, and comment on the stability of the corresponding discrete-time systems.

$$(a) \frac{z^2 + 2z}{32z^3 - 16z^2 - 22z + 1}$$

$$(b) \frac{4z^2}{z^3 - (3/2)z^2 + (3/4)z - (1/8)}$$

$$(c) \frac{3z + 1}{4z^3 - 3z + 1}$$

$$(d) \frac{z^4 - z}{16z^4 - 28z^2 + 22z^2 - 8z + 1}$$

$$(e) \frac{1}{4z^4 + 3z^2 - 1}$$

$$(f) \frac{z^3 + 2z^2 + z}{2z^3 - 5z^2 + 6z - 2}$$

4.60 Prove $\sum_{k=0}^{\infty} |h_k| < \infty$ is a sufficient condition for BIBO stability of an LTI discrete-time system.

4.61 A discrete-time system is described by the difference equation

$$4y_k - 3y_{k-2} + y_{k-3} = u_k, \quad k = 0, 1, 2, 3, \dots \quad (y_{-1} = y_{-2} = y_{-3} = 0)$$

- a. Find the weighting sequence $h_k, k = 0, 1, 2, 3, \dots$ of the system.
- b. Check whether the condition $\sum_{k=0}^{\infty} |h_k| < \infty$ is satisfied. Is the system stable?
- c. Find and graph the system response to the input $u_k = 1 + 2(-1)^k, k = 0, 1, 2, 3, \dots$

4.62 Show the work required to establish Equations 4.608 through 4.610 in part (b) and Equations 4.613 through 4.615 in part (c) of Example 4.30.

4.63 Find the inverse z -transform of $Y(z) = \frac{(\sin \theta \cdot z N(z))}{((z - e^{j\theta})^2(z - e^{-j\theta})^2)}$ in Equation 4.633.

4.64 For a discrete-time system with z -domain transfer function given by

$$H(z) = \frac{Y(z)}{U(z)} = \frac{z^2 + z + 1}{z^3 - 0.5z^2 - z + 0.5}$$

- a. Find the zeros and poles of $H(z)$.
- b. Find the impulse response sequence $h_k, k = 0, 1, 2, 3, \dots$.
- c. Find the unit step response.
- d. Find the forced response to $u_k = (-1)^k, k = 0, 1, 2, 3, \dots$.

4.65 For the control system in Figure 4.68,

- a. Find the transfer function $H_E(s) = E(s)/R(s)$.
- b. Use explicit Euler integration with integration step T to obtain $H_E(z)$, an approximation to the continuous-time transfer function $H_E(s)$.
- c. Assume $K = 1, \omega_n = 10, \zeta = 1.0, K_p = 0.5, K_I = 2$, and $T = 0.05$, and find the poles of $H_E(z)$. Compare your answer with the results shown in Table 4.9 for the same parameter values.
- d. Find the difference equation relating e_k and $r_k, k = 0, 1, 2, 3, \dots$.
- e. Solve the difference equation when $r_k = 1, k = 0, 1, 2, 3, \dots$.

4.66 For the control system in Figure 4.69 with baseline parameters specified in the last row of Table 4.9, find the poles of $H(z)$ and plot the magnitude of the most distant pole(s) from the origin when

- a. ζ varies from 0 to 2
- b. T varies from 0.01 to 0.5 s
- c. K_p varies from 0.5 to 5
- d. ω_n varies from 5 to 50 rad/s

4.67 End-of-month deposits $d_k, k = 0, 1, 2, 3, \dots$ are placed in an investment account paying interest at a rate of i per month. The initial account balance is P_0 . The difference equation for P_k , the account balance after k months, is

$$P_{k+1} = (1+i)P_k + d_{k+1}, \quad k = 0, 1, 2, 3, \dots$$

- a. Find the z -domain transfer function $H(z) = P(z)/D(z)$ and its pole.

Hint: Use the left shifting property, $z\{P_{k+1}\} = zP(z) - zP_0$. Comment on the stability of the discrete-time system.

- b. Sketch P_k , $k = 0, 1, 2, 3, \dots$ when no deposits are made and $i > 0$. Repeat for $i < 0$.
- c. Find the general solution for P_k , $k = 1, 2, 3, \dots$ when

$$(i) \quad d_k = \begin{cases} 0, & k = 0 \\ D, & k = 1, 2, 3, \dots \end{cases}$$

$$(ii) \quad d_k = \begin{cases} 0, & k = 0, 2, 4, \dots \\ D, & k = 1, 3, 5, \dots \end{cases}$$

$$(iii) \quad d_k = \begin{cases} 0, & k = 0 \\ 2D, & k = 1, 3, 5, \dots \\ -D, & k = 2, 4, 6, \dots \end{cases}$$

- 4.68 Figure E4.68 shows the relationship between acceleration, velocity, and position of a particle moving along a straight line.

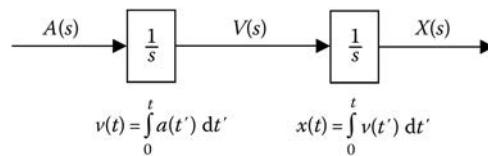


FIGURE E4.68

- a. Write the differential equations relating $v(t)$ and $a(t)$, $x(t)$ and $v(t)$, and $x(t)$ and $a(t)$.
- b. Use trapezoidal integration to approximate the three differential equations. That is, find the difference equations relating v_k and a_k , x_k and v_k , and x_k and a_k .
- c. Find the poles of the three transfer functions $V(z)/A(z)$, $X(z)/V(z)$, and $X(z)/A(z)$. Comment on the stability of each.
- d. Suppose the acceleration is given by

$$a(t) = \begin{cases} t, & 0 \leq t < 1 \\ 1, & 1 \leq t < 2 \\ 3-t, & 2 \leq t < 3 \\ 0, & 3 < t \end{cases}$$

Find analytical expressions for $v(t)$ and $x(t)$.

- e. Solve the difference equations recursively for a suitable value of T . Plot $v(t)$, $t \geq 0$ along with v_k , $k = 0, 1, 2, 3, \dots$ on the same graph, and do the same for $x(t)$, $t \geq 0$ and x_k , $k = 0, 1, 2, 3, \dots$
- 4.69 The motion of a mass suspended from a spring without friction is governed by $md^2x/dt^2 + kx = f$, where $f = f(t)$ is the applied force acting on the mass.
- a. Find the transfer function $H(s) = X(s)/F(s)$ in terms of the natural frequency $\omega_n = \sqrt{k/m}$ and the constant $c = 1/m$. Where are the poles located?
- b. Use explicit Euler, implicit Euler, and trapezoidal integration to obtain discrete-time approximations, that is, find $H(z) = X(z)/F(z)$. Leave your answers in terms of c , ω_n , and the integration step size T .

- c. Find the poles for each z -domain transfer function $H(z)$ in part (b). Comment on the stability in each case.
- d. Let $m = 1$ slug, $k = 0.5$ lb/in., $x(0) = 2$ in., $\dot{x}(0) = 0$ in./s, and $f(t) = 0$, $t \geq 0$. Find and graph the continuous-time response $x(t)$.
- e. Choose the integration step T , so that $\omega_n T = 0.01$. Find the poles of each transfer function $H(z)$ and the discrete-time responses x_k , $k = 0, 1, 2, 3, \dots$ for the same conditions in part (d). Plot the discrete-time responses on the same graph as $x(t)$.

4.9 FREQUENCY RESPONSE OF DISCRETE-TIME SYSTEMS

By now, it should be apparent that the methods for describing and analyzing the behavior of LTI continuous-time and discrete-time systems are similar. Indeed, both types of systems possess a natural response, independent of the system's input (or inputs), and similar in form to the impulse response of the system. The impulse response and the system transfer function form a Laplace transform pair for continuous-time systems and a z -transform pair in the case of discrete-time systems. The forced response of each is expressible by convolution, an integral for continuous-time systems and a sum for discrete-time systems.

BIBO stable systems are characterized by the location of transfer function poles in the s and z complex planes. Alternatively, the impulse response function of the continuous-time system and the impulse response (weighting sequence) of the discrete-time system satisfy equivalent types of constraints (in integral or summation form) when the systems are stable. In both instances, stability is an inherent property requiring the asymptotic decay of the natural modes.

The analogy continues into the realm of frequency response. The response of discrete-time systems to sinusoidal inputs characterizes the system dynamics in the same way as it does for continuous-time systems. Moreover, methods based on frequency response often play a critical role in the overall design of a discrete-time system.

4.9.1 STEADY-STATE SINUSOIDAL RESPONSE

We begin by considering the response of a stable LTI discrete-time system with z -domain transfer function $H(z)$ to the sinusoidal input

$$u_k = \sin k\omega T, \quad k = 0, 1, 2, \dots \quad (4.641)$$

The z -transform of the output y_k , $k = 0, 1, 2, \dots$, is expressed as

$$Y(z) = H(z)U(z) = H(z) \left[\frac{(\sin \omega T)z}{z^2 - (2 \cos \omega T)z + 1} \right] \quad (4.642)$$

$$= H(z) \left[\frac{(\sin \omega T)z}{(z - e^{j\omega T})(z - e^{-j\omega T})} \right] \quad (4.643)$$

A partial fraction expansion of $Y(z)$ in Equation 4.643 includes the first two terms shown in Equation 4.644 along with additional terms resulting from the poles of $H(z)$.

$$Y(z) = c_1 \frac{z}{z - e^{j\omega T}} + c_2 \frac{z}{z - e^{-j\omega T}} + \{\text{terms due to poles of } H(z)\} \quad (4.644)$$

where c_1 and c_2 are a complex conjugate pair. The constant c_1 is obtained from

$$c_1 \frac{z - e^{j\omega T}}{z} \left[H(z) \frac{(\sin \omega T)z}{(z - e^{j\omega T})(z - e^{-j\omega T})} \right]_{z=e^{j\omega T}} = H(e^{j\omega T}) \frac{\sin \omega T}{e^{j\omega T} - e^{-j\omega T}} \quad (4.645)$$

From Euler's identity, $e^{j\theta} = \cos \theta + j \sin \theta$, the denominator reduces to $2j \sin \omega T$, and the constants c_1 and c_2 become

$$c_1 = \frac{H(e^{j\omega T})}{2j}, \quad c_2 = \bar{c}_1 = \frac{H(e^{j\omega T})}{-2j} \quad (4.646)$$

Combining Equations 4.644 and 4.646 yields

$$Y(z) = \frac{H(e^{j\omega T})}{2j} \left[\frac{z}{z - e^{j\omega T}} \right] - \frac{H(e^{-j\omega T})}{2j} \left[\frac{z}{z - e^{-j\omega T}} \right] + \{\text{terms due to poles of } H(z)\} \quad (4.647)$$

$H(e^{j\omega T})$ and $H(e^{-j\omega T})$ are complex conjugates. In polar form,

$$H(e^{j\omega T}) = M e^{j\theta}, \quad H(e^{-j\omega T}) = M e^{-j\theta} \quad (4.648)$$

where

$$M = |H(e^{j\omega T})|, \quad \theta = \text{Arg}\{H(e^{j\omega T})\} \quad (4.649)$$

Substituting both parts of Equation 4.648 into Equation 4.647 gives

$$Y(z) = \frac{M}{2j} \left[\frac{e^{j\theta} z}{z - e^{j\omega T}} - \frac{e^{-j\theta} z}{z - e^{-j\omega T}} \right] + \{\text{terms due to poles of } H(z)\} \quad (4.650)$$

Inverting $Y(z)$ produces the discrete-time system response

$$y_k = \frac{M}{2j} [e^{j\theta} (e^{j\omega T})^k - e^{-j\theta} (e^{-j\omega T})^k] + z^{-1} \{\text{terms due to poles of } H(z)\} \quad (4.651)$$

$$= M \left[\frac{e^{j(k\omega T + \theta)} - e^{-j(k\omega T + \theta)}}{2j} \right] + z^{-1} + \{\text{terms due to poles of } H(z)\} \quad (4.652)$$

The first term is equal to $M \sin(k\omega T \pm \theta)$, and $z^{-1} \{\text{terms due to poles of } H(z)\}$ is the transient response, which decays to zero at steady state for a stable system. Hence, at steady state, the response of a stable LTI discrete-time system with transfer function $H(z)$ to the sinusoidal input in Equation 4.641 is

$$(y_k)_{ss} = M \sin(k\omega T + \theta) \quad (4.653)$$

$$= |H(e^{j\omega T})| \sin[k\omega T + \text{Arg}\{H(e^{j\omega T})\}] \quad (4.654)$$

$H(e^{j\omega T})$ is the discrete-time frequency response function of the system. Note the dependence of M and θ in Equation 4.649 on the period T as well as on the frequency ω .

An expression similar to Equation 4.654 applies to LTI continuous-time systems (see Equation 4.304). Once the transient response has vanished, the steady-state response consists of a sinusoid at the same frequency as the input. The frequency response function, $H(j\omega)$, in the case of continuous-time systems, and $H(e^{j\omega T})$ for discrete-time systems, establishes the frequency dependent amplitude and phase shift in the steady-state response.

Linear systems are effectively filters that pass certain frequency components in their inputs more readily than others. Analog and digital filters are examples of how a system designer can exploit frequency response to produce a system with desirable frequency discrimination characteristics.

Frequency response is important in the study of continuous-time system simulation. It allows us to characterize the dynamic errors of a discrete-time system model intended to approximate (simulate) the dynamics of a continuous-time system. In particular, the frequency response function $H_1(e^{j\omega T})$ of a numerical integrator can be compared with the frequency response function of a continuous-time integrator $H_1(j\omega) = (1/s)|_{s=j\omega} = 1/j\omega$. For the most part, this is deferred until [Chapter 8](#); however, we will lay some of the groundwork for what is to come later in this section.

4.9.2 PROPERTIES OF THE DISCRETE-TIME FREQUENCY RESPONSE FUNCTION

There are several important properties of $H(e^{j\omega T})$ worthy of discussion. First and foremost is its periodic nature. The argument $e^{j\omega T}$ is a unit vector beginning at $(1,0)$ in the complex plane when $\omega = 0$. As the frequency ω increases, the vector rotates counterclockwise around the Unit Circle as shown in [Figure 4.72](#). At the sampling frequency $\omega_s = 2\pi/T$, the unit vector has completed one revolution, that is,

$$H(e^{j\omega_s T}) = H(e^{j(2\pi/T)T}) = H(e^{j2\pi}) = H(e^{j0}) = H(1) \quad (4.655)$$

The complex values of $H(e^{j\omega T})$ generated by the first revolution around the Unit Circle are repeated during subsequent revolutions, that is, as ω increases from $k\omega_s$ to $(k+1)\omega_s$, $k = 1, 2, 3, \dots$. In mathematical terms, the periodicity property is

$$H(e^{j(\omega + k\omega_s)T}) = H(e^{j\omega T}), \quad k = 1, 2, 3, \dots \quad (4.656)$$

Another property of the frequency response function is the symmetry of $|H(e^{j\omega T})|$ about the angles $\omega T = \pi, 3\pi, 5\pi, \dots$, that is, the magnitude of $|H(e^{j\omega T})|$ is symmetric or folded about the frequencies $\omega = \pi/T, 3\pi/T, 5\pi/T, \dots$. $|H(e^{j\omega T})|$ is referred to as “even” function about the radian frequencies $\omega = 0.5\omega_s, 1.5\omega_s, 2.5\omega_s, \dots$. In mathematical terms,

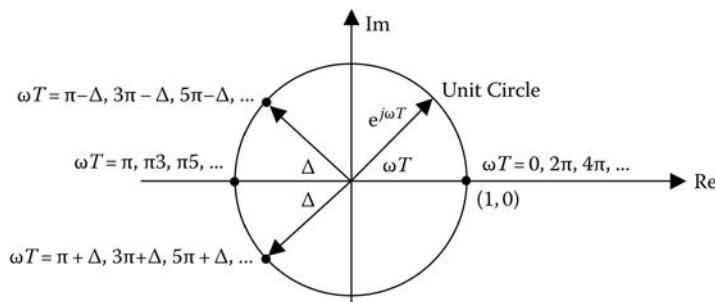


FIGURE 4.72 Rotation of unit vector $e^{j\omega T}$ around the Unit Circle.

$$|H(e^{j[n\omega_s + \Delta]})| = |H(e^{j[n\omega_s - \Delta]})|, \quad n = 0.5, 1.5, 2.5, \dots \quad (4.657)$$

The phase of $H(e^{j\omega T})$ is an “odd” function about the same frequencies, that is,

$$\text{Arg } H(e^{j[n\omega_s + \Delta]}) = -\text{Arg } H(e^{j[n\omega_s - \Delta]}), \quad n = 0.5, 1.5, 5.5, \dots \quad (4.658)$$

Due to the periodic behavior of $H(e^{j\omega T})$, it is unnecessary to plot $|H(e^{j\omega T})|$ and $\text{Arg}[H(e^{j\omega T})]$ outside ($0 \leq \omega < \omega_s$). In fact, it is customary to draw a Bode plot of $H(e^{j\omega T})$ from a lower frequency (greater than or equal to zero) up to the so-called Nyquist frequency $\omega_N = 0.5\omega_s$, from the theory of sampling. We shall have more to say about the sampling theorem after the following example, which illustrates the frequency response function of a discrete-time system and its aforementioned properties.

EXAMPLE 4.33

A first-order system described by $\tau dy/dt + y = u$ is to be simulated using explicit Euler integration with a step size T .

- Find the z-domain transfer function $H(z)$ of the resulting discrete-time system.
- Plot the magnitude (in db) and phase components of the frequency response function $H(e^{j\omega T})$ over the interval $0.1 \leq \omega \leq 0.5\omega_s$. The system time constant τ is 0.25 s, and the step size T is chosen according to $T/\tau = 0.1$.
- Find the transient and steady-state response of the continuous-time system when the input is given by $u(t) = 2\sin 3t$, $t \geq 0$.
- Use the discrete-time frequency response function $H(e^{j\omega T})$ to determine the steady-state output of the simulated (discrete-time) system. Verify the results graphically.
- Compare the steady-state sinusoidal responses of the continuous- and discrete-time systems.
- The transfer function of the continuous-time system is

$$H(s) = \frac{1}{\tau s + 1} \quad (4.659)$$

The z-domain transfer function of the discrete-time system approximation obtained by the use of explicit Euler integration is

$$H(z) = \left. \frac{1}{\tau s + 1} \right|_{s \leftarrow (z-1)/T} = \frac{T/\tau}{z - 1 + (T/\tau)} \quad (4.660)$$

- The frequency response function of the discrete-time system is given by

$$H(e^{j\omega T}) = \left. \frac{T/\tau}{z - 1 + (T/\tau)} \right|_{z \leftarrow e^{j\omega T}} = \frac{0.1}{e^{j\omega T} - 0.9} \quad (4.661)$$

$$= \frac{0.1}{(\cos \omega T - 0.9) + j \sin \omega T} \quad (4.662)$$

The magnitude function is

$$|H(e^{j\omega T})| = \frac{0.1}{[(\cos \omega T - 0.9)^2 + \sin^2 \omega T]^{1/2}} \quad (4.663)$$

Using the trigonometric identity $\sin^2 \theta + \cos^2 \theta = 1$, the magnitude function becomes

$$|H(e^{j\omega T})| = \frac{0.1}{[1.81 - 1.8 \cos \omega T]^{1/2}} \quad (4.664)$$

From Equation 4.662, the phase angle of $H(e^{j\omega T})$ is

$$\text{Arg}[H(e^{j\omega T})] = -\tan^{-1} \left(\frac{\sin \omega T}{\cos \omega T - 0.9} \right) \quad (4.665)$$

The sampling frequency $\omega_s = 2\pi/T = 80\pi$ rad/s. The Nyquist frequency $\omega_N = 0.5\omega_s = 40\pi = 125.67$ rad/s. "Ch4_Ex4_33.m" contains the MATLAB code to generate the Bode plot shown in Figure 4.73.

- c. The continuous-time response $y(t)$ to the input $u(t) = 2 \sin 3t$, $t \geq 0$ is obtained from the transfer function of the continuous-time system

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1}{\tau s + 1} = \frac{1}{0.25s + 1} \quad (4.666)$$

$$Y(s) = H(s)U(s) = \frac{1}{0.25s + 1} \left[2 \left(\frac{3}{s^2 + 9} \right) \right] \quad (4.667)$$

$$= \frac{24}{(s + 4)(s^2 + 9)} \quad (4.668)$$

Inverting $Y(s)$ by partial fractions leads to

$$y(t) = \frac{24}{25} \left(e^{-t/0.25} + \frac{4}{3} \sin 3t - \cos 3t \right) \quad (4.669)$$

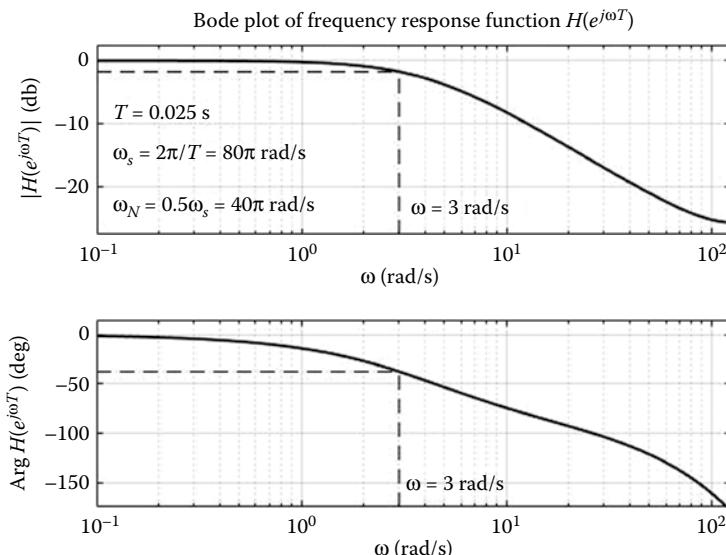


FIGURE 4.73 Bode plot for system with frequency response function $H(e^{j\omega T})$ in Equation 4.661.

The transient and steady-state components of $y(t)$ are

$$y_{tr} = \frac{24}{25} e^{-t/0.25}, \quad y_{ss} = \frac{24}{25} \left(\frac{4}{3} \sin 3t - \cos 3t \right) \quad (4.670)$$

- d. The magnitude and phase of the discrete-time frequency response function at $\omega = 3$ rad/s and $T = 0.025$ s are obtained from Equations 4.664 and 4.665, respectively.

$$\left| H(e^{j3(0.025)}) \right| = \frac{0.1}{[1.81 - 1.8 \cos 3(0.025)]^{1/2}} = 0.815 \quad (4.671)$$

$$\text{Arg} \left| H(e^{j3(0.025)}) \right| = -\tan^{-1} \left[\frac{\sin 3(0.025)}{\cos 3(0.025) - 0.9} \right] = -0.657 \text{ rad} \quad (4.672)$$

The dashed lines in Figure 4.73 show the gain $20 \log(0.815) = -1.78$ db and phase angle $-0.657 \text{ rad} = -37.63^\circ$ at $\omega = 3 \text{ rad/s}$. For a sinusoidal input with magnitude $|u(t)| = 2$, the discrete-time system output at steady state is from Equation 4.654

$$(y_k)_{ss} = 2[0.815 \sin(0.075k - 0.657)] \quad (4.673)$$

A graph of u_k and $(y_k)_{ss}$ is shown in Figure 4.74.

The steady-state output component lags the input by 37.63° , and its amplitude is

$$2 |H(e^{j3(0.025)})| = 2(0.815) = 1.63.$$

In order to verify the frequency response values in Equations 4.671 and 4.672, a blown-up portion of Figure 4.74 near consecutive peaks is shown in Figure 4.75.

The amplitude of $(y_k)_{ss}$ is in agreement with the predicted value of 1.63. The peak amplitudes occur at approximately $k = 105$ for u_k and $k = 114$ for $(y_k)_{ss}$. The measured

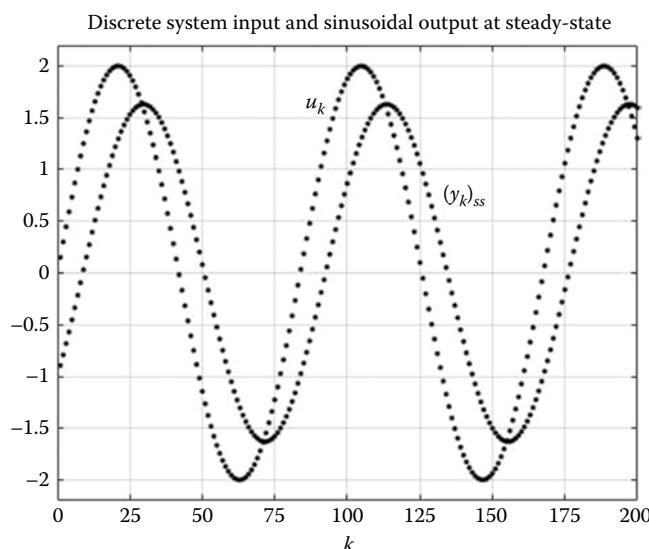


FIGURE 4.74 Sinusoidal input and steady-state sinusoidal output of discrete-time system.

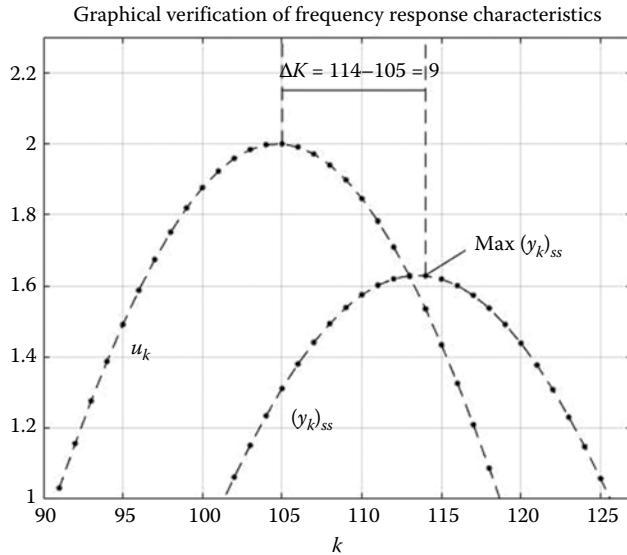


FIGURE 4.75 Graphical enlargement of Figure 4.74 for verifying frequency response characteristics.

phase lag is $\Delta k\omega T = (114 - 105)(3 \text{ rad/s})(0.025 \text{ s}) = 0.675 \text{ rad}$, which compares favorably with the analytical value of 0.657 rad.

- e. The continuous-time system response $y(t)$ in Equation 4.669 for $0 \leq t \leq 5$ and $(y_k)_{ss}$, the steady-state component of the discrete-time system output, are plotted in Figure 4.76.

Note the agreement between the two outputs once the continuous-time transient response has vanished.

The discrete-time system transient component cannot be obtained solely from the discrete-time frequency response function $H(e^{j\omega T})$. However, from Equation 4.660, we know it has the form cp^k where $p = 1 - T/\tau = 0.9$. Finding the constant c requires partial

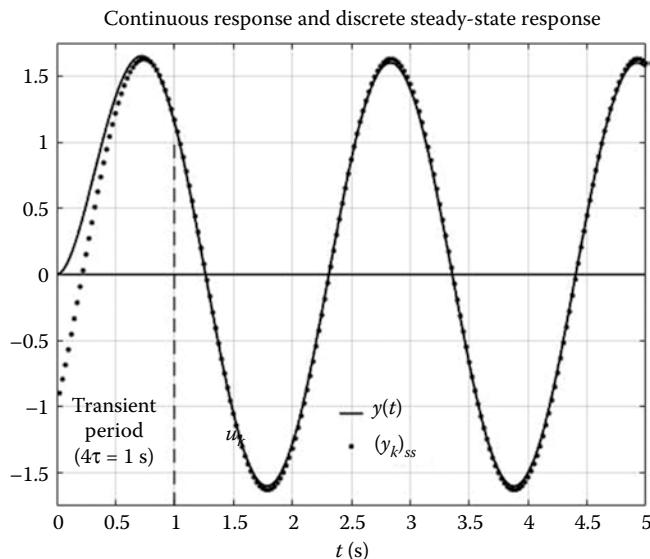


FIGURE 4.76 Continuous-time system response and sinusoidal steady-state response of discrete-time system obtained using frequency response function.

fraction expansion of $Y(z) = H(z)U(z)$ where $U(z)$ is the z-transform of the discrete-time input $u_k = 2 \sin k\omega T, k = 0, 1, 2, \dots$

4.9.3 SAMPLING THEOREM

The Bode plot in [Figure 4.73](#) displays the frequency response characteristics of the discrete-time system used to simulate a first-order continuous-time system. $H(e^{j\omega T})$ is a periodic function with period $\omega_s = 2\pi/T$. The maximum frequency on Bode plots of discrete-time systems is generally limited to $\omega_N = \omega_s/2 = \pi/T$, where ω_N is called the Nyquist frequency. The limitation is based on the sampling theorem, which we shall explore in a very rudimentary fashion.

First, let us verify the periodic nature of $H(e^{j\omega T})$ as well as its symmetry about the frequencies $0.5\omega_s, 1.5\omega_s, 2.5\omega_s, \dots$ or equivalently $\omega_N, 3\omega_N, 5\omega_N, \dots$. [Figure 4.77](#) shows the magnitude $|H(e^{j\omega T})|$ and phase $\text{Arg}[H(e^{j\omega T})]$ corresponding to the frequency response function $H(e^{j\omega T})$ in Equation 4.662. The sampling frequency $\omega_s = 2\pi/T = 251.3$ rad/s and the plots in [Figure 4.77](#) extend for three periods, that is $(0 \geq \omega < 3\omega_s)$. The symmetry of $H(e^{j\omega T})$ in both magnitude and phase about the Nyquist frequency, $\omega_N = \omega_s/2 = 125.7$ rad/s, $3\omega_N, 5\omega_N, 7\omega_N, \dots$ is evident.

The sampling theorem, as the name suggests, applies to sample data systems where a continuous-time signal is periodically sampled. The theorem, however, has important ramifications for discrete-time systems and continuous system simulation.

Continuous-time sinusoids $\sin \omega t$ are sampled every T s as shown in [Figure 4.78](#). In the top frame, the frequency of the sinusoid is $\omega = 0.2\pi$ rad/s, and sampling occurs at a rate of one sample per second ($\omega_s = 2\pi$ rad/s). The period of the sinusoid is $2\pi/\omega = 10$ s, and the sampling rate of 10 samples per period is sufficient to reconstruct the original sinusoid.

In the middle plot, the frequency of the continuous-time sinusoid is 1.4π rad/s while the sampling remains fixed at 2π rad/s. The sampled points appear to come from the lower frequency ($\omega = 0.4\pi$ rad/s) continuous-time sinusoid shown in dashed form. In the bottom graph, the continuous-time sinusoid has a frequency of 2.2π rad/s. Sampling it at the rate of 2π rad/s produces the

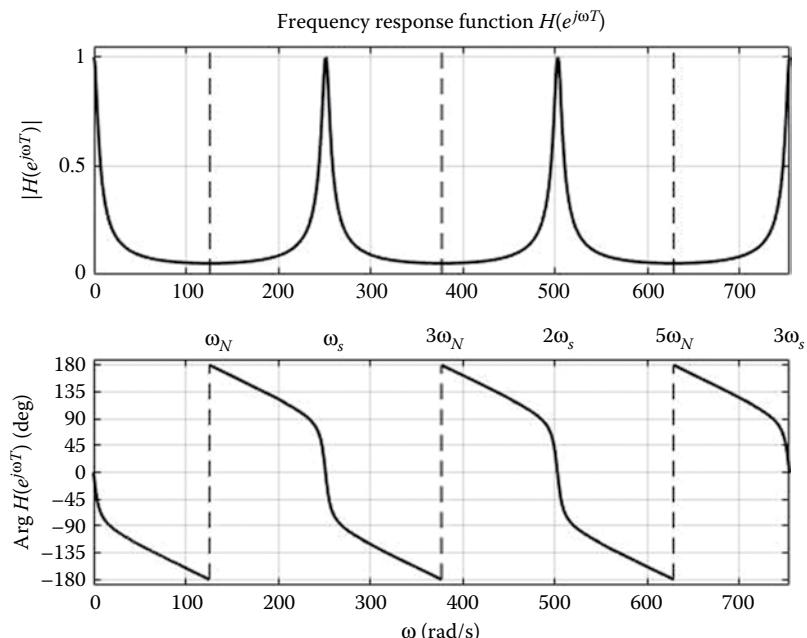


FIGURE 4.77 (a) Periodic nature and (b) symmetry of frequency response function $H(e^{j\omega T})$.

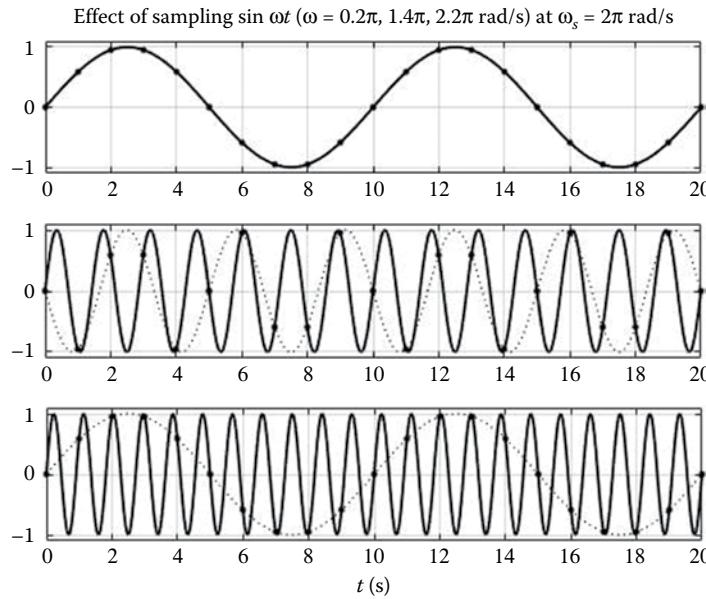


FIGURE 4.78 Illustration of aliasing of a sampled sinusoid.

identical set of data points obtained in the top graph making it appear as if the continuous-time sinusoid being sampled is the one at $\omega = 0.2\pi$ rad/s.

There is no ambiguity in identifying the correct sinusoid being sampled provided sampling occurs more than twice as fast as the frequency of the sinusoid, that is, $\omega_s > 2\omega$. In other words, the sampled points uniquely determine any continuous-time sinusoid whose frequency is less than one half the sampling frequency, that is, $\omega < \omega_s/2$.

By definition, the Nyquist frequency is $\omega_N = \omega_s/2$. Hence, for a given sampling frequency ω_s , only sinusoids with frequency less than the Nyquist frequency can be distinguished from lower frequency sinusoids. A sinusoid with frequency greater than the Nyquist frequency will be “aliased” into a lower frequency sinusoid as in the last two cases shown in Figure 4.78. This explains why Bode plots of discrete-time frequency response functions range from a lower frequency up to the Nyquist frequency ω_N .

The sampling theorem applies to sampling of continuous-time signals in general. Aliasing occurs when $\omega_s \leq 2\omega_0$, where ω_0 represents the highest frequency present in the band-limited signal. In terms of the sampling period, $T < \pi/\omega_0$ to prevent aliasing. The sampling theorem presents a formula, albeit difficult to implement, for reconstructing the band-limited continuous-time signal from the numerical values of the samples (Cadzow 1973).

The sampling theorem extends to simulation of continuous-time systems. The sampling interval T becomes the integration step size. Continuous-time inputs to the differential equations are sampled in the process of generating the discrete-time inputs to the difference equations. Consequently, the frequency content of the continuous-time input signals influences the choice of appropriate step size in the simulation.

EXAMPLE 4.34

The continuous-time first-order system in Example 4.33 is to be simulated using trapezoidal integration

- Find the z-domain transfer function of the discrete-time system and the difference equation. Leave your answers in terms of the continuous-time system time constant τ and the integration step size T .

- b. Find the sampling frequency and Nyquist frequency when $\tau = 5$ s and $T = 0.25$ s.
- c. Find the continuous-time output $y(t)$ when the input $u(t) = \sin \omega t$, $t \geq 0$.
- d. Plot the continuous-time and discrete-time outputs on the same graph when

$$(i) \omega = \pi \text{ rad/s} \quad \omega = 7\pi \text{ rad/s} \quad \omega = 8\pi \text{ rad/s.}$$

- e. Compare $H(j\omega)$ and $H(e^{j\omega T})$ at $\omega = \pi$, 7π , and 8π rad/s.

- a. The z-domain transfer function of the discrete-time system is

$$H(z) = H(s) \Big|_{s \leftarrow (2/T)((z-1)/(z+1))} = \frac{1}{\tau s + 1} \Big|_{s \leftarrow (2/T)((z-1)/(z+1))} \quad (4.674)$$

$$= \frac{1}{\tau[(2/T)((z-1)/(z+1))] + 1} \quad (4.675)$$

$$= \frac{T(z+1)}{(2\tau+T)z - (2\tau-T)} \quad (4.676)$$

$$\Rightarrow H(z) = \frac{Y(z)}{U(z)} = \frac{T(1+z^{-1})}{(2\tau+T) - (2\tau-T)z^{-1}} \quad (4.677)$$

Inverting Equation 4.677 produces the difference equation

$$(2\tau+T)y_k - (2\tau-T)y_{k-1} = T(u_k + u_{k-1}), \quad k = 1, 2, 3, \dots \quad (4.678)$$

- b. $\omega_s = 2\pi/T = 2\pi/0.25 = 8\pi$ rad/s, $\omega_N = \omega_s/2 = 8\pi/2 = 4\pi$ rad/s.
- c. The continuous-time output $y(t)$ is obtained by inverse Laplace transformation of

$$Y(s) = H(s)U(s) = \frac{1}{\tau s + 1} \left[\frac{\omega}{s^2 + \omega^2} \right] \quad (4.679)$$

Following partial fraction expansion and inverse Laplace transformation, the result is

$$y(t) = \frac{\tau\omega}{1 + (\tau\omega)^2} \left[e^{-t/\tau} - \cos \omega t + \frac{1}{\tau\omega} \sin \omega t \right] \quad (4.680)$$

- d. Substituting $\tau = 5$ s, $\omega = \pi$, 7π , and 8π rad/s gives the continuous-time output for the three cases enumerated. The simulated output is obtained by recursive solution of the difference equation after solving explicitly for y_k in Equation 4.678.

$$y_k = \frac{1}{2\tau+T} [(2\tau-T)y_{k-1} + T(u_k + u_{k-1})], \quad k = 1, 2, 3, \dots \quad (4.681)$$

The continuous- and discrete-time outputs are evaluated in "Ch4_Ex4_34.m." Plots of $y(t)$, $t \geq 0$ and y_k , $k = 0, 1, 2, \dots$ for the three input sinusoids are presented in Figures 4.79 through 4.81 along with the continuous- and discrete-time inputs.

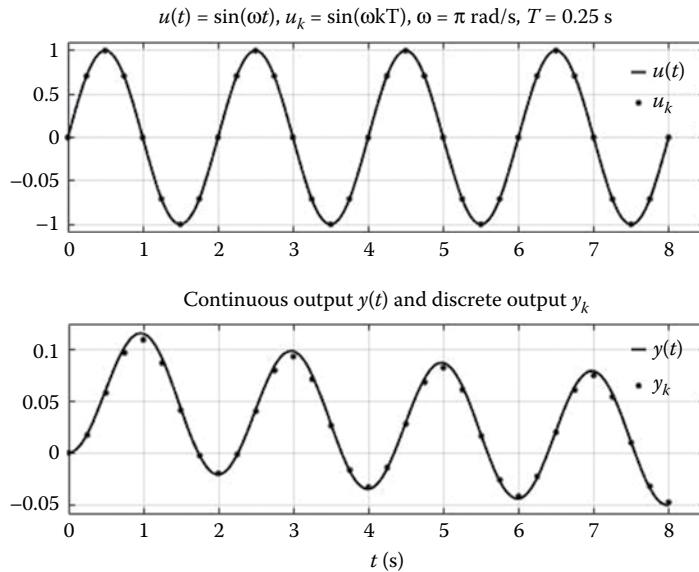


FIGURE 4.79 Continuous- and discrete-time inputs and outputs ($\omega = \pi$ rad/s).

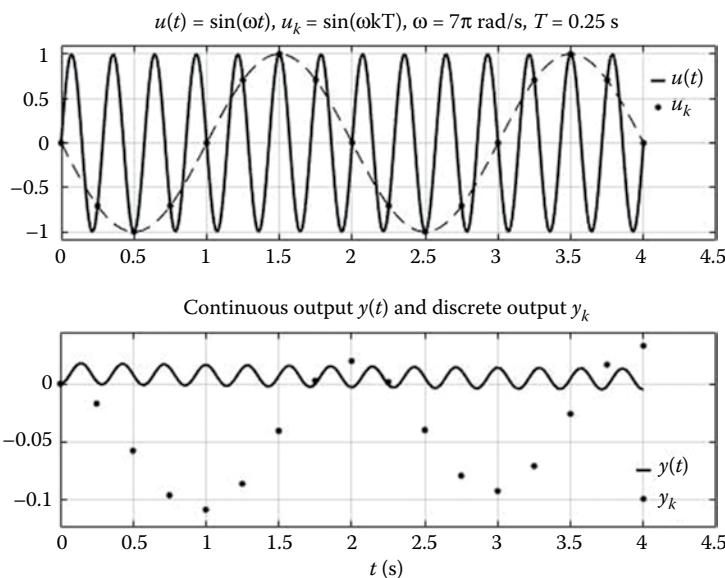


FIGURE 4.80 Continuous- and discrete-time inputs and outputs ($\omega = 7\pi$ rad/s).

In Figure 4.79, the input frequency $\omega = \pi$ rad/s is well below the Nyquist frequency $\omega_N = 4\pi$ rad/s. Sampled values of the discrete-time input $u_k = \sin(k\omega T)$ are an accurate reflection of the continuous-time sinusoidal input $u(t) = \sin \omega t$. As a result, the continuous-time response $y(t)$ and simulated response y_k are in close agreement at the sample times $t_k = kT$, $k = 0, 1, 2, \dots$.

In Figure 4.80, $\omega = 7\pi$ rad/s exceeds the Nyquist frequency $\omega_N = 4\pi$ rad/s. The simulated output is the response to the alias term whose frequency is π rad/s (shown dotted in Figure 4.80). Understandably, the simulated response y_k bears no resemblance to the continuous-time response $y(t)$.

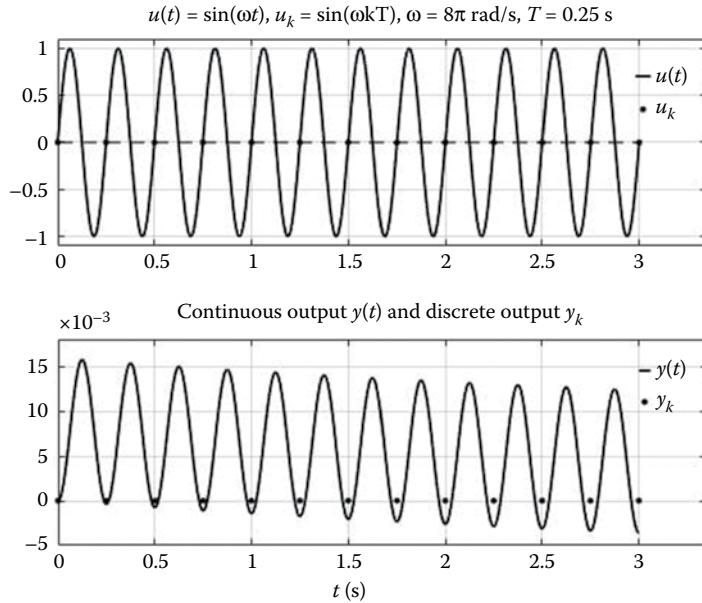


FIGURE 4.81 Continuous- and discrete-time system inputs and outputs ($\omega = 8\pi$ rad/s).

In Figure 4.81, the sampling frequency is the same as the frequency of the sinusoid, that is, $\omega_s = \omega = 8\pi$ rad/s. The same value of zero is sampled once per cycle making the effective input to the discrete-time system $u_k = 0$, $k = 0, 1, 2, \dots$. The simulated (discrete-time) output is identically zero as well. The continuous-time response is also shown.

- e. There are several ways to determine the magnitude and phase of the continuous-time and discrete-time frequency response functions. One is to start with $H(j\omega)$ and $H(e^{j\omega T})$ and express the magnitude and phase components in terms of ω with τ and T (in the discrete-time case) as parameters.

$$H(j\omega) = H(s)|_{s=j\omega} = \frac{1}{\tau s + 1} \Big|_{s=j\omega} = \frac{1}{\tau j\omega + 1} \quad (4.682)$$

$$= \frac{1}{[(\tau\omega)^2 + 1]^{1/2}} \angle -\tan^{-1}(\tau\omega) \quad (4.683)$$

$$H(e^{j\omega T}) = H(z)|_{z=e^{j\omega T}} = \frac{T(z+1)}{(2\tau+T)z - (2\tau-T)} \Big|_{z=e^{j\omega T}} \quad (4.684)$$

$$= \frac{T(e^{j\omega T} + 1)}{(2\tau+T)e^{j\omega T} - (2\tau-T)} \quad (4.685)$$

$$= \frac{T(\cos \omega T + j \sin \omega T + 1)}{(2\tau+T)(\cos \omega T + j \sin \omega T) - (2\tau-T)} \quad (4.686)$$

$$|H(e^{j\omega T})| = \frac{T[(1 + \cos \omega T)^2 + \sin^2 \omega T]^{1/2}}{\{(2\tau+T)\cos \omega T - (2\tau-T)\}^2 + [(2\tau+T)\sin \omega T]^2} \quad (4.687)$$

$$\text{Arg}[H(e^{j\omega T})] = \tan^{-1} \left(\frac{\sin \omega T}{1 + \cos \omega T} \right) - \tan^{-1} \left(\frac{(2\tau + T)\sin \omega T}{(2\tau + T)\cos \omega T - (2\tau - T)} \right) \quad (4.688)$$

A simpler alternative for computing either frequency response function for a given frequency ω is by direct substitution of $s = j\omega$ and $z = e^{j\omega T}$. The resulting complex numbers in rectangular form are then expressed in polar form. To illustrate, consider the continuous-time frequency response function $H(j\omega)$ when $\omega = \pi$ rad/s.

$$H(j\pi) = \frac{1}{5j\pi + 1} = \frac{1 - j5\pi}{1 - j5\pi} \cdot \frac{1}{1 + j5\pi} \quad (4.689)$$

$$= \frac{1}{1 + (5\pi)^2} - j \frac{5\pi}{1 + (5\pi)^2} \quad (4.690)$$

$$= 0.0040 - j0.0634$$

$$|H(j\pi)| = [(0.0040)^2 + (-0.0634)^2]^{1/2} = 0.0635$$

$$\text{Arg}[H(j\pi)] = \tan^{-1} \left(\frac{-0.0634}{0.0040} \right) = -1.5072 \text{ rad}(-86.4^\circ)$$

The discrete-time frequency response function is evaluated in the same fashion.

$$H(e^{j\pi 0.25}) = \frac{0.25(e^{j\pi 0.25} + 1)}{[(2)(5) + 0.25]e^{j\pi 0.25} - [(2)(5) - 0.25]} \quad (4.691)$$

$$|H(e^{j\pi 0.25})| = 0.0036 - j0.0601 = [(0.0036)^2 + (-0.0601)^2]^{1/2} = 0.0602$$

$$\text{Arg}[H(e^{j\pi 0.25})] = \tan^{-1} \left(\frac{-0.0601}{0.0036} \right) = -1.5105 \text{ rad}(-86.6^\circ)$$

Table 4.10 lists the magnitude and phase of the continuous-time and discrete-time frequency response functions evaluated at $\omega = \pi$, 7π , and 8π rad/s. The DC ($\omega = 0$) values are also included. It follows from Equation 4.682 that the DC gain and phase of the continuous-time system are 1° and 0° , respectively. For the discrete-time system, the DC gain and phase are obtained from me $H(e^{j0T}) = H(1) = 1$. The period of $H(e^{j\omega T})$ is $\omega_s = 8\pi$ rad/s and its symmetric about the Nyquist frequency $\omega_N = 4\pi$ rad/s.

In order to verify the magnitudes shown in **Table 4.10**, it is necessary to extend the time scale in Figures 4.79 through 4.81. For example, in **Figure 4.79**, when the input is $u(t) = \sin \pi t$, $t \geq 0$, the continuous- and discrete-time steady-state responses are sinusoids with amplitudes in the neighborhood of 0.06 (see **Table 4.10**). Looking at **Figure 4.79**,

TABLE 4.10

Continuous- and Discrete-Time Frequency Response ($\omega = 0, \pi, 7\pi, 8\pi$)

ω , rad/s	$ H(j\omega) $	$\text{Arg}[H(j\omega)]$ (°)	$ H(e^{j\omega 0.25}) $	$\text{Arg}[H(e^{j\omega 0.25})]$ (°)
0	1	0	1	0
π	0.0635	-86.36	0.0602	-86.54
7π	0.0091	-89.48	0.0602	-86.54
8π	0.0080	-89.54	1	0

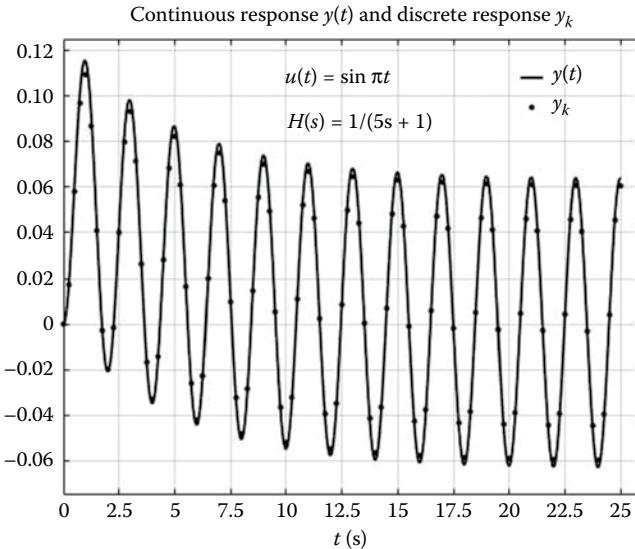


FIGURE 4.82 Continuous- and discrete-time responses showing steady state.

the continuous- and discrete-time responses have yet to reach steady state. Figure 4.82 shows both responses for a period of time equal to five time constants ($5\tau = 25$ s). The steady-state amplitudes are in agreement with the values in the table.

4.9.4 DIGITAL FILTERS

Linear discrete-time systems process signals with known frequency content in a predictable fashion. Digital filters are designed to block or pass selected frequencies present in the discrete-time inputs. Numerous references in the area of digital signal processing are available for further reading about digital filters.

Low-pass filters, as the name suggests, are intended to readily pass low frequencies and attenuate all others. Figure 4.73 showed the discrete-time frequency response function of a low-pass filter. The cut-off frequency (bandwidth) of a low-pass filter with DC gain of 0 db is the frequency at which the gain equals -3 db. A low-pass filter can be thought of as passing those frequencies within its bandwidth.

Two ways of implementing a low-pass digital filter are illustrated next.

EXAMPLE 4.35

Twenty-five years of end-of-month lake water temperature readings T_k , $k = 0, 1, 2, 3, \dots, 300$ months, are stored in MATLAB data file “*Ch4_LakeTemp.mat*.” Researchers would like to determine if the lake temperature, adjusted for monthly variations, has changed over that time.

- A moving average of the past 12 readings is used to smooth the seasonal temperature variations, that is,

$$\hat{T}_k = \frac{1}{12}[T_{k-1} + T_{k-2} + \dots + T_{k-12}], \quad k = 12, 13, \dots, 300 \quad (4.692)$$

where \hat{T}_k is the seasonally adjusted end-of-month lake temperature starting with end-of-month 12. (Note that T_0 is the lake temperature on December 31 of a given year and \hat{T}_{12} represents the seasonally adjusted lake temperature on December 31 of the following year.) Find the z-domain transfer function $H(z) = \hat{T}(z)/T(z)$, and plot the magnitude of the discrete-time frequency response function.

- b. The period of the seasonal variation is $P = 12$ months. The frequency is $\omega_0 = 2\pi/P = \pi/6$ rad/month. Find $|H(e^{j\omega T})|$ and plot where $T = 1$ month (sampling period).
- c. Find \hat{T}_k , $k = 12, 13, \dots, 300$ and plot the values on the same graph with T_k , $k = 0, 1, 2, \dots, 300$. Estimate the yearly increase in lake temperature.
- d. A first-order low-pass digital filter with z-domain transfer function

$$H(z) \frac{\hat{T}(z)}{T(z)} = \frac{(1-\alpha)z}{z-\alpha} = \frac{(1-\alpha)}{1-\alpha z^{-1}} \quad (4.693)$$

is used to filter the monthly temperature variations and pass any low-frequency lake temperature variation with time. Plot the discrete-time frequency response magnitude and phase for $\alpha = 0.99$. Find the cutoff frequency and determine whether it is less or greater than $\omega_0 = \pi/6$ rad/month. In addition, find $|H(e^{j\omega_0 T})|$.

- e. Find the difference equation relating \hat{T}_k and T_k . Solve it recursively for \hat{T}_k , $k = 1, 2, \dots, 300$ and plot both input and output on the same graph.

- a. Taking the z-transform of Equation 4.692 gives

$$\hat{T}(z) = \frac{1}{12}[z^{-1}T(z) + z^{-2}T(z) + \dots + z^{-12}T(z)] \quad (4.694)$$

$$\Rightarrow H(z) = \frac{\hat{T}(z)}{T(z)} = \frac{1}{12}(z^{-1} + z^{-2} + \dots + z^{-12}) \quad (4.695)$$

The magnitude function is shown in Figure 4.83. The data points for generating the graph in Figure 4.83 are computed in M-file "Ch4_Ex4_35.m."

- b. From Figure 4.83, we see that the zeros of $|H(e^{j\omega T})|$ are located at $\omega_0 = \pi/6$ and multiples of ω_0 , namely, $2\pi/6, 3\pi/6, 4\pi/6, 5\pi/6, \pi, \dots$
- c. The smoothing algorithm, Equation 4.692, is applied to the monthly lake temperature data, and the results \hat{T}_k , $k = 12, 13, \dots, 300$ are plotted along with the discrete-time input T_k , $k = 0, 1, 2, \dots, 300$ in Figure 4.84.

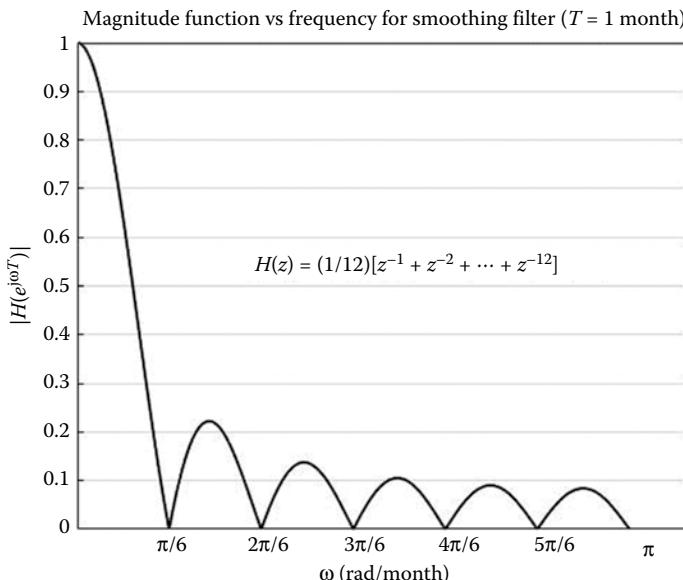


FIGURE 4.83 Magnitude function for smoothing filter.

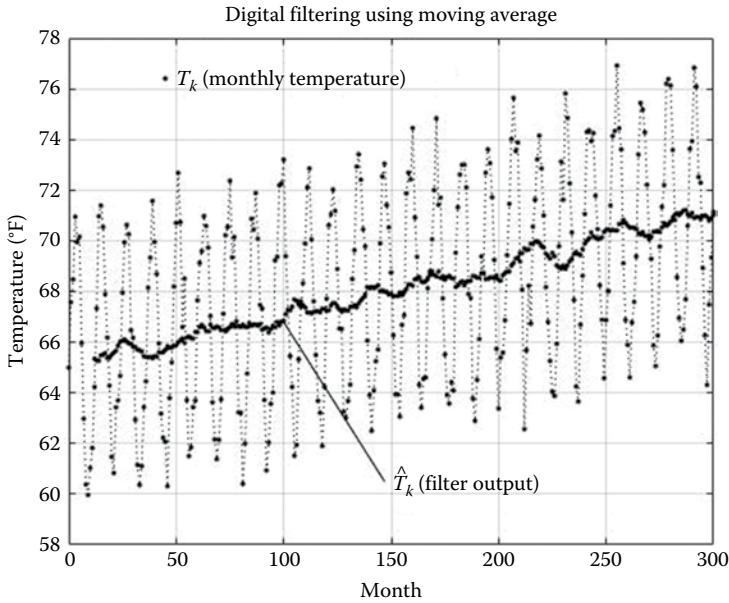


FIGURE 4.84 Input T_k , $k = 0, 1, 2, \dots, 300$ and smoothing filter output \hat{T}_k , $k = 12, 13, \dots, 300$.

The estimated annual increase in lake temperature is

$$m_l = \frac{\hat{T}_{300} - \hat{T}_{12}}{24} = \frac{71.016 - 65.353}{24} = 0.236 \frac{\text{°F}}{\text{year}} \quad (4.696)$$

- d. The gain of the filter with z-domain transfer function in Equation 4.693 and $\alpha = 0.99$ is shown in Figure 4.85.

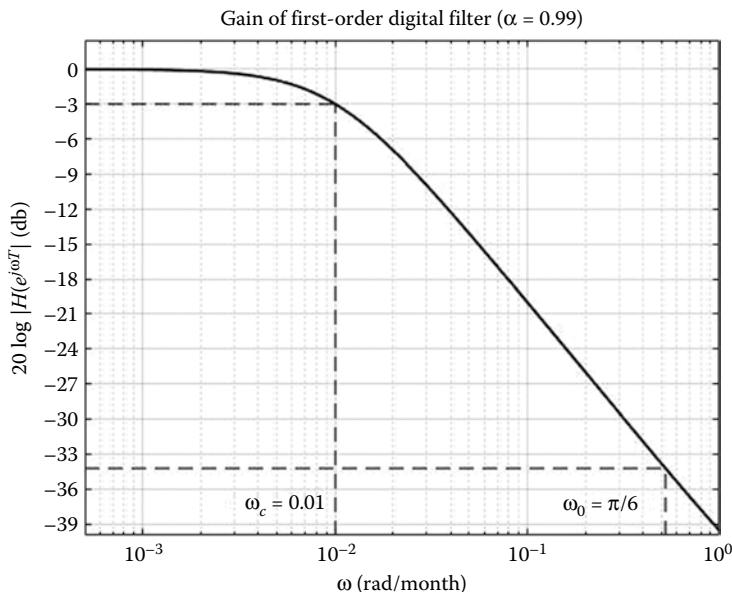


FIGURE 4.85 Gain of first-order low-pass filter with $H(z)$ in Equation 4.693.

The cutoff frequency ω_c is obtained from

$$20 \log |H(e^{j\omega_c T})| = -3 \quad (4.697)$$

It is left as an exercise problem to show that

$$\omega_c = \frac{1}{T} \cos^{-1} \left(\frac{1 + \alpha^2 - 10^{0.3+2\log(1-\alpha)}}{2\alpha} \right) \quad (4.698)$$

Substituting $\alpha = 0.99$, $T = 1$ into Equation 4.698 yields $\omega_c = 0.01$ rad/month. Referring to Figure 4.85, $\omega_0 = \pi/6$ rad/month is well beyond the cutoff frequency, and we should expect the seasonal fluctuations to be removed by the digital filter. The magnitude of $H(e^{j\omega_0 T})$ is 0.0194 (-34.2 db).

- e. The difference equation is obtained from Equation 4.693 by inverse z-transformation.

$$\hat{T}_k - \alpha \hat{T}_{k-1} = (1 - \alpha) T_k, \quad k = 1, 2, 3, \dots, 300 \quad (4.699)$$

where $\hat{T}_0 = T_0$. Equation 4.699 is solved recursively for \hat{T}_k , $k = 1, 2, 3, \dots, 300$ in "Ch4_Ex4_35.m." The discrete-time input and output are shown in Figure 4.86.

Example 4.35 illustrates the use of FIR and IIR filters. From Equation 4.695, the FIR smoothing filter impulse response is

$$h_k = \frac{1}{12} (\delta_{k-1} + \delta_{k-2} + \dots + \delta_{k-12}) \quad (4.700)$$

$$= \begin{cases} 0, & k = 0 \\ \frac{1}{12}, & k = 1, 2, \dots, 12 \\ 0, & k = 13, 14, \dots \end{cases} \quad (4.701)$$

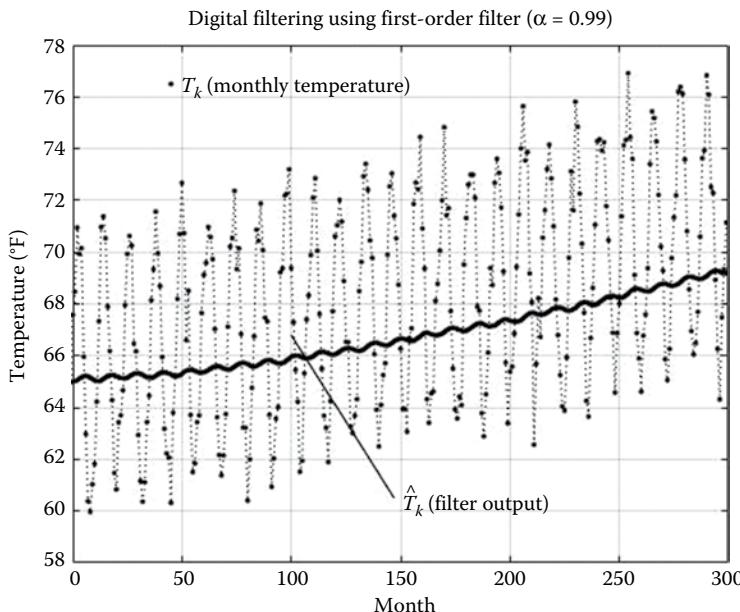


FIGURE 4.86 Input T_k , $k = 0, 1, 2, \dots, 300$ and first-order filter output \hat{T}_k , $k = 1, 2, \dots, 300$.

From Equation 4.693, the first-order IIR low-pass digital filter impulse response is

$$h_k = (1 - \alpha)\alpha^k, \quad k = 0, 1, 2, 3, \dots \quad (4.702)$$

Based on the convolution sum for the output of a discrete-time system, the FIR filter output depends solely on the past 12 inputs (not surprising) while the infinite memory IIR filter output relies on the entire set of past inputs.

Choosing $\alpha = 0.99$ places the pole of $H(z)$ precariously close to the Unit Circle, the stability boundary in the z -plane. As a consequence, discrete-time input signals with poles near $z = 0.99$, for example, a step input with pole at $z = 1$, are readily passed.

The transient response period is considerable since the natural mode $\alpha^k = 0.99^k$ takes a long while to decay to zero. In Figure 4.86, if we arbitrarily assume the transient period to be 150 months [$0.99^{150} = 0.22$], the estimated slope of the linear rise in lake temperature is computed as

$$m_2 = \frac{(\hat{T}_{300} - \hat{T}_{150})^\circ\text{F}}{(300 - 150)\text{month} \times 1/12 \text{ year/month}} = \frac{69.179 - 66.649}{12.5} = 0.202 \frac{\text{°F}}{\text{year}} \quad (4.703)$$

which is close to the value obtained using the FIR smoothing filter.

EXERCISES

- 4.70 Repeat Example 4.33 using implicit Euler instead of explicit Euler integration for approximating the continuous-time system.
- 4.71 A second-order system with damping ratio ζ and natural frequency ω_n is simulated using trapezoidal integration. The DC gain of the system is unity.
- Find the discrete-time frequency response function $H(e^{j\omega T})$. Leave your answer in terms of, ζ , ω_n , T , and ω .
 - Draw a Bode plot of $H(e^{j\omega T})$ when the continuous-time system poles are as shown in Figure E4.71. Assume $\omega_n T = 0.1$.

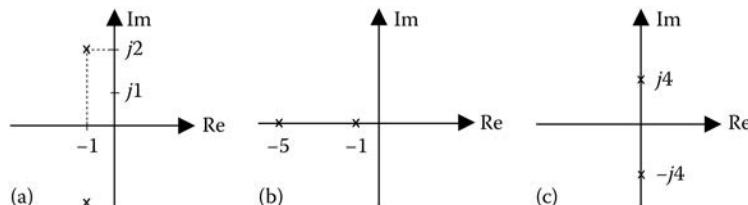


FIGURE E4.71

- 4.72 The electrical circuit shown in Figure E4.72 is that of a biquad filter, so named because the transfer function from the input to the output contains quadratic factors in the numerator and denominator. The differential equation of the circuit is

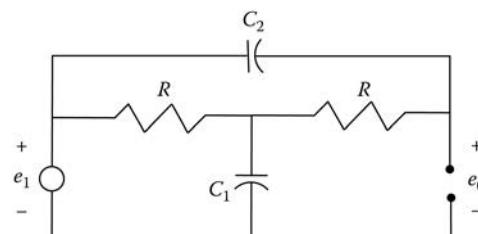


FIGURE E4.72

$$a_2\ddot{e}_0 + a_1\dot{e}_0 + a_0e_0 - b_2\ddot{e}_1 + b_1\dot{e}_1 + b_0e_1$$

where the constants a_0, a_1, a_2 and b_0, b_1, b_2 are related to R, C_1, C_2 by

$$a_0 = 1, a_1 = RC_1 + 2RC_2, a_2 = RC_1RC_2, b_0 = 1, b_1 = 2RC_2, b_2 = RC_1RC_2$$

- Find the transfer function $G(S) = E_0(S)/E_1(S)$.
- A discrete-time system approximation based on trapezoidal integration has a z -domain transfer function $G(z)$ given by

$$G(z) = \frac{\beta_2 z^2 + \beta_1 z + \beta_0}{\alpha_2 z^2 + \alpha_1 z + \alpha_0}$$

Show that

$$\beta_0 = 4\tau_1\tau_2 - 4\tau_2T + T^2, \quad \beta_1 = -8\tau_1\tau_2 + 2T^2, \quad \beta_2 = 4\tau_1\tau_2 + 4\tau_2T + T^2$$

$$\alpha_0 = 4\tau_1\tau_2 - 2(\tau_1 + 2\tau_2)T + T^2, \quad \alpha_1 = -8\tau_1\tau_2 + 2T^2$$

$$\alpha_2 = 4\tau_1\tau_2 + 2(\tau_1 + 2\tau_2)T + T^2$$

where $\tau_1 = RC_1$ and $\tau_2 = RC_2$ and T is the integration step size.

- Draw a Bode plot for the discrete-time frequency response $G(e^{j\omega T})$ when $\tau_1 = 0.1$ s, $\tau_2 = 0.001$ s, and $T = 2 \times 10^{-4}$ s.
- Fill in the following table.

$\omega, \text{ rad/s}$	$ G(j\omega) $	$\text{Arg}[G(j\omega)]$	$ G(e^{j\omega T}) $	$\text{Arg}[G(e^{j\omega T})]$
0				
5				
100				
5000				

- 4.73 An analog signal $r(t)$ is the command input to a digital control system, part of which is shown in Figure E4.73. The signal $r(t)$ must be sampled and converted to a discrete-time signal for use by the digital controller. The command input consists of a signal component $s(t)$ and a high-frequency (compared to the sampling rate $1/T_s$) noise component $n(t) = N \sin \omega t$. An antialiasing filter is inserted before sampling to eliminate aliasing in \hat{r}_k the input to the controller.

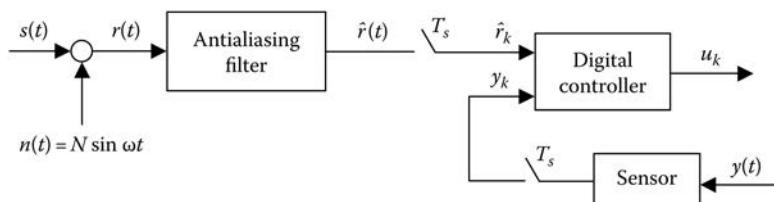
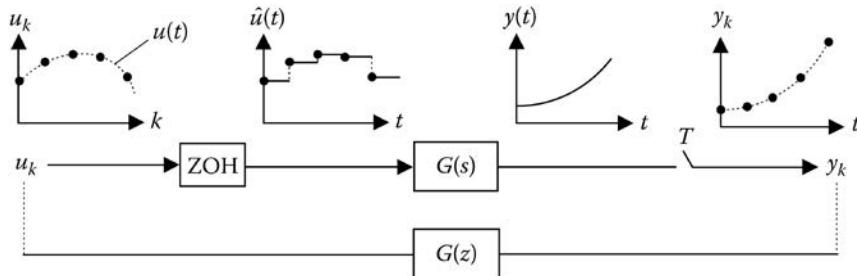


FIGURE E4.73

A fourth-order Butterworth low-pass filter is chosen. The transfer function is

$$G(s) = \frac{\hat{R}(s)}{R(s)} = \left(\frac{\omega_n^2}{s^2 + 2\cos(\pi\omega_n/8)s + \omega_n^2} \right) \left(\frac{\omega_n^2}{s^2 + 2\cos(3\pi\omega_n/8)s + \omega_n^2} \right)$$

- a. The control system sampling rate is 1000 Hz. Find the Nyquist frequency ω_N .
 b. Find ω_n so that the magnitude of $G(j\omega)$ is -60 db at the Nyquist frequency.
Hint: Use trial and error guesses for ω_n along with Bode plots until the condition $|G(j\omega_N)| = -60$ db is approximately satisfied.
- c. The signal and noise components of the command input $r(t)$ are $s(t) = 1$, $t \geq 0$ and $n(t) = 5 \times 10^{-3} \sin(2 \times 10^6 t)$, $t \geq 0$. Find the filter output $\hat{r}(t)$ at steady state.
- d. Find $G(z)$, the z -domain transfer function of the discrete-time system approximation to $G(s)$ using explicit Euler integration. Leave your answer in terms of the integration step size T .
- e. Comment on the choice of T necessary to simulate the filter response by recursive solution of the difference equation corresponding to $G(z)$.
- 4.74 A method for approximating a continuous-time system with transfer function $G(s)$ is illustrated in [Figure E4.74](#). A continuous-time input $u(t)$ is sampled every T s to produce the discrete-time input u_k . A zero-order hold (ZOH) reconstructs a piecewise continuous approximation to $u(t)$ denoted $\hat{u}(t)$, which is the input to the continuous-time system. The continuous-time output $y(t)$ is sampled every T s resulting in the discrete-time output y_k . The discrete-time system with input u_k and output y_k serves as an approximation to the continuous-time system with input $u(t)$ and output $y(t)$. The z -domain transfer function of the discrete-time system is (Jacquot)

**FIGURE E4.74**

$$G(z) = \frac{Y(z)}{U(z)} = \left(\frac{z-1}{z} \right) z \left\{ \mathcal{L}^{-1} \left[\frac{\mathcal{G}(s)}{s} \right] \right\}$$

where $z[\mathcal{L}^{-1}[G(s)/s]]$ stands for the z -transform of the discrete-time signal resulting from sampling the continuous-time signal $\mathcal{L}^{-1}[g(s)/s]$.

- a. Find the z -domain transfer function using the ZOH approximation method when the continuous-time system is first order with transfer function $G(s) = 1/(\tau s + 1)$. Leave your answer in terms of the time constant τ and sampling period T .
- b. Find the discrete-time frequency response function $G(e^{j\omega T})$, and obtain expressions for the magnitude $|G(e^{j\omega T})|$ and phase $\text{Arg}[G(e^{j\omega T})]$.
- c. Plot the magnitude and phase of $G(e^{j\omega T})$ when $\tau = 1$ s and $T = 0.1$ s.
- d. Compare the continuous- and discrete-time unit step responses and comment on the results.
- e. Find $G(e^{j\omega T})$ and $\text{Arg}[G(e^{j\omega T})]$ and compare with the values given in [Table 4.10](#) where $\tau = 5$ s and $T = 0.25$ s.
- 4.75 Derive Equation 4.698 for the cutoff frequency of the first-order low-pass digital filter with z -domain transfer function $H(z) = (1 - \alpha)z/(z - \alpha)$.

- 4.76 A notch filter is designed to attenuate input signals at one specific frequency called the notch frequency. The transfer function of a notch filter is

$$G(s) = \frac{s^2 + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (\omega_n \text{ is notch frequency})$$

- a. Find $G(z)$, the z -domain transfer function of a digital filter obtained by approximation of $G(s)$ using trapezoidal integration. Leave your answer in terms of ζ , ω_n and the integration step size T .
 - b. The digital filter is to be used to filter out the monthly lake temperature fluctuations in Example 4.35. The notch frequency is $\omega_n = \pi/6$ rad/month and the sampling period is $T = 1$ month. On the same graph, plot $|G(e^{j\omega T})|$ vs. ω from zero to the Nyquist frequency for $\zeta = 0.25, 0.5, 0.75$.
 - c. Choose the value of ζ , which produces the largest attenuation at the notch frequency, and use the digital notch filter to filter out the monthly lake temperature fluctuations in the dataset "Ch4_LakeTemp.mat." Prepare a graph similar to the ones in Figures 4.84 and 4.86.
- 4.77 The design of a digital filter calls for the placement of a pair of poles and zeros as shown in Figure E4.77.

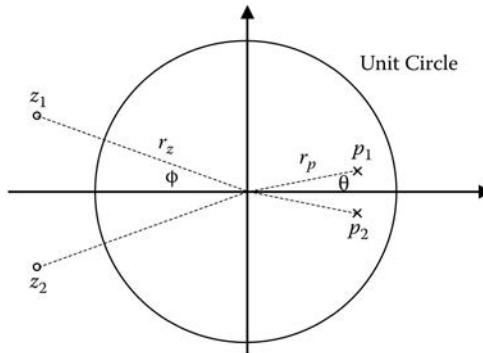


FIGURE E4.77

- a. Find the difference equation relating the filter's input $u(n)$ and output $y(n)$. The filter coefficients should be expressed in terms of r_p , θ , r_z and ϕ .
- b. Express the magnitude function $|H(e^{j\omega T})|$ in terms of the parameters r_p , θ , r_z , ϕ and the sampling time T .
- c. Plot the magnitude function for the case when $T = 1$ s, $r_p = 0.9$, $r_z = 2$, $\phi = \pi/4$ and $\theta = 0.2, 0.4, 0.6, 0.8, 1$ rad. Comment on the results.

4.10 CONTROL SYSTEM TOOLBOX

This chapter has emphasized analytical methods for obtaining continuous- and discrete-time system response to elementary types of inputs. In this section, we explore the use of MATLAB functions in the control system toolbox designed to facilitate the process of modeling and simulation of LTI dynamic systems. The control system toolbox is a supplement to MATLAB. The reader is encouraged to check out the entire suite of available functions either online or in the control system toolbox lab manual (from The Mathworks, Inc.). Many of the functions are discussed and illustrated in recent linear controls texts and companion lab manuals (D'Azzo and Houpis, 1995; Ogata 1998; Dorf and Bishop 2005).

Continuous- and discrete-time transfer functions are defined by specifying numerator and denominator polynomials in vector form. SISO and MIMO dynamic systems portrayed in block

diagram form can be reduced to obtain specific transfer functions, which can be analyzed (by other control system toolbox functions) in the time and frequency domain. Impulse and step responses as well as responses to arbitrary inputs of both types of systems are easily obtained. The z -domain transfer functions for simulating continuous-time systems based on various methods of approximation are available. Conversion between state-space and transfer function descriptions of a system is accomplished using specific toolbox commands.

This section contains some relatively simple examples of the control system toolbox functions. Exposition is kept to a minimum. For more information, the reader should check out the robust set of online interactive demos, tutorials, and case studies illustrating how the toolbox can be used to support modeling and simulation functions.

4.10.1 TRANSFER FUNCTION MODELS

Continuous- and discrete-time transfer functions are constructed using “`tf`” with proper arguments and stored as a named MATLAB object such as “`sys`.” For example, the transfer function

$$G_1(s) = \frac{(10s+1)(s+2)}{2s^4 + 5s^3 + 4s + 1} \quad (4.704)$$

is implemented by the following statements:

```
num = 25*conv ([10 1], [1 2])
den = [2 5 0 4 1]
sys_G1 = tf (num, den)
```

Note `conv ([10 1], [1 2])` produces the numerator vector [10 21 2]. A more intuitive way of creating the same transfer function is

```
s = tf ('s')
sys_G1 = 25* (10*s^2+21*s+2) / (2*s^4+5*s^3+4*s+1)
```

A discrete-time system with sampling period $T = 0.01$ s and pulse (z -domain) transfer function

$$G_2(z) = \frac{5z^2 + 3z + 2}{z^2 + 10z + 4} \quad (4.705)$$

is created from either of the two sets of statements below:

```
num = [5 3 2]; den = [1 10 4]
sys_G2 = tf (num, den, 0.01)
z = tf ('z', 0.01)
sys_G2 = (5*z^2+3*z+2) / (z^2+10*z+4)
```

The poles and zeros of a continuous- or discrete-time system transfer function are obtained using the “`pzmap (sys)`” command where “`sys`” refers to the MATLAB description of the transfer function. A pole-zero map of the transfer function $G_1(s)$ in Equation 4.704 is obtained from the command “`pzmap (sys_G1)`” and shown in [Figure 4.87](#).

The numerical values of the poles and zeros shown in [Figure 4.87](#) are returned in “`P`” and “`Z`” after issuing the command “[`P`, `Z`] = `pzmap (sys_G1)`. The result is

$$P = -2.7418, 0.2385 + 0.8475i, 0.2385 - 0.8475i, -0.2353$$

$$Z = -2.0000, -0.1000$$

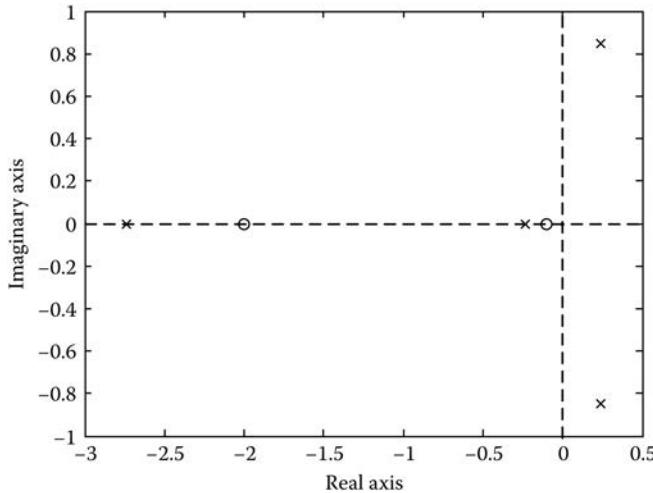


FIGURE 4.87 Pole-zero map for $G_l(s)$ in Equation 4.704.

4.10.2 STATE-SPACE MODELS

State-space models of continuous-time systems are described by matrices A , B , C , and D appearing in the state equations. The same holds for a discrete-time system, which also requires a sampling time T for a complete representation. State-space models for continuous-time systems are created using “`sys = ss (A, B, C, D)`,” while discrete-time models in state space are generated by

```
"sys = ss (A, B, C, D, T) ."
```

A continuous-time second-order system with damping ratio $\zeta = 0.5$ and natural frequency $\omega_n = 2$ rad/s was approximated using trapezoidal integration with step size $T = 0.025$ s in Section 4.7 resulting in discrete-time system state equations

$$\underline{x}_{k+1} = A\underline{x}_k + B\underline{u}_k \quad (4.706)$$

$$\underline{y}_k = C\underline{x}_k + D\underline{u}_k \quad (4.707)$$

with A , B , C , and D given in Equations 4.516 through 4.518.

The resulting matrices A , B , C , and D and sampling time T appear in the M-file “*Ch4_Tustin.m*” statement “`sys = ss (A, B, C, D, T)`” to create a discrete-time system state-space model with numerical values

<code>a =</code>		
	x1	x2
x1	1.949	-0.9512
x2	1	0
<code>b =</code>		
	u1	
x1	1	
x2	0	
<code>c =</code>		
	x1	x2

(Continued)

```

y1      0.002406   2.971e-005
d =
          ul
y1      0.0006094

```

Sampling time: 0.025 discrete-time model.

The object “sys” can be referenced by other control system toolbox commands to investigate frequency response characteristics of the discrete-time system as well as dynamic response to specific types of forcing functions. It is also instrumental in the process of converting a state-space model to a transfer function representation, the next subject of discussion.

4.10.3 STATE-SPACE/TRANSFER FUNCTION CONVERSION

The state equations for a submarine depth control system were developed in Section 2.8. The closed-loop control system is third order with three outputs, θ (stern plane angle), v (depth rate), c (depth), and a single input r (commanded depth). The MATLAB file “Ch4_sub.m” below illustrates several commands for converting between state-space models of the system and the transfer function form.

```

% Ch4_sub.m
KC = 0.6; KI = 0.1;
tau = 10; Kthd = 20; Kth = 10;
a11 = -Kthd*KC/tau; a12 = (Kth - (Kthd/tau)); a13 = Kthd*KI/tau;
a21 = -KC/tau; a22 = -1/tau; a23 = KI/tau;
a31 = -1; a32 = 0; a33 = 0;
b1 = Kthd*KC/tau; b2 = KC/tau; b3 = 1;
c11 = -KC; c12 = 0; c13 = KI;
c21 = -Kthd*KC/tau; c22 = Kth - (Kthd/tau); c23 = Kthd*KI/tau;
c31 = 1; c32 = 0; c33 = 0;
d1 = KC; d2 = Kthd*KC/tau; d3 = 0;
A1 = [a11 a12 a13; a21 a22 a23; a31 a32 a33];
B1 = [b1; b2; b3];
C1 = [c11 c12 c13; c21 c22 c23; c31 c32 c33];
D1 = [d1; d2; d3];

Sys_ss_1 = ss(A1,B1,C1,D1)% creates state-space system object for (A1,B1,C1,D1)
Sys_tf = tf(sys_ss_1)% converts state-space system object to transfer
                      function system object
[numl,denl] = % alternate method for converting state space
ss2tf(A1,B1,C1,D1)% (A1,B1,C1,D1) to transfer function
sys_ss_2 = ss(sys_tf)% converts transfer function object to state-space
                      object
[A3,B3,C3,D3] = % converts transfer function to state-space control
tf2ss(numl,denl)% canonical form with matrices (A3,B3,C3,D3)
[num2,den2] = % converts state-space (A3,B3,C3,D3) to transfer
ss2tf(A3,B3,C3,D3)% functions

```

Numerical values are assigned to matrices A_1 , B_1 , C_1 , and D_1 using the baseline system parameter values from Section 2.8. The system matrices are

A1 =	-1.2000	8.0000	0.2000
	-0.0600	-0.1000	0.0100
	-1.0000	0	0

(Continued)

B1 =	1.2000
	0.0600
	1.0000
C1 =	-0.6000 0 0.1000
	-1.2000 8.0000 0.2000
	1.0000 0 0
D1 =	0.6000
	1.2000
	0

The statement “`sys_ss_1 = ss(A1,B1,C1,D1)`” creates the object “`sys_ss_1`” associated with the continuous-time system matrices A_1 , B_1 , C_1 , and D_1 . The next statement “`sys_tf(sys_ss_1)`” creates the transfer function object “`sys_tf`” with embedded information about the three system transfer functions, one each from the command input to the three outputs. The transfer functions are displayed as

Transfer function from input to output ...

#1:	$\frac{0.6 s^3 + 0.16 s^2 + 0.01 s - 1.506e-018}{s^3 + 1.3 s^2 + 0.8 s + 0.1}$
#2:	$\frac{1.2 s^3 + 0.8 s^2 + 0.1 s - 3.474e-017}{s^3 + 1.3 s^2 + 0.8 s + 0.1}$
#3:	$\frac{1.2 s^2 + 0.8 s + 0.1}{s^3 + 1.3 s^2 + 0.8 s + 0.1}$

Note that the first two transfer functions are consistent with the control system simulation diagram ([Figure 2.55](#)), which shows direct paths from the input r to outputs θ and v . The numerator of transfer function #3 is second order due to the presence of the integrator in the path from r to c .

An alternative approach to finding the same three transfer functions uses “[numl,denl] = ss2tf(A1,B1,C1,D1).” Output matrix “`numl`” (with three rows, one for each output) stores the coefficients of the three numerator polynomials, and row vector “`denl`” contains the coefficients of the denominator, that is, characteristic polynomial. The result is

numl =	0.6000 0.1600 0.0100 0.0000
	1.2000 0.8000 0.1000 0.0000
	0 1.2000 0.8000 0.1000
denl =	1.0000 1.3000 0.8000 0.1000

Converting the transfer function of an SISO system to a state-space model is achieved using either “`ss`” or “`tf2ss`.” The command “`sys_ss_2 = ss(sys_tf)`” computes a state-space realization of the transfer function object “`sys_tf`” displayed as

a =	x1	x2	x3
x1	-1.3	-0.4	-0.1
x2	2	0	0
x3	0	0.5	0

(Continued)

```

b =
      u1
      x1    1
      x2    0
      x3    0
c =
      x1      x2      x3
      y1   -0.62   -0.235   -0.06
      y2   -0.76   -0.43    -0.12
      y3    1.2     0.4     0.1
d =
      u1
      y1    0.6
      y2    1.2
      y3    0

```

Referring to the above matrices as A_2 , B_2 , C_2 , and D_2 , it is not surprising that they differ from A_1 , B_1 , C_1 , and D_1 since the state-space model representation of a continuous-time system is not unique.

An alternative method for creating a state-space model from a transfer function is to use “[A3,B3,C3,D3] = tf2ss(num1,den1)” where “num1” and “den1” are the numerator and denominator arrays, respectively, created previously by the command “ss2tf.” This results in creation of output matrices A_3 , B_3 , C_3 , and D_3 given below:

```

A3 =   -1.3000   -0.8000   -0.1000
       1.0000     0         0
       0         1.0000     0
B3 =   1
       0
       0
C3 =   -0.6200   -0.4700   -0.0600
   -0.7600   -0.8600   -0.1200
   1.2000     0.8000   0.1000
D3 =   0.6000
       1.2000
       0

```

State-space models created by “tf2ss” are in controller canonical form (Ogata 1998).

The last statement [num2,den2] = ss2tf(A3,B3,C3,D3) in “Ch4_sub.m” converts the state-space model in controller canonical form back to the three transfer functions.

The state-space models for the submarine control system are summarized in Table 4.11. A good way of checking the results is to compute the eigenvalues of the coefficient matrices A_1 , A_2 , and A_3 in the table. The MATLAB command “eig(A)” returns the same characteristic roots, namely, $-0.5687 \pm j0.5400$ and -0.1626 , for all three matrices.

4.10.4 SYSTEM INTERCONNECTIONS

Block diagrams can be systematically reduced in complexity using control system toolbox functions such as “parallel,” “series,” and “feedback.” Consider the block diagram shown in Figure 4.88.

$$G_c(s) = 2 \left(\frac{10s+1}{2s+1} \right), \quad H(s) = \frac{1}{50s+1} \quad (4.708)$$

TABLE 4.11
Three Different State-Space Models of Submarine Depth Control System

<i>i</i>	<i>Ai</i>	<i>Bi</i>	<i>Ci</i>	<i>Di</i>
1	$\begin{bmatrix} -1.2 & 8 & 0.2 \\ -0.06 & -0.1 & 0.01 \\ -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1.2 \\ 0.06 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -0.6 & 0 & 0.1 \\ -1.2 & 8 & 0.2 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.6 \\ 1.2 \\ 0 \end{bmatrix}$
2	$\begin{bmatrix} -1.3 & -0.4 & -0.1 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.62 & -0.235 & -0.06 \\ -0.76 & -0.43 & -0.12 \\ 1.2 & 0.4 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.6 \\ 1.2 \\ 0 \end{bmatrix}$
3	$\begin{bmatrix} -1.3 & -0.8 & -0.1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.62 & -0.47 & -0.06 \\ -0.76 & -0.86 & -0.12 \\ 1.2 & 0.80 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.6 \\ 1.2 \\ 0 \end{bmatrix}$

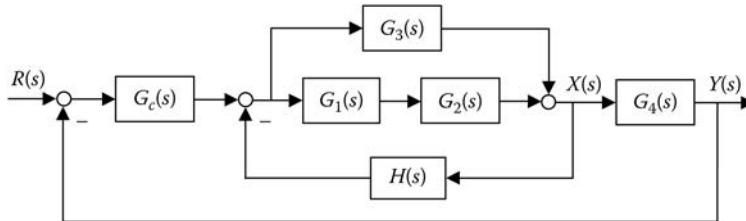


FIGURE 4.88 Block diagram of a continuous-time system.

$$G_1(s) = \frac{8}{3s+1}, \quad G_2(s) = \frac{s+5}{s^2+12s+25}, \quad G_3(s) = \frac{1}{0.2s+1}, \quad G_4(s) = \frac{1}{s} \quad (4.709)$$

Using block diagram algebra, the transfer function $Y(s)/R(s)$ can be found by executing the statements below found in M-file “Ch4_block_diagram.m.”

```

1. s = tf('s');
2. Gc = 5*(10*s+1)/(2*s+1);
3. G1 = 8/(3*s+1);
4. G2 = (s+5)/(s^2+12*s+25);
5. G3 = 1/(0.2*s+1);
6. G4 = 1/s;
7. H = 1/(50*s+1);
8. G1G2 = series(G1,G2);
9. G1G2_plus_G3 = parallel(G1G2,G3);
10. TF_inner_loop = feedback(G1G2_plus_G3,H)
11. G = series(Gc,TF_inner_loop);
12. G_forward_path_1 = series(G,G4);
13. TF_outer_loop_1 = feedback(G_forward_path_1,1)
  
```

The inner loop transfer function “TF_inner_loop” and outer loop transfer function TF_outer_loop_1 are

Transfer function:

$$150 s^4 + 1933 s^3 + 5189 s^2 + 3353 s + 65$$

$$30 s^5 + 520.6 s^4 + 2733 s^3 + 4693 s^2 + 1445 s + 90$$

Transfer function:

$$7500 s^5 + 97400 s^4 + 269095 s^3 + 193593 s^2 + 20015 s + 325$$

$$60 s^7 + 1071 s^6 + 1.349e004 s^5 + 1.095e005 s^4 + 276678 s^3 + 195218 s^2 + 20105 s + 325$$

Other transfer functions may be obtained by proper use of the three system interconnection commands. For example, $X(s)/R(s)$ in Figure 4.88 can be found by deleting statement 11 and changing statements 12 and 13 to read

- 14. `G_forward_path_2 = series(Gc,TF_inner_loop);`
- 15. `TF_outer_loop_2 = feedback(G_forward_path_2,G4)`

An alternate implementation of the transfer function $X(s)/R(s)$ is possible by expressing it in terms of $Y(s)/R(s)$. Starting with

$$Y(s) = G_4(s)X(s) \quad (4.710)$$

$$\Rightarrow \frac{Y(s)}{R(s)} = G_4(s) \frac{X(s)}{R(s)} \quad (4.711)$$

$$\Rightarrow \frac{X(s)}{R(s)} = \frac{1}{G_4(s)} \frac{Y(s)}{R(s)} \quad (4.712)$$

The transfer function $X(s)/R(s)$ can now be obtained by statement 14 below:

- 16. `TF_outer_loop_2 = series(1/G4,TF_outer_loop_1)`

The functions “parallel,” “series,” and “feedback” to reduce a system with forward and feedback connections apply to discrete-time system block diagrams as well.

4.10.5 SYSTEM RESPONSE

The impulse and step response of continuous- and discrete-time LTI systems can be generated in either graphical form or stored in an array of data points. To illustrate, suppose we are interested in the step response of the submarine depth control system considered earlier. Unit step responses of the stern plane angle θ , depth rate v , and depth c are obtained by appending “`step(sys_ss_1)`” or “`step(sys_tf)`” at the end of M-file “*Ch4_sub.m*.” The graphs are shown in Figure 4.89.

Step and impulse responses of the system in Figure 4.88 with $y(t)$ as output are obtained by issuing the control system toolbox commands “`step(TF_outer_loop_1)`” and “`impulse(TF_outer_loop_1)`” in M-file “*Ch4_block_diagram.m*.” The step and impulse responses are shown in Figure 4.90.

The response of an LTI system to an arbitrary input is obtained using “`LSIM(SYS,U,T)`” where “`SYS`” represents a MATLAB system object. “`U`” and “`T`” are arrays used to define the input (s) values and corresponding regularly spaced values of time, respectively.

The case study in Section 3.7 involved the ascent of a diver subject to a vertical cable force f_c . A state-space model was formulated and repeated as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\mu g & 0 \\ K \gamma & 0 & -K \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{g}{W} \\ 0 \end{bmatrix} [f_n] \quad (4.713)$$

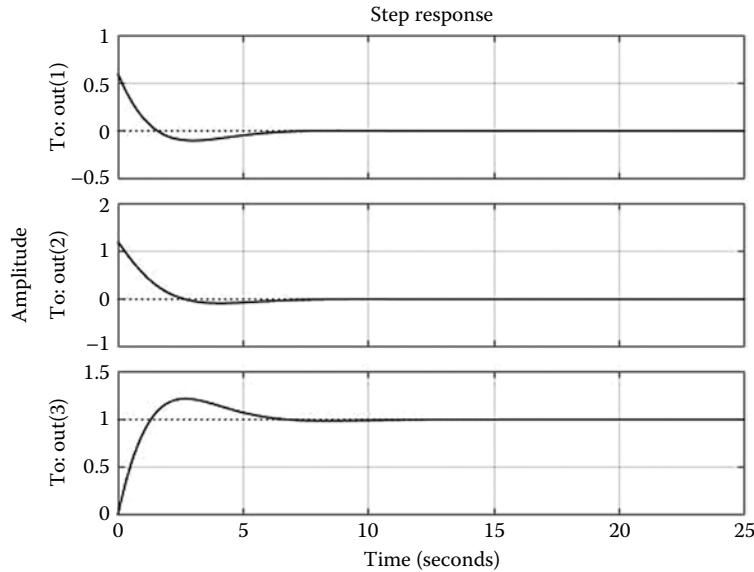


FIGURE 4.89 Unit-step response in θ , v , and c .

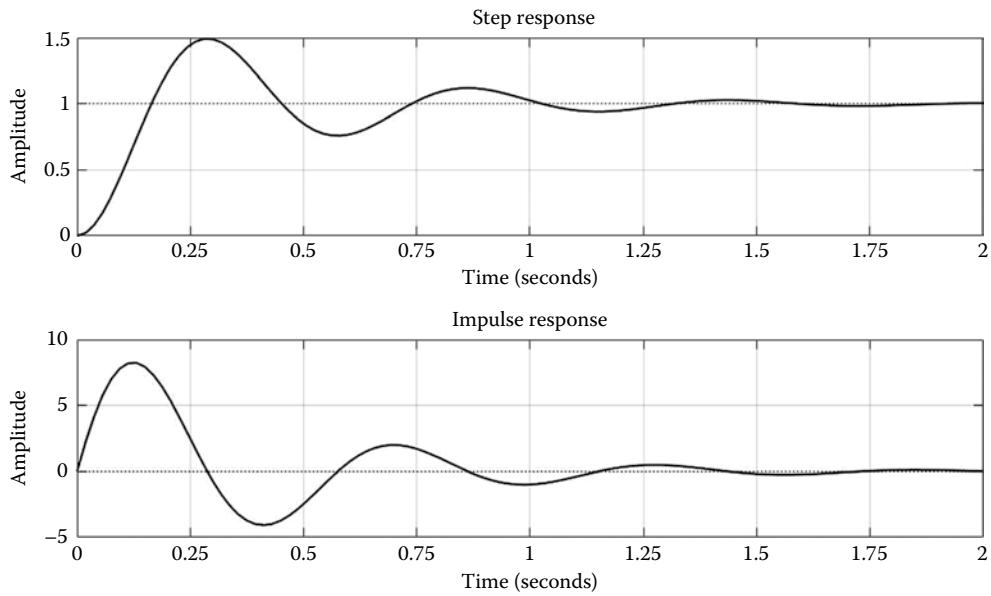


FIGURE 4.90 (a) Step and (b) impulse response of continuous-time system in Figure 4.88.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -\gamma & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.714)$$

The input $f_n = W - \gamma V - f_c$ is the net force (weight–buoyant force–cable force) acting on the diver. The output y_i is depth below the surface, and y_2 is the difference between the internal body pressure of the diver and the local (same depth as diver) underwater pressure. The states x_1 , x_2 , and

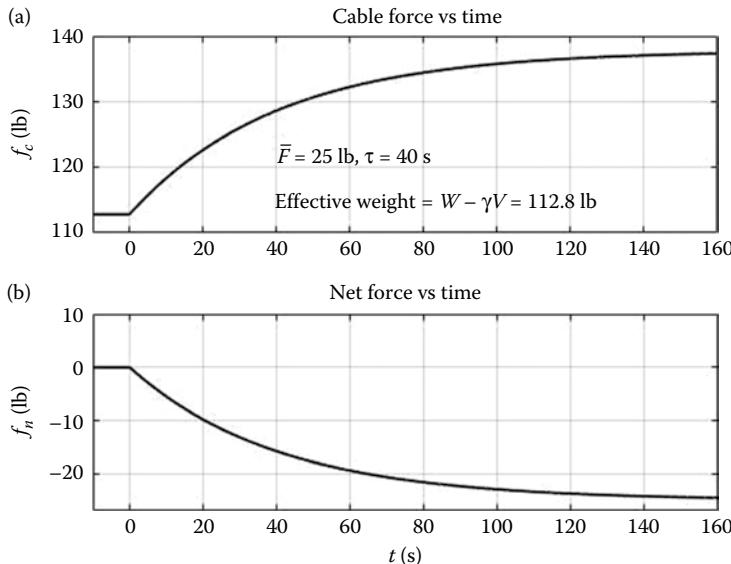


FIGURE 4.91 (a) Cable force and (b) net force on diver vs. time.

x_3 are depth, velocity, and internal pressure of the diver, respectively. The system parameters are μ , W , and K ; and g and γ are physical constants.

Suppose the diver's ascent from an initial equilibrium state $x_{1,e} = 500 \text{ ft}$, $x_{2,e} = 0 \text{ ft/s}$, and $x_{3,e} = \gamma x_{1,e} - 62.4 \text{ lb/ft}^3 \times 500 \text{ ft} = 31,200 \text{ lb/ft}^2$ (216.7 psi) is required. A cable force

$$f_c(t) = (W - \gamma V) + \bar{F}(1 - e^{-t/\tau}), \quad t \geq 0 \quad (4.715)$$

where \bar{F} and τ are design parameters is under investigation. The cable force $f_c(t)$ and the resulting net force $f_n(t)$ are plotted in Figure 4.91 for the case where $\bar{F} = 25 \text{ lb}$ and $\tau = 40 \text{ s}$ (see M-file "Ch4_diver.m".)

The M-file "Ch4_diver.m" includes a statement to create the state-space object "sys" from matrices A , B , C , and D in Equations 4.713 and 4.714. The time vector "t" is defined and input vector "fn" is calculated from the equation $f_n = (W - \gamma V) - f_c$. The statement "y = LSIM(sys, fn, t, x0)" where "x0" is the initial state vector, returns data points for outputs y_1 and y_2 in the array "y." Graphs of $y_1(t)$ and $y_2(t)$ are shown in Figure 4.92.

4.10.6 CONTINUOUS-/DISCRETE-TIME SYSTEM CONVERSION

We are well aware of the need to approximate the dynamics of continuous-time systems using discrete-time systems. Replacing the differential equations of LTI continuous-time system models with difference equations is an important aspect of continuous system simulation. Section 4.4.7 introduced a technique for accomplishing the task based on substitution of a suitable function of z for the Laplace variable s in the continuous-time system transfer function. Examples were presented illustrating how to obtain the z -domain transfer function of the discrete-time system based on the use of explicit Euler integration and trapezoidal integration, also known as Tustin's method.

Additional transformations $s = f(z)$ for other methods are discussed in a later chapter. For all but the simplest continuous-time systems, the algebraic manipulation required to obtain the z -domain or pulse transfer function in a suitable form is unwieldy at best. The MATLAB control system toolbox "c2d" function expedites the process of converting continuous-time models to discrete-time approximations. The required arguments are a MATLAB system object for the continuous-time

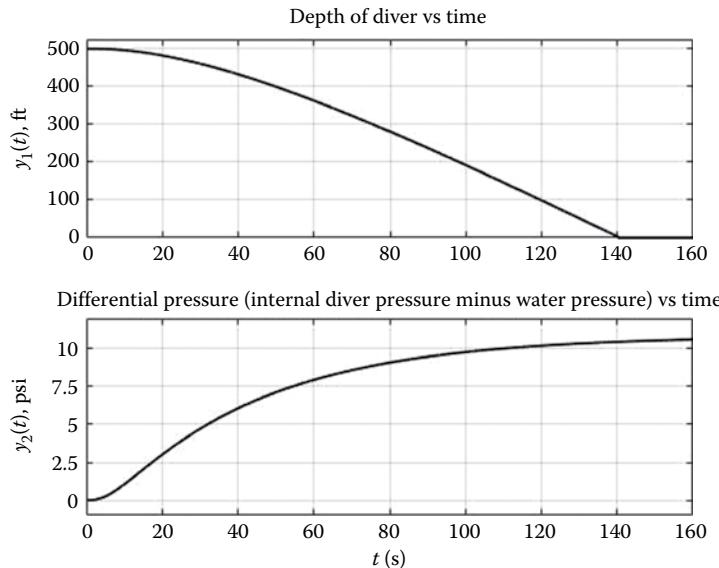


FIGURE 4.92 Outputs y_1 and y_2 from diver state-space model with input f_n .

system, the sample time (integration step size), and an optional string to select one of the five available approximation methods listed below:

```
'zoh' Zero-order hold on the inputs.
'foh' Linear interpolation of inputs (triangle apex.).
'imp' Impulse-invariant discretization.
'tustin' Bilinear (Tustin) approximation.
'prewarp' Tustin approximation with frequency prewarping. The critical
frequency  $W_c$  (rad/sec) is specified as 4th input by  $\text{SYSD} = \text{C2D}$  ( $\text{SYSC}$ ,  $T_s$ ,
'prewarp',  $W_c$ )
'matched' Matched pole-zero method (for SISO systems only).
```

To illustrate, consider the problem of approximating a second-order system with natural frequency $\omega_n = 2$ rad/s, $\zeta = 0.5$, and DC gain of unity. Example 4.27 presented solutions based on the use of explicit Euler integration and trapezoidal integration (Tustin's method), also known as the bilinear transform method. The following statements are from the M-file “*Ch4_Tustin.m*,” which creates the continuous-time transfer function “*H_s*” and generates the discrete-time system transfer function “*H_z*” using Tustin's method.

```
T = 0.025; wn = 2; zeta = 0.5; K = 1;
H_s = tf(K*wn^2, [12*zeta*wn*wn^2])
H_z = c2d(H_s, T, 'tustin')
```

The continuous- and discrete-time transfer functions appear in the MATLAB Command Window as

Transfer function:

4

$s^2 + 2 s + 4$

Transfer function

0.0006094 $s^2 + 0.001219 s + 0.0006094$

$s^2 - 1.949 s + 0.9512$

Sampling time: 0.025

The pulse transfer function approximation of the continuous-time second-order system using Tustin's method is (see Equation 4.503)

$$H(z) = \frac{K(\omega_n T)^2(z^2 + 2z + 1)}{[4(1 + \zeta\omega_n T) + (\omega_n T)^2]z^2 + 2[(\omega_n T)^2 - 4]z + 4(1 - \zeta\omega_n T) + (\omega_n T)^2} \quad (4.716)$$

Substituting the numerical values $\omega_n = 2$, $\zeta = 0.5$, $K = 1$, and $T = 0.025$ for the system parameters gives

$$H(z) = \frac{0.0025(z^2 + 2z + 1)}{4.1025z^2 - 7.9950z + 3.9025} \quad (4.717)$$

$$= \frac{0.00060938z^2 + 0.0012187z + 0.00060938}{z^2 - 1.9488z + 0.9512} \quad (4.718)$$

in agreement with the result from using the “c2d” function.

There is also a function called “d2c” for converting a discrete-time transfer function previously created as an object “sysd” to an equivalent continuous-time transfer function object “sysc.” The syntax is “SYSC = D2C(SYSD,METHOD)” where the second argument is a string signifying the method of approximation.

4.10.7 FREQUENCY RESPONSE

The magnitude and gain of a system transfer function at a particular frequency ω were evaluated in earlier sections by substituting $j\omega$ for s in continuous-time transfer functions and $e^{j\omega T}$ for z in discrete-time transfer functions. Choosing a range of values for ω led to plots of magnitude, $\text{gain} = 20(\log[\text{magnitude}])$ and phase vs. frequency.

The control system toolbox provides an easier way of obtaining the frequency response characteristics of both continuous- and discrete-time system models. Assuming an LTI model object called “sys” has been created using “tf” or possibly “ss,” a Bode plot is drawn by execution of the command “BODE(sys).” If “sys” represents a discrete-time system, the call is modified to include an additional argument for the sampling time T , namely, “BODE(sys,T).”

Optional arguments permit specifying multiple systems with different line plot characteristics and a user selectable range of frequencies. To illustrate, consider the two blocks in series shown in Figure 4.93.

The first component is a low-pass filter with transfer function

$$G_1(s) = \frac{X_1(s)}{U(s)} = \frac{1}{(\tau_1 s + 1)^n} \quad (4.719)$$

and break frequency $\omega_l = 1/\tau_1$. The second component transfer function

$$G_2(s) = \frac{X_2(s)}{X_1(s)} = \left(\frac{s}{\tau_2 s + 1} \right)^n \quad (4.720)$$

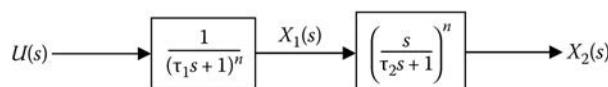


FIGURE 4.93 Low- and high-pass filters in series.

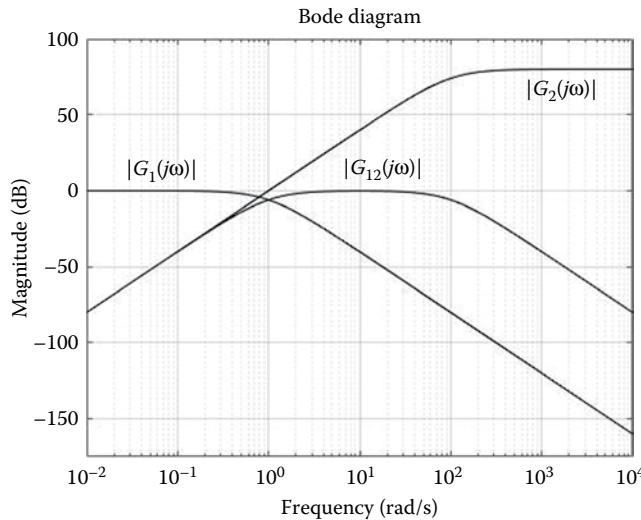


FIGURE 4.94 Gain of individual and combined blocks in Figure 4.93.

represents a high-pass filter with break frequency $\omega_2 = 1/\tau_2$. The frequency response characteristics of the series combination with transfer function

$$G_{12}(s) = \frac{1}{(\tau_1 s + 1)^n} \left(\frac{s}{\tau_2 s + 1} \right)^n \quad (4.721)$$

$$= \left[\frac{s}{(\tau_1 s + 1)(\tau_2 s + 1)} \right]^n \quad (4.722)$$

are obtained using the “BODE” function for a model object “sys” corresponding to Equation 4.722. The following M-file statements generate plots of the gain (magnitude in db) for the low-pass filter ($\tau_1 = 1$ s), high-pass filter ($\tau_2 = 0.01$ s), and the band-pass filter with pass band ($\omega_1 \leq \omega \leq \omega_2$) resulting from the combination of the two filters in series. The plots are shown in Figure 4.94. The exponent n was chosen to be three.

```
taul = 1; tau2 = 0.01; n = 3;
sys1 = tf(1, [taul 1])
sys2 = tf([1 0], [tau2 1])
for i = 1:n-1
    sysG1 = SERIES(sys1, sys1)
    sysG2 = SERIES(sys2, sys2)
end
sysG12 = SERIES(sysG1, sysG2)
BODEMAG(sysG1, 'b', sysG2, 'r', sysG12, 'k')
```

A discrete-time approximation of the continuous-time band-pass filter using Tustin’s method is obtained by adding the statements

```
T = pi/1e4; % sample time to make wN = 10^4 rad/sec
sysG12_d = C2D(sysG12, T, 'tustin'); % converts continuous-time filter % to
discrete-time filter using Tustin's method
```

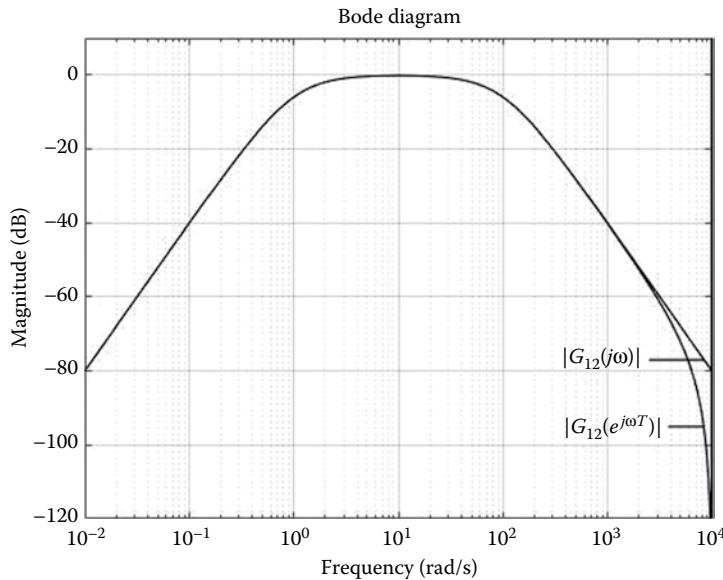


FIGURE 4.95 Gain of continuous- and discrete-time band-pass filters.

```
BODEMAG(sysG12_d, 'r') % plot gain of discrete-time filter
BODEMAG(sysG12, 'b') % plot gain of continuous-time filter
```

The sample time should be at least an order of magnitude less than $\tau_2 = 0.01$ s and possibly smaller depending on the frequency content of the continuous-time input. A value of $T = \pi/10^4$ s was chosen to make the Nyquist frequency $\omega_N = \pi/T = 10^4$ s, the same as the upper limit in Figure 4.94. Selecting appropriate values of T for discrete-time models is deferred until Chapter 8.

A comparison of the continuous-time and discrete-time band-pass filter gains for ($10^{-2} \leq \omega \leq \omega_N = 10^4$) is shown in Figure 4.95. The two gains are nearly identical for ω up to 10^3 rad/s. Frequency response includes phase characteristics as well as gain. The phase properties of the two filters are left for an exercise problem.

4.10.8 Root Locus

For simple feedback control systems with a controller gain K_C , the closed-loop system poles depend on the value of K_C . A root-locus plot displays the location of all the poles as the design parameter K_C varies from zero to infinity. The starting point is creation of the open-loop system model “sys” followed by a call to the control system toolbox function “rlocus(sys).” The following example illustrates the use of “BODE” and “rlocus” to determine the limits of stability for a simple control system.

EXAMPLE 4.36

An overdamped second-order system is subject to proportional control as shown in Figure 4.96.

A sensor is present in the feedback loop.

Baseline values of the system and sensor parameters are

$$K_p = 15, \tau_1 = 3\text{s}, \tau_2 = 15\text{s}, K_T = 0.1, \tau_T = 0.25\text{s}$$

- Create a model “sys” for the open-loop system with $K_C = 1$.
- Use the control system toolbox to draw a Bode plot of the open-loop system.

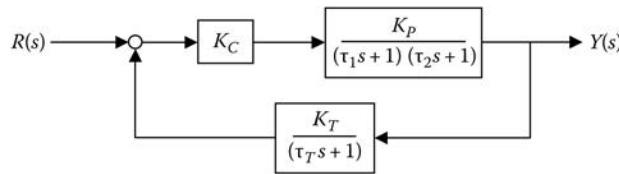


FIGURE 4.96 Feedback control system with proportional control.

- c. Determine the stability margins of the control system and the critical gain K_{cr} .
- d. Find ω_o , the frequency of oscillations for the marginally stable system.
- e. Check the results for K_{cr} using a root-locus plot and the characteristic equation.
- f. Plot step responses of the closed-loop system for $K_C = 0.25K_{cr}, 0.5K_{cr}, 0.75K_{cr}, K_{cr}$
- a. The model object “sys” is created in “Ch4_Ex4_36.m” with the statements

```
KP = 15; tau1 = 3; tau2 = 15; KT = 0.5; tauT = 0.25; KC = 1;
denG = conv([tau1 1], [tau2 1])
G = tf(KP, denG); % process transfer function
denH = [tauT 1];
H = tf(KT, denH); %sensor transfer function
sys = KC*SERIES(G,H)
```

- b. The command “BODE(sys)” results in the Bode plot in Figure 4.97.
- c. The stability margins were defined in Section 4.4.5. The gain margin is the open-loop system gain at the frequency where the phase of the open-loop system equals -180° . The phase margin is the difference between the open-loop phase and -180° at the frequency where the gain is 0 db. Figure 4.97 shows the gain margin is 20.5 db and the phase margin is 50.5° . Increasing the controller gain K_C by the equivalent of 20.5 db moves the gain plot in a vertical direction to a point where the system is marginally stable, that is, the new gain margin is 0 db. Solving for K_{cr} in magnitude,

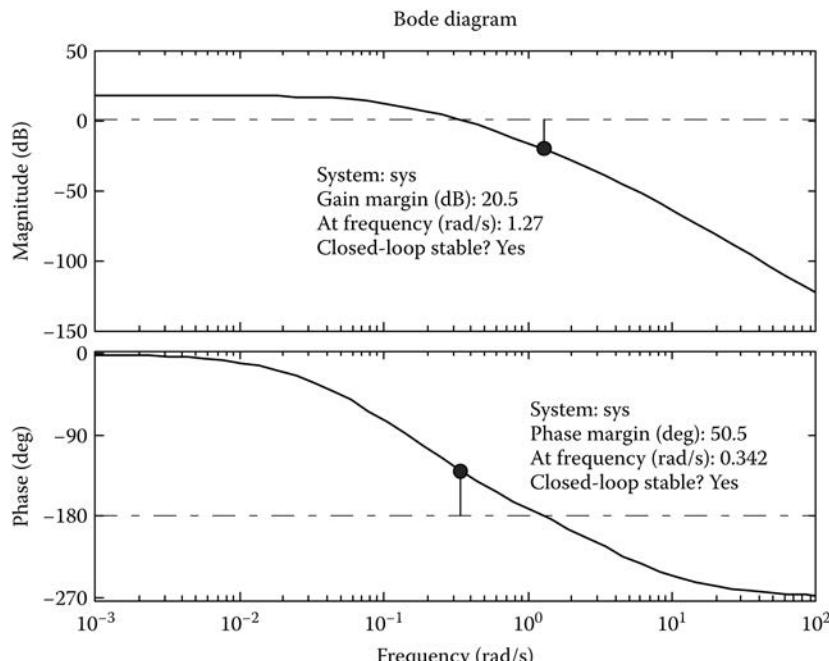


FIGURE 4.97 Bode plot for control system in Figure 4.96.

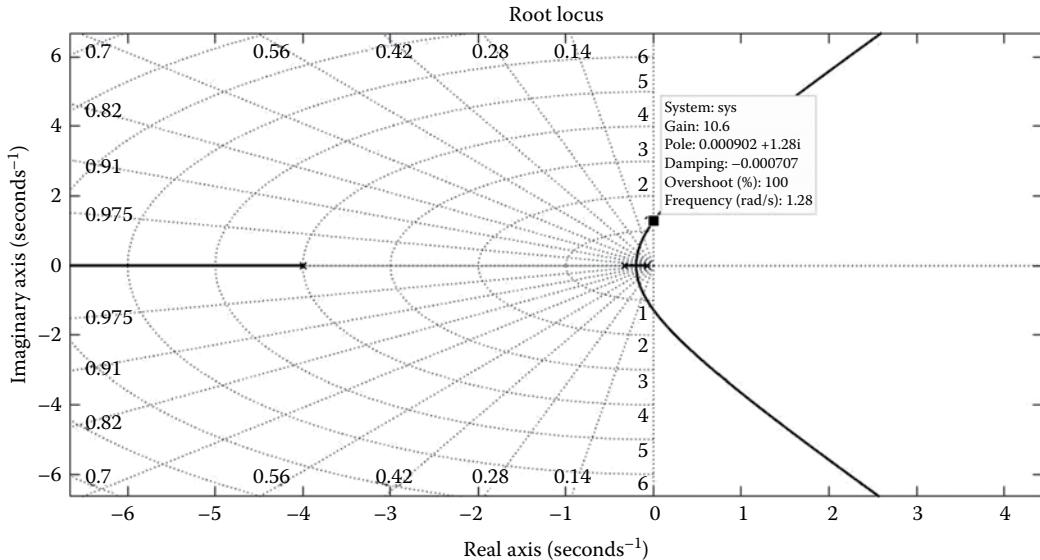


FIGURE 4.98 Root-locus plot for control system in Figure 4.96.

$$20 \log K_{cr} = 20.5 \Rightarrow K_{cr} = 10^{20.5/20} = 10.5925$$

- d. The 0 db gain margin would occur at the same frequency as the 20.5 db gain margin in Figure 4.97, that is, 1.27 rad/s, which is also ω_0 , the frequency of oscillations of the marginally stable system.
- e. The root-locus plot is shown in Figure 4.98. The approximate value of K_{cr} is 10.6, that is, the value of K_c where the locus intersects the imaginary axis. Note that the imaginary part of the complex pole is $\omega_0 = 1.27$ rad/s, in agreement with the crossover frequency shown in Figure 4.97.

As a check on the value of K_{cr} from part (c), the statement

```
[R, K] = rlocus(sys, Kcr)
```

returns the three closed-loop poles in array $R = [-4.4006, 0.003 \pm j1.2747]$. The real part of the complex poles should be zero when $K_c = K_{cr}$; however, 0.003 results because of the round-off in the gain margin value of 20.5 shown in Figure 4.97.

The exact values of K_{cr} and ω_0 can be obtained from the characteristic equation

$$K_c K_p K_T + (\tau_1 s + 1)(\tau_2 s + 1)(\tau_T + 1) = 0 \quad (4.723)$$

with $K_c = K_{cr}$ and $s = j\omega_0$. Setting the real and imaginary components of the resulting equation to zero leads to the following two equations:

$$\omega_0^2 = \frac{\tau_1 + \tau_2 + \tau_T}{\tau_1 \tau_2 \tau_T} \quad (4.724)$$

$$K_{cr} = \frac{[\tau_1 \tau_T + \tau_T (\tau_1 + \tau_2) \omega_0^2 - 1]}{K_p K_T} \quad (4.725)$$

The solution is (see “Ch4_Ex4_36.m”) $K_r = 10.5733$, $\omega_0 = 1.273665$ rad/s.

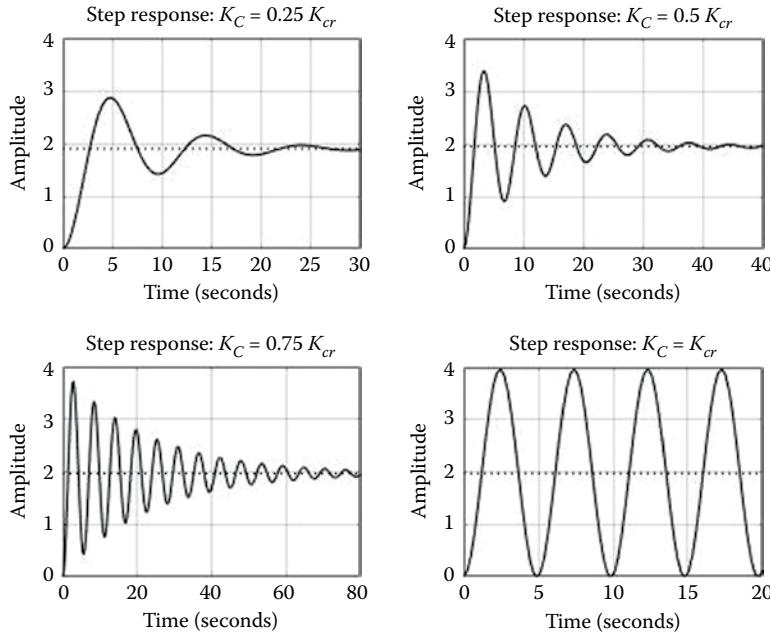


FIGURE 4.99 Step responses of control system in Figure 4.96.

f. Step responses of the closed-loop system with $K_C = 0.25K_{cr}$, $0.5K_{cr}$, $0.75K_{cr}$, K_{cr} are generated by the statements

```
for i = 1:4
subplot(2,2,i)
sys_cl = FEEDBACK(0.25*i*KCR*G,H); % closed-loop system
step(sys_cl) % step response
end
```

where “KCR” is the exact value for K_{cr} . The step responses, shown in Figure 4.99, exhibit less damping as the controller gain increases. The step response of the marginally stable system ($K_C = K_{cr}$) contains an oscillatory component at the frequency $\omega_0 = 1.27$ rad/s.

EXERCISES

Use the control system toolbox whenever possible to do the following problems:

- 4.78 The block diagram of a typical feedback control system was presented in Figure 4.31 and redrawn Figure E4.78:

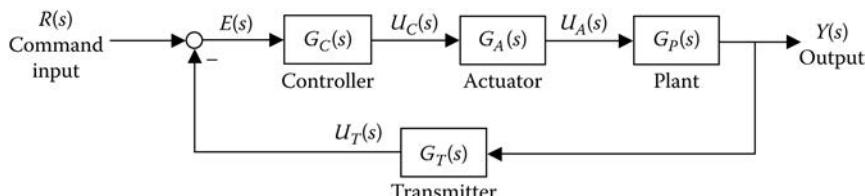


FIGURE E4.78

Use the transfer functions given in Section 4.4.5 and the baseline parameter values unless stated otherwise.

- Find the magnitude and phase of each component $G_C(s)$, $G_A(s)$, $G_P(s)$, and $G_T(s)$ at the open-loop system crossover frequency $\omega_0 = 0.9936$ rad/s. Compare the results to the magnitude and phase of the open-loop transfer function $G_{OL}(s) = G_C(s)G_A(s)G_P(s)G_T(s)$ at the same frequency.
- Input to the open-loop system (feedback path broken at summer) is $r(t) = \sin \omega_0 t$. Generate graphs of $e(t) = r(t)$, along with $u_C(t)$, $u_A(t)$, $y(t)$, and $u_T(t)$. Comment on the stability of the closed-loop system.
Hint: Recall the closed-loop system is unstable if the magnitude of $u_T(t)$ is greater than or equal to 1 at the phase crossover frequency ω_0 , that is, the frequency where $u_T(t)$ lags $e(t)$ by 180° .
- Graph the step response of the closed-loop system.
- Repeat parts (a), (b), and (c) using $K_C = (K_C)_{\max} = 2.62$.

4.79 The block diagram of a heading control system for a ship, presented in Section 4.4.4, is shown in [Figure E4.79](#). The baseline parameter values are

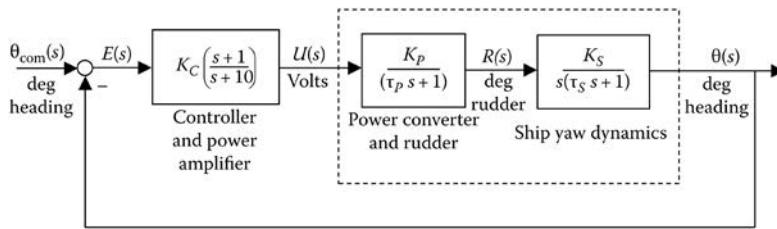


FIGURE E4.79

$$K_C = 10 \text{ V/}^\circ \text{ (heading)}$$

$$K_p = 10^\circ \text{ (rudder)/volt}, \tau_p = 0.2 \text{ s}$$

$$K_S = 0.5^\circ \text{ (heading)}/\text{s}^\circ \text{ (rudder)}, \tau_S = 7.5 \text{ s}$$

- Find the closed-loop transfer functions

$$\frac{E(s)}{\theta_{\text{com}}(s)}, \frac{U(s)}{\theta_{\text{com}}(s)}, \frac{R(s)}{\theta_{\text{com}}(s)}, \text{ and } \frac{\theta(s)}{\theta_{\text{com}}(s)}$$

- For a step input $\theta_{\text{com}} = 5^\circ$, $t \geq 0$ graph $e(t)$, $u(t)$, $r(t)$, and $\theta(t)$.

4.80 A system of two interacting tanks is shown in [Figure E4.80a](#):

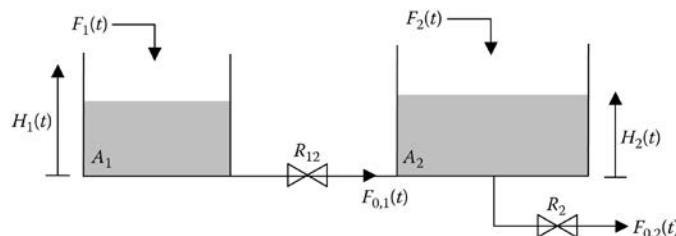


FIGURE E4.80A

The state equations are given as

$$\begin{bmatrix} \frac{dH_1/dt}{dH_2/dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_{12}} & \frac{1}{A_1 R_{12}} \\ \frac{1}{A_2 R_{12}} & -\frac{1}{A_2 R_{12}} - \frac{1}{A_2 R_{12}} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \begin{bmatrix} \frac{1}{A_1} & 0 \\ 0 & \frac{1}{A_2} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ A_1 & A_2 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}$$

The parameter values are

$$A_1 = 25 \text{ ft}^2, A_2 = 100 \text{ ft}^2, R_{12} = 0.1 \text{ ft}/\text{ft}^3/\text{min}, R_2 = 0.4 \text{ ft}/\text{ft}^3/\text{min}$$

- Find the transfer functions $V_T(s)/F_1(s)$ and $V_T(s)/F_2(s)$.
- With both tanks initially empty, find and graph $H_1(t)$ and $H_2(t)$ in response to
 - $F_1(t) = 12 \text{ ft}^3/\text{min}, F_2(t) = 0 \text{ ft}^3/\text{min}$
 - $F_1(t) = 0 \text{ ft}^3/\text{min}, F_2(t) = 12 \text{ ft}^3/\text{min}$
 - $F_1(t) = 12 \text{ ft}^3/\text{min}, F_2(t) = 12 \text{ ft}^3/\text{min}$
 - $F_1(t)$ in Figure E4.80b

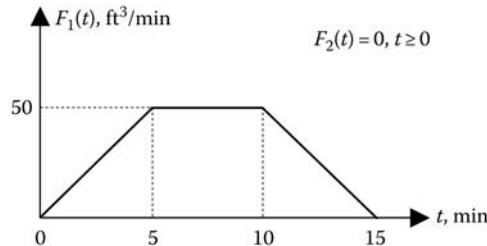


FIGURE E4.80B

4.81 The transfer function for the circuit in Figure E4.81 is (see Equation 4.183)

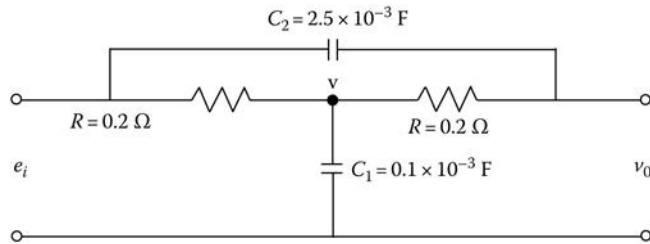
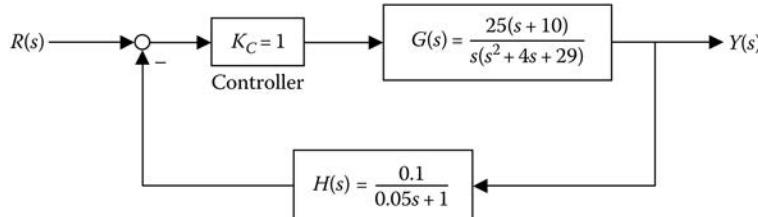


FIGURE E4.81

$$\frac{V_0(s)}{E_i(s)} = \frac{R^2 C_1 C_2 s^2 + 2 R C_2 s + 1}{R^2 C_1 C_2 s^2 + R(C_1 + 2C_2)s + 1}$$

- Convert the system transfer function to a state variable model with output v_0 .
- Use the state variable model to find and plot the impulse response.
- Find the unit step response of the circuit by inverse Laplace transforming $V_0(s)$.
- Repeat part (c) using the control system toolbox to find the unit step response. Compare the results from parts (c) and (d).
- Approximate the continuous-time transfer function with a discrete-time z-domain transfer function based on Tustin's method. Choose an appropriate integration step size.
- Find and plot the unit step response of the discrete-time system. Compare the step responses of the continuous-time and discrete-time systems.

- 4.82 Use “BODE” instead of “BODEMAG” to plot the magnitude and phase plots for the filters with transfer functions in Equations 4.719 through 4.721.
- 4.83 Compare the phase characteristics of the continuous- and discrete-time band-pass filters introduced in this section.
- 4.84 A simple control system block diagram is shown in Figure E4.84:

**FIGURE E4.84**

- Find the closed-loop transfer function of the system using block diagram reduction.
 - Check your answer to part (a) using the control system toolbox.
 - Draw a simulation diagram of the system.
 - Represent the system in state variable form based on your simulation diagram.
 - Use the control system toolbox to find a state variable model for the system.
 - Compare the eigenvalues (characteristic poles) of the coefficient matrix A in parts (d) and (e).
 - Use “bode” to plot the frequency response of the open-loop system transfer function. Find the gain and phase margins of the system.
 - Compute the maximum gain (K_C) which makes the system marginally stable. Redraw the Bode plot for $K_C = (K_C)_{crit}$
 - Check your answer to part (h) using a root-locus plot and identifying the value of gain K_C where the locus is on the Imaginary axis.
- 4.85 A continuous-time system is modeled by the differential equation

$$\frac{d^3y}{dt^3} + 5\frac{d^2y}{dt^2} + 33\frac{dy}{dt} + 29y = u$$

- Find the transfer function $H(s) = Y(s)/U(s)$ of the system.
- Create a model object “sys” to represent $H(s)$.
- Use the control system toolbox to plot the impulse and step response of the system.
- Approximate the continuous-time transfer function $H(s)$ with a discrete-time z-domain transfer function $H(z) = Y(z)/U(z)$ using Tustin’s method with appropriate sample time T .
- Find the difference equation for the discrete-time system approximation.
- Write a MATLAB M-file to find and plot the step response of the discrete-time system.
- Use the control system toolbox to plot the step response of the discrete-time system, and compare the result with your answer in part (f).

4.11 CASE STUDY: LONGITUDINAL CONTROL OF AN AIRCRAFT

The equations of motion for an aircraft are derived using a moving coordinate system fixed to the aircraft as shown in Figure 4.100. The $x-y-z$ axes are referred to as body axes. The x -axis is aligned with the longitudinal axis of the airplane. The equations are based on Newton’s laws of motion for a rigid body in translation and rotation. The result is a system of six coupled nonlinear differential equations. Three of the six equations express accelerations $\dot{u}, \dot{v}, \dot{w}$ in terms of body axis velocities u, v, w , angular velocities p, q, r , and external, aerodynamic, and gravitational forces acting on the plane. The remaining three equations relate the angular accelerations $\dot{p}, \dot{q}, \dot{r}$ to p, q, r and moments produced by the external and aerodynamic forces about the plane’s center of mass.

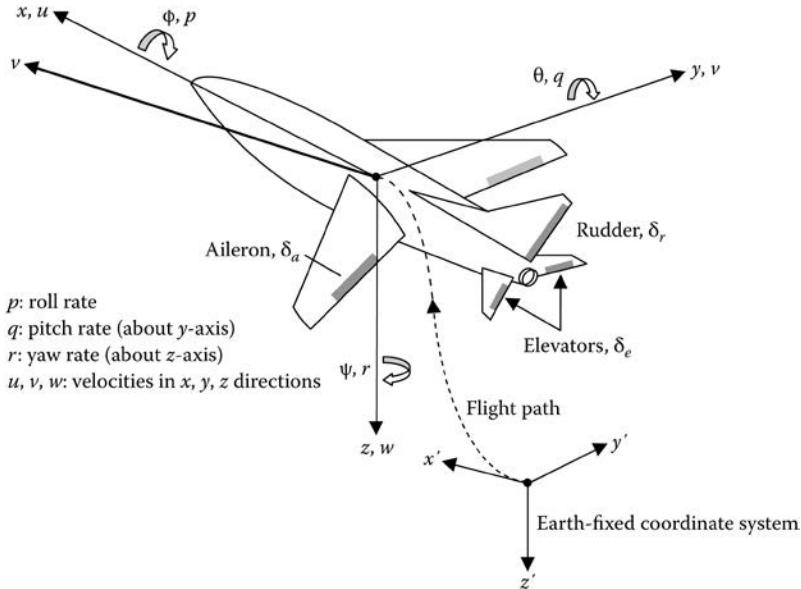


FIGURE 4.100 Body axis coordinates (x, y, z) and Euler angles (ψ, θ, ϕ).

The position and orientation of the airplane are referenced to an inertial (earth-fixed) coordinate system $x'-y'-z'$ also shown in Figure 4.100. The horizontal x' -axis is in the vertical plane containing the initial velocity vector, and the plane's center of mass is located at the origin of the $x'-y'-z'$ system at $t = 0$. The plane's attitude is fixed by three rotations of the $x-y-z$ axes starting from an orientation initially aligned with the $x'-y'-z'$ axes of the inertial coordinate system. The angular rotations ψ , θ , and ϕ are called Euler angles and denote the roll, pitch, and yaw of the plane, respectively.

Solution to the flight dynamics equations yields u, v, w in the $x-y-z$ body axis coordinate system. The velocity vector v is converted from body axis components u, v, w to inertial components $\dot{x}', \dot{y}', \dot{z}'$ by a transformation matrix C_e^b (Etkin 1982),

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{bmatrix} = C_e^b \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (4.726)$$

$$C_e^b = \begin{bmatrix} \cos\theta \cos\psi & \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi & \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ \cos\theta \sin\psi & \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi & \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ -\sin\theta & \sin\phi \cos\theta & \cos\phi \cos\theta \end{bmatrix} \quad (4.727)$$

The position of the plane's center of mass in inertial coordinates x', y', z' is obtained by integration of the respective velocities in Equation 4.726.

Solving the equations of motion also yields the angular velocities p, q, r , which are transformed into, $\dot{\psi}, \dot{\theta}, \dot{\phi}$ by

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \\ 0 & \cos\phi & -\sin\phi \\ 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (4.728)$$

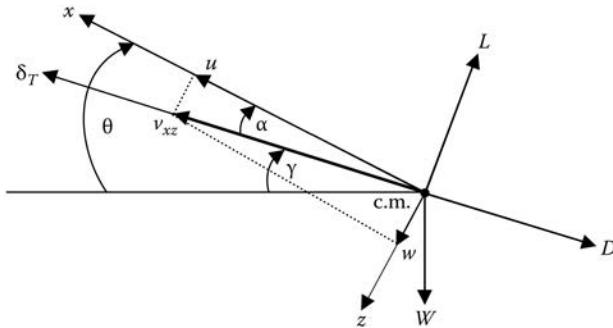


FIGURE 4.101 Illustration of angle of attack (α) and forces influencing flight dynamics.

The Euler angles ψ , θ , and ϕ are obtained by integration of the respective velocities in Equation 4.728.

Solution of the nonlinear flight dynamics equations is complicated by the dependency of the aerodynamic forces and moments on the variable flight conditions, for example, altitude, cruising speed, weight, angle of attack, side slip, and control surface positions. A simpler approach is based on a linearized model that describes the aircraft's motion provided the excursions in flight from a known steady state are small. The subject of linearization is treated in some detail in [Chapter 7](#).

When the conditions for linearization of the flight equations are satisfied, the linearized model can be decoupled into two sets of equations. One set describes the longitudinal dynamics of the aircraft, and the remaining equations apply to the lateral dynamics. The longitudinal dynamics involve changes in u and w , the plane's velocity in the x - and z -directions, and the pitch rate q about the y -axis. Lateral dynamics involve changes in side velocity v and the yaw and roll rates r and p about the z - and x -axes, respectively.

[Figure 4.100](#) shows the velocity vector v aligned differently from the x -axis. The projection of v in the $x-z$ plane is v , shown in [Figure 4.101](#). The angle between v_{xz} and the x -axis (longitudinal axis of plane) is called the angle of attack. Note that when the lateral dynamics of the plane are zero, the flight path is confined to the $x-z$ plane, $v = v_{xz}$, and the instantaneous direction of flight is given by γ in [Figure 4.101](#), the angle between the velocity vector and the horizontal direction. The thrust (δ_T) from the engine, the aerodynamic forces, lift (L) and drag (D), and the gravitational force (W) are also shown in [Figure 4.101](#).

The primary control surfaces for controlling the aircraft's position and attitude are the elevators, ailerons, and rudder. The longitudinal dynamics respond to changes in elevator deflection δ_e and thrust δ_T . Elevator deflection and thrust result from changes to the yoke and throttle by the pilot (or autopilot). The rudder and ailerons are used primarily to control the lateral response for banking and turning maneuvers.

Our interest is solely in the longitudinal dynamics, specifically pitch and altitude response of the aircraft to changes in elevator deflection when the plane is flying at a constant cruising speed in horizontal flight under steady-state conditions. From [Figure 4.101](#), for the plane to be in level flight, the velocity vector v must be horizontal, the flight angle $\gamma = 0$, and the pitch is equal to the angle of attack. The plane is pitched slightly in order for the wings to develop sufficient lift to overcome gravity. The steady-state conditions are shown in [Figure 4.102](#) with v_0 (horizontal cruising speed), \bar{u} (longitudinal speed), \bar{w} (speed in z -direction), $\bar{\theta}$ (pitch), and $\bar{\alpha}$ (angle of attack). The elevator input and engine thrust necessary to maintain these conditions are δ_e and δ_T , respectively.

The deviations in u , α , θ , w , and q from their steady-state operating levels are

$$\Delta u = u - \bar{u}, \Delta w = w - \bar{w} = w, \Delta \alpha = \alpha - \bar{\alpha}, \Delta \theta = \theta - \bar{\theta}, \Delta q = q - \bar{q} = q \quad (4.729)$$

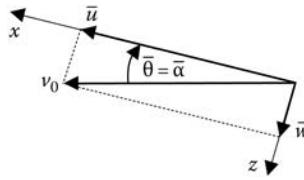


FIGURE 4.102 Initial steady-state conditions of aircraft.

Since we are considering only changes in elevator deflection,

$$\Delta\delta_e = \delta_e - \bar{\delta}_e, \quad \Delta\delta_T = \delta_T - \bar{\delta}_T = 0 \quad (4.730)$$

The state vector $\underline{\Delta x}$ in a linearized model of the longitudinal dynamics can be chosen as either $[\Delta u \Delta w \Delta q \Delta\theta]^T$ or $[\Delta u \Delta\alpha \Delta q \Delta\theta]^T$. The relationship between u , w , and α is (see Figure 4.101)

$$\tan \alpha = \frac{w}{u} \quad (4.731)$$

For small angles of attack, $\tan \alpha = \sin \alpha / \cos \alpha \approx \alpha$. Replacing $\tan \alpha$ in Equation 4.731 with α and solving for w give

$$w = u\alpha \quad (4.732)$$

Solving for u , α , and w in Equation 4.729 and substituting the results into Equation 4.732,

$$\bar{w} + \Delta w = (\bar{u} + \Delta u)(\bar{\alpha} + \Delta\alpha) = \bar{u}\bar{\alpha} + \bar{u}\Delta\alpha + \bar{\alpha}\Delta u + \Delta u\Delta w \quad (4.733)$$

Recognizing that $\bar{w} = \bar{u}\bar{\alpha}$ and ignoring the high-order term $\Delta u\Delta w$ lead to

$$\Delta w = \bar{u}\Delta\alpha + \bar{\alpha}\Delta u \quad (4.734)$$

Suppose the linearized model of an aircraft cruising in level flight under steady-state conditions with $v_0 = 500$ ft/s and $\bar{\alpha} = \bar{\theta} = 0.05$ rad (2.86°) is

$$\frac{d}{dt} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} -0.04 & 11.59 & 0 & -32.2 \\ -0.00073 & -0.65 & 1 & 0 \\ 0.000048 & -0.49 & -0.58 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} 0 & 0.1 \\ 0 & 0 \\ -0.014 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_T \end{bmatrix} \quad (4.735)$$

where

Δu has units of ft/s

$\Delta\alpha$, $\Delta\theta$ are in rad

Δq is in rad/s

$\Delta\delta_e$ is in degree of elevator deflection

$\Delta\delta_T$ is in lb of thrust

Choosing the output $\underline{\Delta y} = \underline{\Delta x} = [\Delta u \Delta\alpha \Delta q \Delta\theta]^T$ leads to the system of state equations $\dot{\underline{x}} = A\underline{\Delta x} + B\underline{\Delta u}$, $\underline{y} = C\underline{\Delta x} + D\underline{\Delta u}$ with A and B the matrices in Equation 4.735, C equal to the 4×4 identity matrix and D is a 4×2 matrix of zeros. Note that $\underline{\Delta u} = [\Delta\delta_e \Delta\delta_T]^T$ is the input vector, not to be confused with Δu , the first component of the state vector.

The linearized equations in state variable form can be converted to a transfer function matrix relating the four outputs $\Delta u(s)$, $\Delta\alpha(s)$, $q(s)$, and $\Delta\theta(s)$ to the two inputs $\Delta\delta_e(s)$ and $\Delta\delta_T(s)$. The transfer function matrix can be found using Equation 4.231, repeated again for convenience in Equation 4.736.

$$G(s) = \begin{bmatrix} \frac{\Delta u(s)}{\Delta\delta_e(s)} & \frac{\Delta u(s)}{\Delta\delta_T(s)} \\ \frac{\Delta\alpha(s)}{\Delta\delta_e(s)} & \frac{\Delta\alpha(s)}{\Delta\delta_T(s)} \\ \frac{q(s)}{\Delta\delta_e(s)} & \frac{q(s)}{\Delta\delta_T(s)} \\ \frac{\Delta\theta(s)}{\Delta\delta_e(s)} & \frac{\Delta\theta(s)}{\Delta\delta_T(s)} \\ \frac{\Delta\delta_e(s)}{\Delta\delta_e(s)} & \frac{\Delta\delta_T(s)}{\Delta\delta_T(s)} \end{bmatrix} = C(sI - A)^{-1}B + D \quad (4.736)$$

The control system toolbox in MATLAB contains a function “`ss2tf`” for expediting the process of converting from the state-space model to the transfer function description of an LTI system. Calling this function with arguments (A, B, C, D, i) , where $i = 1$ designates the first input $\Delta\delta_e$ and $i = 2$ specifies the second input $\Delta\delta_T$, generates the eight transfer functions in Equation 4.736.

The MATLAB statement “[numG denG] = ss2tf (A, B, C, D, 1)” returns

```
numG = 0      0.0000   -0.0000    0.2906    0.2951
        0      0.0000   -0.0141   -0.0006   -0.0003
        0     -0.0141   -0.0097   -0.0005    0.0000
        0      0.0000   -0.0141   -0.0097    0.0125

denG =  1.0000   1.2700   0.9247   0.0406   0.0125
```

The transfer function relating elevator input to aircraft pitch is therefore

$$G_{\Delta\delta_e}^{\Delta\theta}(s) = \frac{\Delta\theta(s)}{\Delta\delta_e(s)} = \frac{-0.0141s^2 - 0.0097s - 0.0005}{s^4 + 1.2700s^3 + 0.9247s^2 + 0.0406s + 0.0125} \quad (4.737)$$

Factoring the numerator and denominator gives

$$G_{\Delta\delta_e}^{\Delta\theta}(s) = \frac{\Delta\theta(s)}{\Delta\delta_e(s)} = \frac{K_\theta(s + c_1)(s + c_2)}{(s^2 + a_1s + b_1)(s^2 + a_2s + b_2)} \quad (4.738)$$

The constants in Equation 4.738, computed in M-file “*Ch4_CaseStudy1.m*,” are

$$K_\theta = -0.0141, c_1 = 0.6358, c_2 = 0.0542, a_1 = 1.2440, a_2 = 0.0260,$$

$$b_1 = 0.8780, b_2 = 0.0143$$

The quadratic factors in the denominator of Equation 4.738 are both underdamped, regardless of whether the aircraft is a small passenger plane, a commercial jet, or a high-performance military aircraft. However, as we shall soon learn, the natural frequencies and damping ratios of each quadratic are quite different.

We begin by finding the pitch response to a step change in elevator input of “A” deg. The Laplace transform of the response is

$$\Delta\theta(s) = \frac{K_\theta(s + c_1)(s + c_2)}{(s^2 + a_1s + b_1)(s^2 + a_2s + b_2)} \cdot \frac{A}{s} \quad (4.739)$$

Using partial fraction expansion, Equation 4.739 is written as

$$\Delta\theta(s) = K_\theta A \left[\frac{R_1}{s - p_1} + \frac{R_2}{s - p_2} + \frac{R_3}{s - p_3} + \frac{R_4}{s - p_4} + \frac{R_5}{s} \right] \quad (4.740)$$

where p_1 and p_2 are the poles from the quadratic $s^2 + a_1s + b_1$, and p_3 and p_4 are the poles associated with the quadratic $s^2 + a_2s + b_2$. R_1, R_2, R_3, R_4 , and R_5 are the constants (residues) in the partial fraction expansion. Letting $p_1 = \alpha_1 + j\beta_1$, $p_3 = \alpha_3 + j\beta_3$ and recognizing that $p_2 = \bar{p}_1 = \alpha_1 - j\beta_1$, $p_4 = \bar{p}_3 = \alpha_3 - j\beta_3$ as well as $R_2 = \bar{R}_1$, $R_4 = \bar{R}_3$ give

$$\Delta\theta(t) = \mathcal{L}^{-1}\{\theta(s)\} = K_\theta A [R_1 e^{pt} + \bar{R}_1 e^{\bar{p}t} + R_3 e^{p_3 t} + \bar{R}_3 e^{\bar{p}_3 t} + R_5], \quad t \geq 0 \quad (4.741)$$

It is left as an exercise to show that

$$Re^{pt} + \bar{R}e^{\bar{p}t} = 2e^{\alpha t} [\operatorname{Re}(R)\cos\beta t - \operatorname{Im}(R)\sin\beta t] \quad (4.742)$$

where

$$p = \alpha + j\beta, \bar{p} = \alpha - j\beta, R = \operatorname{Re}(R) + j\operatorname{Im}(R), \bar{R} = \operatorname{Re}(R) - j\operatorname{Im}(R)$$

The pitch response (in rad) to an $A = 1^\circ$ elevator deflection is given by

$$\begin{aligned} \Delta\theta(t) = K_\theta & \{ 2e^{\alpha_1 t} [\operatorname{Re}(R_1)\cos\beta_1 t - \operatorname{Im}(R_1)\sin\beta_1 t] \\ & + 2e^{\alpha_3 t} [\operatorname{Re}(R_3)\cos\beta_3 t - \operatorname{Im}(R_3)\sin\beta_3 t] + R_5 \} \end{aligned} \quad (4.743)$$

Assuming the aircraft's natural dynamics are stable, the poles are located in the left-half plane, that is, $\alpha_1 < 0$ and $\alpha_3 < 0$. From Equation 4.739 and the final value theorem and Equation 4.743 with $t \rightarrow \infty$, the steady-state pitch response to a unit step input is

$$\Delta\theta_{ss} = \frac{K_\theta c_1 c_2}{b_1 b_2} = K_\theta R_5 \quad (4.744)$$

The poles and residues are obtained in “Ch4_CaseStudy1.m.”

$$\begin{aligned} p_{1,2} &= -0.6220 \pm j0.7008, \quad p_{3,4} = -0.0130 \pm j0.1187 \\ R_{1,2} &= -0.0331 \pm j0.5589, \quad R_{3,4} = -1.3429 \pm j2.9777, \quad R_5 = 2.7519 \end{aligned}$$

From Equation 4.743, the pitch step response is

$$\begin{aligned} \Delta\theta(t) = -0.0141 & \{ 2e^{-0.6220t} [-0.0331 \cos 0.7008t - 0.5586 \sin 0.7008t] \\ & + 2e^{-0.0130t} [-1.3429 \cos 0.1187t - 2.9777 \sin 0.1187t] + 2.7519 \} \end{aligned} \quad (4.745)$$

The two damped oscillatory components are referred to as the short period and phugoid modes. The natural frequencies, damping ratios, and exponential envelope time constants are given in [Table 4.12](#).

TABLE 4.12
Short Period and Phugoid Mode Parameters

Mode	ω_n (rad/s)	Z	$\tau_{\text{envelope}} = 1/\zeta\omega_n$ (s)
Short period	0.9370	0.6638	1.6077
Phugoid	0.1194	0.1089	76.9042

The complete step response is shown in Figure 4.103. The steady-state pitch is from Equation 4.744, $\theta_{ss} = -0.0388$ rad (-2.2232°).

The short period and phugoid mode oscillation components of the step response are shown in Figure 4.104.

Shortly, we will look at the design of an autopilot to control the plane's altitude. Before doing so, a way of determining altitude is needed. From Equations 4.726 and 4.727,

$$\dot{z}' = (-\sin \theta)u + (\sin \varphi \cos \theta)v + (\cos \varphi \cos \theta)w \quad (4.746)$$

where \dot{z}' is the rate of change of altitude, a positive value indicating that the plane is descending. For small values of θ and motion in the longitudinal direction only, $v = 0$, $\phi = 0$, $\sin \theta \approx \theta$, $\cos \theta \approx 1$, $\sin \phi = 0$, $\cos \phi = 1$ and Equation 4.746 simplifies to

$$\dot{z}' = -\theta u + w \quad (4.747)$$

In terms of steady-state values and deviations, Equation 4.747 becomes

$$\frac{d}{dt}(\bar{z} + \Delta z') = -(\bar{\theta} + \Delta\theta)(\bar{u} + \Delta u) + \bar{w} + \Delta w \quad (4.748)$$

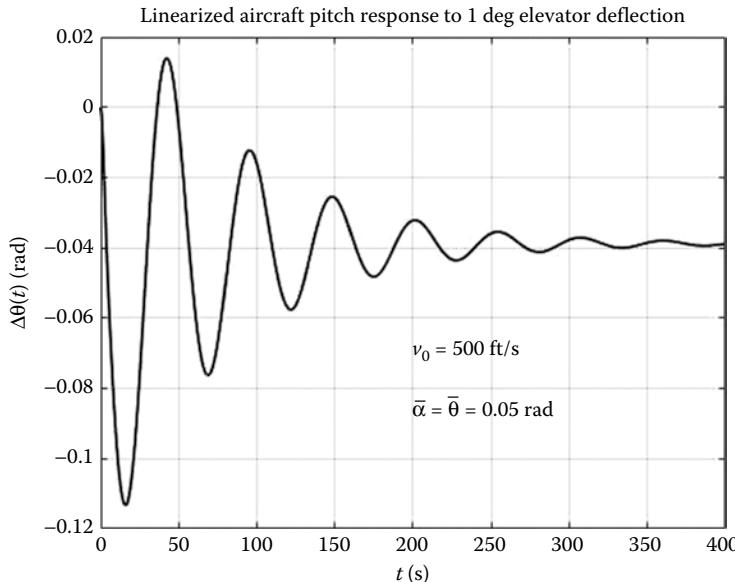


FIGURE 4.103 Linearized aircraft pitch response due to 1° step change in elevator deflection.

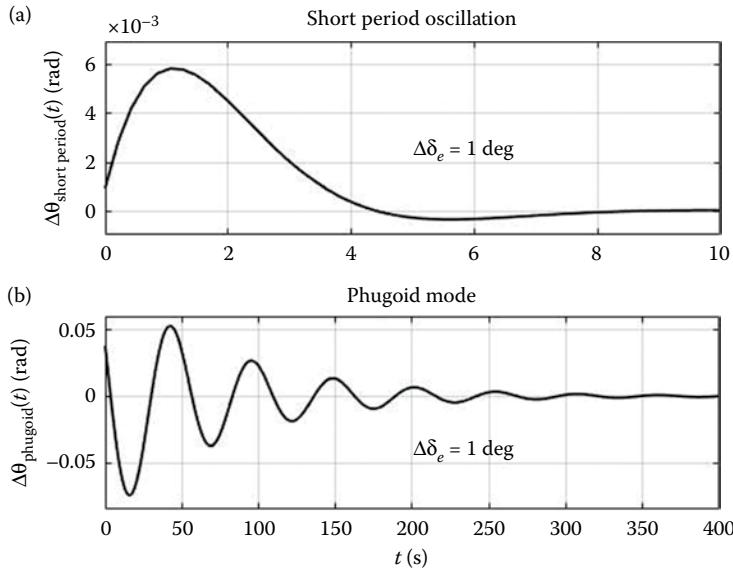


FIGURE 4.104 (a) Short period and (b) phugoid oscillations of elevator unit step response.

$$\Rightarrow \frac{d}{dt}(\bar{z}') + \frac{d}{dt}(\Delta z') = -(\bar{\theta}\bar{u} + \bar{u}\Delta\theta + \bar{\theta}\Delta u + \Delta\theta\Delta u) + \bar{w} + \Delta w \quad (4.749)$$

$$\Rightarrow \frac{d}{dt}(\bar{z}') + \frac{d}{dt}(\Delta\bar{z}') = -(-\bar{\theta}\bar{u} + \bar{w}) - \bar{u}\Delta\theta - \bar{\theta}\Delta u + \Delta\theta\Delta u + \Delta w \quad (4.750)$$

Equation 4.747 evaluated at steady state is

$$\frac{d}{dt}(\bar{z}') = -(\bar{\theta}\bar{u} + \bar{w}) \quad (4.751)$$

Subtracting Equation 4.751 from Equation 4.750, ignoring the higher order term $\Delta\theta \Delta u$, and recognizing that $d\Delta z'/dt = d(z' - \bar{z}')/dt = dz'/dt$ yield

$$\frac{dz'}{dt} = -\bar{u}\Delta\theta - \bar{\theta}\Delta u + \Delta w \quad (4.752)$$

Substituting Δw in Equation 4.734 into Equation 4.752 gives

$$\frac{dz'}{dt} = -\bar{u}\Delta\theta - \bar{\theta}\Delta u + (\bar{u}\Delta\alpha + \bar{\alpha}\Delta u) \quad (4.753)$$

$$= -(\bar{\theta} - \bar{\alpha})(\Delta u - \bar{u})(\Delta\theta - \Delta\alpha) \quad (4.754)$$

$$= -\bar{u}(\Delta\theta - \Delta\alpha) \quad (4.755)$$

Laplace transforming Equation 4.755,

$$\dot{z}'(s) = -\bar{u}[\Delta\theta(s) - \Delta\alpha(s)] \quad (4.756)$$

The transfer function from elevator input $\Delta\delta_e(t)$ to output $\dot{z}'(t)$ is

$$G_{\dot{z}'}(s) = \frac{\dot{z}'(s)}{\Delta\delta_e(s)} = -\bar{u} \left[\frac{\Delta\theta(s)}{\Delta\delta_e(s)} - \frac{\Delta\alpha(s)}{\Delta\delta_e(s)} \right] \quad (4.757)$$

The transfer function $\Delta\alpha(s)/\Delta\delta_e(s)$ is obtained in the same way we found $\Delta\theta(s)/\delta_e(s)$ in Equation 4.738. The result is

$$\frac{\Delta\alpha(s)}{\Delta\delta_e(s)} = \frac{K_\alpha(s^2 + d_1 s + d_0)}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \quad (4.758)$$

where $K_\alpha = -0.141$, $d_1 = 0.0400$, and $d_0 = 0.0235$ are from “Ch4_CaseStudy1.m.”

Substituting Equations 4.738 and 4.758 into Equation 4.757 gives

$$G_{\dot{z}'}(s) = -\bar{u} \left[\frac{K_\theta(s + c_1)(s + c_2)}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} - \frac{K_\alpha(s^2 + d_1 s + d_0)}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \right] \quad (4.759)$$

$$\Rightarrow G_{\dot{z}'}(s) = -\frac{-\bar{u}[(K_\theta - K_\alpha)s^2 + \{K_\theta(c_1 + c_2) - K_\alpha d_1\}s + K_\theta c_1 c_2 - K_\alpha d_0]}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \quad (4.760)$$

$$\Rightarrow G_{\dot{z}'}(s) = \frac{\lambda_2 s^2 + \lambda_1 s + \lambda_0}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \quad (4.761)$$

$$\lambda_2 = -\bar{u}(K_\theta - K_\alpha), \lambda_1 = -\bar{u}[K_\theta(c_1 + c_2) - K_\alpha d_1], \lambda_0 = -\bar{u}(K_\theta c_1 c_2 - K_\alpha d_0) \quad (4.762)$$

From “Ch4_CaseStudy1.m,” $\lambda_2 = 0$, $\lambda_1 = 4.5768$, and $\lambda_0 = 0.0771$.

For a step input in elevator deflection of A° , Equation 4.761 and $\lambda_2 = 0$ give

$$\dot{z}'(s) = \frac{\lambda_1 s + \lambda_0}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \left(\frac{A}{s} \right) \quad (4.763)$$

The partial fraction expansion of $\dot{z}'(s)$ is

$$\dot{z}'(s) = A \left[\frac{R_1}{s - p_1} + \frac{R_2}{s - p_2} + \frac{R_3}{s - p_3} + \frac{R_4}{s - p_4} + \frac{R_5}{s} \right] \quad (4.764)$$

where the residues, evaluated in “Ch4_CaseStudy1.m,” are

$$R_{1,2} = 3.7283 \pm j0.4124, R_{3,4} = -6.8081 \pm j21.2231, R_5 = 6.1596$$

From Equations 4.763 and 4.764, the final value of \dot{z}' is given by

$$\dot{z}'_{ss} = \frac{A\lambda_0}{b_1 b_2} = AR_5 \quad (4.765)$$

The step response is from Equation 4.764,

$$\dot{z}'(t) = A[R_1 e^{p_1 t} + R_2 e^{p_2 t} + R_3 e^{p_3 t} + R_4 e^{p_4 t} + R_5] \quad (4.766)$$

Equation 4.766 is converted to a trigonometric form with real coefficients and real exponents similar to Equation 4.743 for $\Delta\theta(t)$. The unit step response is graphed in Figure 4.105. According to Equation 4.765, the steady-state value $\dot{z}'_{ss} = AR_5 = 1 \times 6.1596 \text{ ft/s}$.

The change in altitude $\Delta z'(t)$ resulting from a step change in elevator input is obtained by integration of $\dot{z}'(t)$. From Equation 4.763,

$$\Delta z'(s) = \frac{1}{s} \dot{z}'(s) = \frac{1}{s} \left[\frac{\lambda_1 s + \lambda_0}{(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \right] \frac{A}{s} \quad (4.767)$$

$$= \frac{A(\lambda_1 s + \lambda_0)}{s^2(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \quad (4.768)$$

The inverse transform of Equation 4.768 is left as an exercise problem. The change in altitude $\Delta z'(t)$ is graphed in Figure 4.105 below the derivative $d\dot{z}'/dt$.

The phugoid mode is an undesirable fact of life when it comes to control of an aircraft. In the previous example, it takes 300–400 s for the plane to establish a new steady-state pitch and rate of descent following a step change in the elevator position.

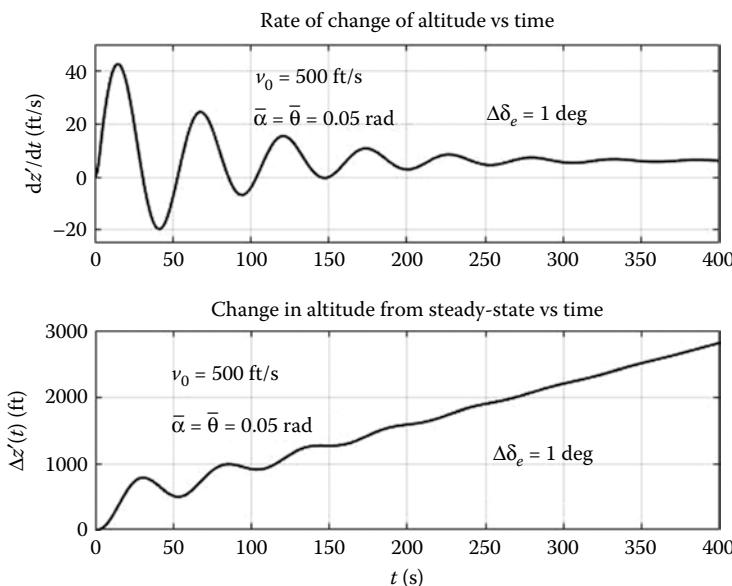


FIGURE 4.105 Changes in altitude rate and altitude from steady-state flight conditions.

Consider a scenario where the plane is required to decrease its cruising altitude by some amount. One approach is for the pilot to pull back on the yoke to increase the elevator deflection from its neutral position, which produces steady-state level flight conditions. The plane will begin a descent similar to the one shown in [Figure 4.105](#). The actual descent will depend on the magnitude of the elevator deflection. Some time later, the yoke is returned to the neutral position, and the plane returns to level flight conditions at a reduced altitude. To illustrate, suppose the pilot's action results in an elevator deflection of $\hat{\Delta\delta}_e$ degree for a period of T_{pulse} s. The aircraft's altitude response to the pulse input in elevator deflection is obtained as the difference between the step response and the delayed step response, that is,

$$\Delta z_p(t) = \hat{\Delta\delta}_p \Delta z_l(t) - \hat{\Delta\delta}_e \Delta z_l(t - T_{pulse}) \hat{u}(t - T_{pulse}) \quad (4.769)$$

where

$\Delta z_l(t)$ is the change in altitude response to a unit step elevator deflection

$\hat{u}(t - T_{pulse})$ is the unit step function starting at $t = T_{pulse}$

$\Delta z_p(t)$ is the change in altitude response to a pulse elevator deflection of $\hat{\Delta\delta}_e$ deg lasting T_{pulse} s

For a 5° elevator pulse input of 30 s, the aircraft's descent is computed according to Equation 4.769 in “*Ch4_CaseStudy1.m*” and shown in [Figure 4.106](#). The label “open-loop” refers to the lack of feedback used to determine the control surface deflection $\Delta\delta_e(t)$.

The open-loop response settles at a value of approximately 927.5 ft once the phugoid oscillations have disappeared. Some form of corrective action is necessary to dampen the excessive phugoid mode oscillations. A feedback control system or autopilot can automate the process without relying on human input.

[Figure 4.107](#) is a simplified block diagram of a control system for regulating an aircraft's altitude. Sensors convert the plane's altitude and rate of descent (or ascent) to voltages, which are transmitted to the autopilot. In [Figure 4.107](#), the gain of the altitude sensor K_z is shown combined with the controller transfer function $G_C(s)$, allowing the command signal Δz_{com} to be in ft rather than volts. (Note that the' symbol designating inertial coordinates is dropped from here on.)

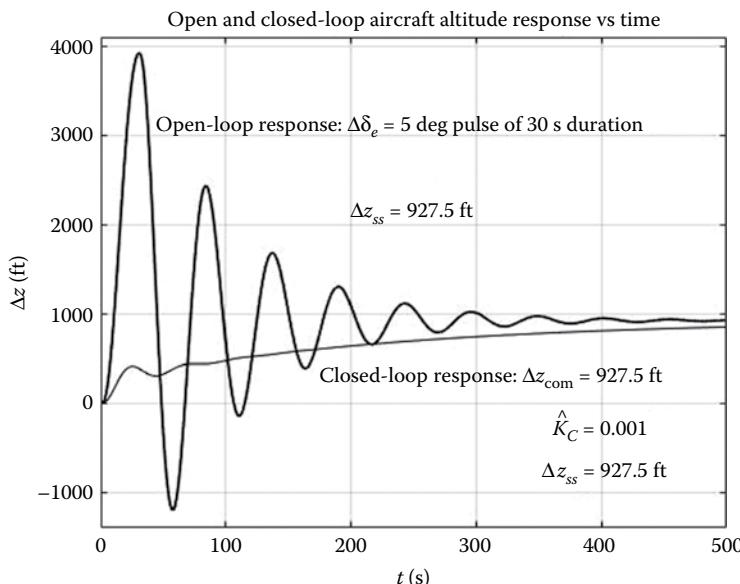


FIGURE 4.106 Open- and closed-loop altitude response vs. time.

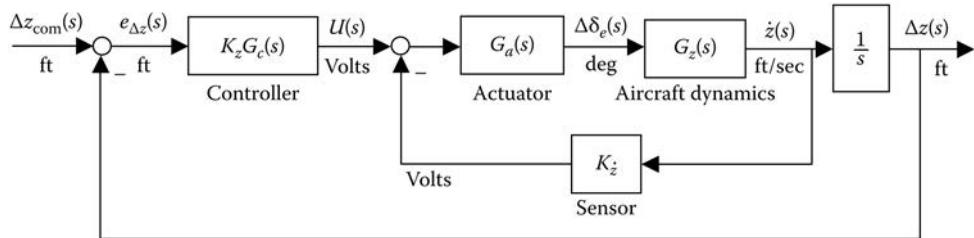


FIGURE 4.107 Block diagram for altitude control system.

The inner loop provides feedback of the altitude rate, which improves the damping and speed of the outer altitude control loop. There are several ways of obtaining the closed-loop transfer function $\Delta z(s)/\Delta z_{\text{com}}(s)$. The inner loop can be reduced to

$$\frac{\dot{z}(s)}{U(s)} = \frac{G_a(s)G_z(s)}{1 + K_z G_a(s)G_z(s)} \quad (4.770)$$

Using the same block diagram reduction formula for the outer loop gives

$$\frac{\Delta z(s)}{\Delta z_{\text{com}}(s)} = \frac{K_z G_c(s)[\dot{z}(s)/U(s)]1/s}{1 + K_z G_c(s)[\dot{z}(s)/U(s)]1/s} \quad (4.771)$$

$$= \frac{K_z G_c(s)[G_a(s)G_z(s)/(1 + K_z G_a(s)G_z(s))]1/s}{1 + K_z G_c(s)[G_a(s)G_z(s)/(1 + K_z G_a(s)G_z(s))]1/s} \quad (4.772)$$

$$= \frac{K_z G_c(s)G_a(s)G_z(s)}{[1 + K_z G_a(s)G_z(s)]s + K_z G_c(s)G_a(s)G_z(s)} \quad (4.773)$$

To start with, a proportional controller $G_c(s) = K_c$ is considered. The product of the gain K_z and controller gain K_c is \hat{K}_c , that is, $\hat{K}_c = K_z K_c$ is the effective controller gain for design purposes. For now, we ignore the actuator dynamics and let $G_a(s) = K_a$ measured in deg/volt. Equation 4.773 becomes

$$\frac{\Delta z(s)}{\Delta z_{\text{com}}(s)} = \frac{\hat{K}_c K_a G_z(s)}{[1 + K_z K_a G_z(s)]s + \hat{K}_c K_a G_z(s)} \quad (4.774)$$

The DC gain of the autopilot is

$$\lim_{s \rightarrow 0} \frac{\Delta z(s)}{\Delta z_{\text{com}}(s)} = \lim_{s \rightarrow 0} \frac{\hat{K}_c K_a G_z(s)}{[1 + K_z K_a G_z(s)]s + \hat{K}_c K_a G_z(s)} = 1 \quad (4.775)$$

Substituting Equation 4.761 with $\lambda_2 = 0$ for $G_z(s)$ into Equation 4.774 gives

$$\frac{\Delta z(s)}{\Delta z_{\text{com}}(s)} = \frac{\hat{K}_c K_a (\lambda_1 s + \lambda_0)}{s^5 + \mu_4 s^4 + \mu_3 s^3 + \mu_2 s^2 + \mu_1 s + \mu_0} \quad (4.776)$$

$$\left. \begin{array}{l} \mu_4 = a_1 + a_2 \\ \mu_3 = a_1 a_2 + b_1 + b_2 \\ \mu_2 = a_1 a_2 + a_2 b_1 + K_z K_a \lambda_1 \\ \mu_1 = b_1 b_2 + K_z K_a \lambda_0 + \hat{K}_c K_a \lambda_1 \\ \mu_0 = \hat{K}_c K_a \lambda_0 \end{array} \right\} \quad (4.777)$$

“Ch4_CaseStudy1.m” creates a system object for the control system transfer function in Equation 4.776 and then issues the MATLAB “step” command to acquire the unit step response values, which are multiplied by Δz_{com} and then plotted. The statements are

```
num_cs_z = Kc_hat*Ka*[lambda1 lambda0];
den_cs_z = [1 mu4 mu3 mu2 mu1 mu0];
sys_cs_z = tf(num_cs_z, den_cs_z)
T = linspace(0, 500, 1000); % t array for step response
[Y, T] = step(sys_cs_z, T); % Y is unit step response of control system
z_com = 927.5; % command input (ft)
z_cs = z_com*Y; % control system response to z_com
plot(T, z_cs, 'r')
```

Numerical values used to obtain the closed-loop response in Figure 4.106 were $\Delta z_{\text{com}} = 927.5$ ft, $K_a = 1^{\circ}/V$, $K_z = 0.1$ volt/ft/s, and $\hat{K}_c = 0.001$. The closed-loop transfer function corresponding to those values is

$$\frac{\Delta z(s)}{\Delta z_{\text{com}}(s)} = \frac{0.004577s + 0.00007713}{s^5 + 1.27s^4 + 0.9247s^3 + 0.08634s^2 + 0.01787s + 0.00007713} \quad (4.778)$$

Both responses in Figure 4.106 approach 927.5 ft; however, the closed-loop response is far superior to the open-loop pulse response. The elevator deflection in the closed-loop system response must be small enough to justify the use of the linearized model in Equation 4.735, which assumes small deviations in u , α , q , and θ . The small angle approximations and omission of high-order terms, key to the linearized model’s accuracy, may not hold if there are sizable changes in any of the responses. We must look at a graph of $\Delta \delta_e(t)$ responsible for the closed-loop response in Figure 4.106.

$\Delta \delta_e(s)/\Delta z_{\text{com}}(s)$ can be obtained by observing from Figure 4.107 that

$$\Delta z(s) = \frac{1}{s} G_z(s) \Delta \delta_e(s) \quad (4.779)$$

Solving Equation 4.779 for $\Delta \delta_e(s)$ and then dividing both sides by $\Delta z_{\text{com}}(s)$ lead to

$$\frac{\Delta \delta_e(s)}{\Delta z_{\text{com}}(s)} = \frac{s}{G_z(s)} \frac{\Delta z(s)}{\Delta z_{\text{com}}(s)} \quad (4.780)$$

Substituting for $G_z(s)$ the expression in Equation 4.761 gives

$$\frac{\Delta \delta_e(s)}{\Delta z_{\text{com}}(s)} = \frac{\hat{K}_c K_a s (s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)}{s^5 + \mu_4 s^4 + \mu_3 s^3 + \mu_2 s^2 + \mu_1 s + \mu_0} \quad (4.781)$$

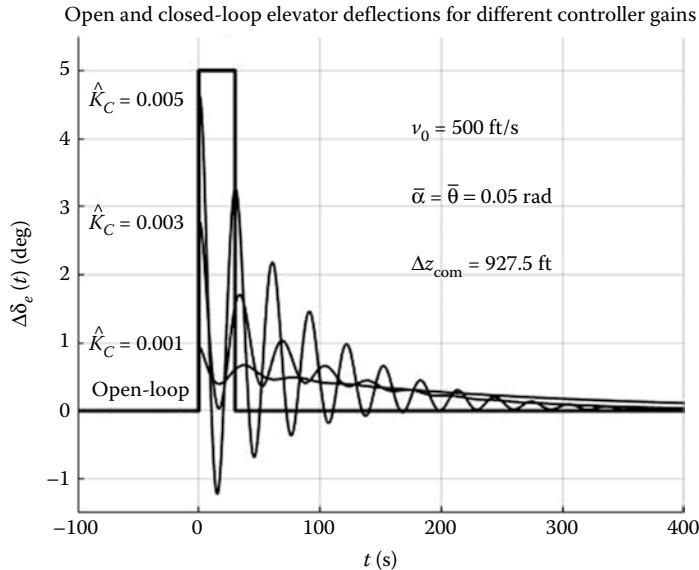


FIGURE 4.108 Elevator response for open- and closed-loop control of altitude.

The closed-loop elevator and altitude step responses for $\hat{K}_C = 0.001, 0.003$, and 0.005 along with the open-loop response are shown in Figures 4.108 and 4.109.

Looking at Figure 4.108, it is clear that the closed-loop system elevator input $\Delta\delta_e(t)$, $t \geq 0$ remains less than the 5° pulse amplitude in the open-loop system. It is left as an exercise problem to investigate the deviations Δu , $\Delta\alpha$, q , and $\Delta\theta$ as well.

The proportional gain compensator for the autopilot is far too simplistic; however, the results are fairly dramatic even for this simple design. One of the problems with this design is related to stability. The sluggish response ($\hat{K}_C = 0.001$) in Figure 4.109 is the most stable, yet the location of the closed-loop system poles, which determine the transient response, is far from optimal. Table 4.13 lists the location of the closed-loop system poles corresponding to the values of \hat{K}_C .

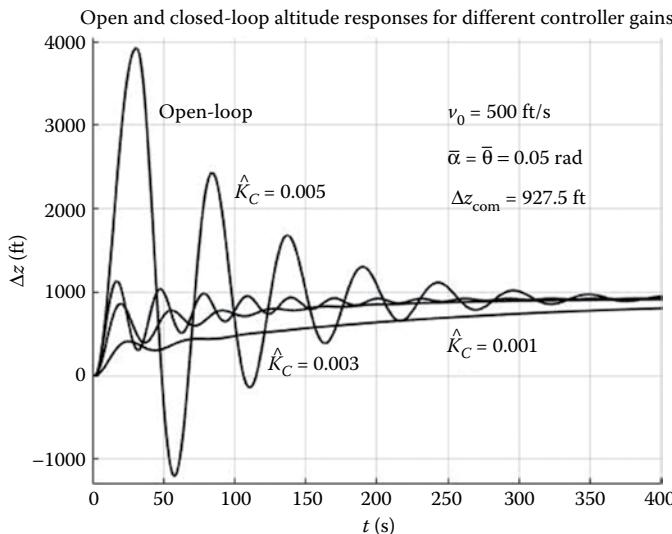


FIGURE 4.109 Altitude response for open- and closed-loop control.

TABLE 4.13
Closed-Loop System Poles for Autopilot with Proportional Control

\hat{K}_c	Closed-Loop Poles
0.001	$-0.5981 \pm j0.6759, -0.0347 \pm j0.1424, -0.0044$
0.003	$-0.6066 \pm j0.6748, -0.0240 \pm j0.1772, -0.0088$
0.005	$-0.6150 \pm j0.6742, -0.0145 \pm j0.2055, -0.0109$

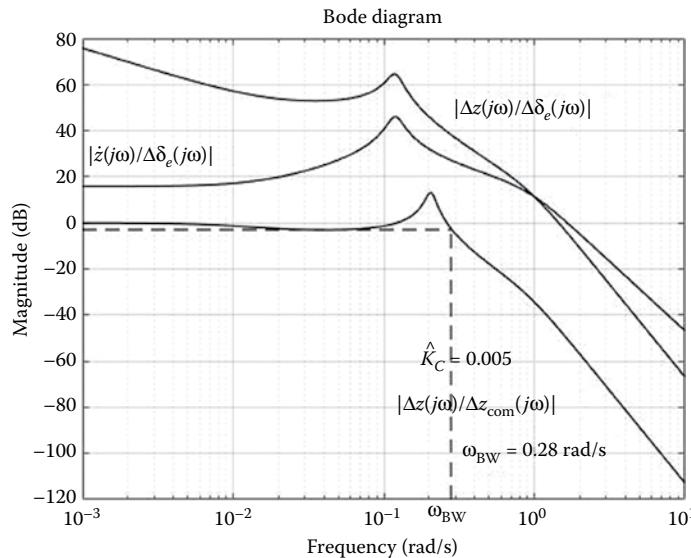


FIGURE 4.110 Open- and closed-loop magnitude functions.

The reader should consult one of the numerous control system texts for a discussion of more sophisticated compensators to achieve superior dynamic response with increased stability margins.

The gain (magnitude in db) of the open- and closed-loop frequency response functions is shown in Figure 4.110. The open-loop $|ż(j\omega)/\Delta\delta_e(j\omega)|$ is obtained from the transfer function in Equation 4.761, (recall $\lambda_2 = 0$). The open-loop $|\Delta z(j\omega)/\Delta\delta_e(j\omega)|$ comes from the transfer function

$$G_{\Delta z}(s) = \frac{\Delta z(s)}{\Delta \delta_e(s)} = \frac{\lambda_1 s + \lambda_0}{s(s^2 + a_1 s + b_1)(s^2 + a_2 s + b_2)} \quad (4.782)$$

The closed-loop $|\Delta z(j\omega)/\Delta z_{com}(j\omega)|$ is based on the transfer function in Equation 4.776 with $\hat{K}_C = 0.005$.

Note that the resonant frequency in the open-loop functions at the natural frequency of the phugoid $\omega_n = 0.1194$ rad/s (see Table 4.12). The closed-loop system gain is close to 0 db from DC to somewhat less than the resonant frequency. The bandwidth of the control system is approximately 0.28 rad/s.

4.11.1 DIGITAL SIMULATION OF AIRCRAFT LONGITUDINAL DYNAMICS

A digital simulation of longitudinal dynamics requires z -domain transfer functions to approximate the corresponding continuous-time transfer functions. A z -domain transfer function to

approximate the continuous-time transfer function in Equation 4.776 based on explicit Euler integration is

$$\frac{\Delta z(z)}{\Delta z_{\text{com}}(z)} = \left. \frac{\hat{K}_c K_a (\lambda_1 s + \lambda_0)}{s^5 + \mu_4 s^4 + \mu_3 s^3 + \mu_2 s^2 + \mu_1 s + \mu_0} \right|_{s=(z-1)/T} \quad (4.783)$$

Substituting $(z - 1)/T$ for s in Equation 4.783 leads to

$$\frac{\Delta z(z)}{\Delta z_{\text{com}}(z)} = \hat{K}_c K_a T^4 \left[\frac{\lambda_1 z - (\lambda_1 - \lambda_0 T)}{z^5 + \gamma_4 z^4 + \gamma_3 z^3 + \gamma_2 z^2 + \gamma_1 z + \gamma_0} \right] \quad (4.784)$$

where

$$\left. \begin{aligned} \gamma_4 &= -5 + \mu_4 T \\ \gamma_3 &= -10 - 4\mu_4 T + \mu_3 T^2 \\ \gamma_2 &= -10 + 6\mu_4 T + 3\mu_3 T^2 + \mu_2 T^3 \\ \gamma_1 &= 5 - 4\mu_4 T + 3\mu_3 T^2 + 2\mu_2 T^3 + \mu_1 T^4 \\ \gamma_0 &= -1 + \mu_4 T - \mu_3 T^2 + \mu_2 T^3 - \mu_1 T^4 + \mu_0 T^5 \end{aligned} \right\} \quad (4.785)$$

To simulate the altitude response to a step input command of magnitude $\Delta z_{\text{com}} = A$, we need the difference equation relating Δz_k and $(\Delta z_{\text{com}})_k$. Cross multiplying Equation 4.784 after multiplying numerator and denominator by z^{-5} gives

$$\begin{aligned} &(1 + \gamma_4 z^{-1} + \gamma_3 z^{-2} + \gamma_2 z^{-3} + \gamma_1 z^{-4} + \gamma_0 z^{-5}) \Delta z(z) \\ &= \hat{K}_c K_a T^4 [\lambda_1 z^{-4} - (\lambda_1 - \lambda_0 T) z^{-5}] \Delta z_{\text{com}}(z) \end{aligned} \quad (4.786)$$

Invert z -transforming both sides of Equation 4.786 and solving for Δz_k give

$$\begin{aligned} \Delta z_k &= -\gamma_4 \Delta z_{k-1} + \gamma_3 \Delta z_{k-2} - \gamma_2 \Delta z_{k-3} - \gamma_1 \Delta z_{k-4} + \gamma_0 \Delta z_{k-5} \\ &= \hat{K}_c K_a T^4 [\lambda_1 (\Delta z_{\text{com}})_{k-4} - (\lambda_1 - \lambda_0 T) (\Delta z_{\text{com}})_{k-5}] \end{aligned} \quad (4.787)$$

The first several values of Δz_k are evaluated sequentially from Equation 4.787 as

$$k = 0, 1, 2, 3 : \Delta z_k = 0 \quad (4.788)$$

$$k = 4 : \Delta z_4 = \hat{K}_c K_a T^4 \lambda_1 (\Delta z_{\text{com}})_0 = \hat{K}_c K_a T^4 \lambda_1 A \quad (4.789)$$

$$k = 5 : \Delta z_5 = -\gamma_4 \Delta z_4 + \hat{K}_c K_a T^4 [\lambda_1 (\Delta z_{\text{com}})_1 - (\lambda_1 - \lambda_0 T) (\Delta z_{\text{com}})_0] \quad (4.790)$$

$$= -\gamma_4 (\hat{K}_c K_a T^4 \lambda_1 A) + \hat{K}_c K_a T^4 [\lambda_1 A - (\lambda_1 - \lambda_0 T) A] \quad (4.791)$$

$$= \hat{K}_c K_a T^4 \lambda_1 A (-\gamma_4 \lambda_1 + \lambda_0 T) \quad (4.792)$$

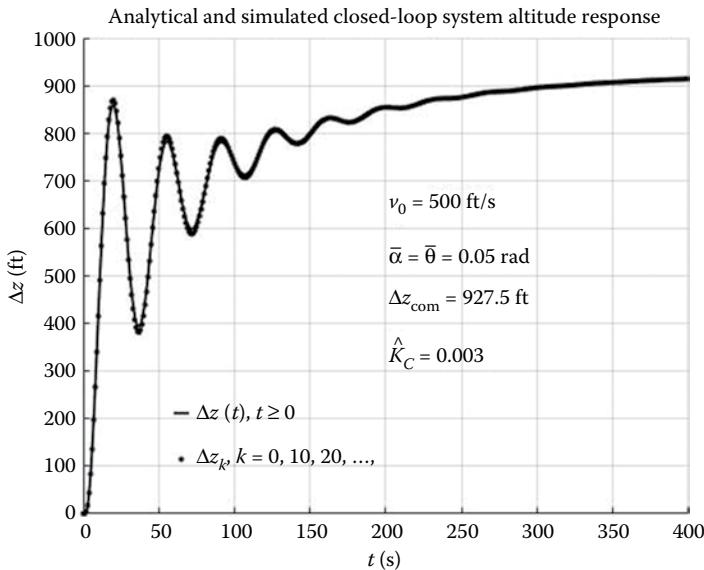


FIGURE 4.111 Altitude step responses of analytical and simulated closed-loop system.

$\Delta z_k, k = 6, 7, 8, \dots$ is computed by recursion according to

$$\Delta z_k = -\gamma_4 \Delta z_{k-1} - \gamma_3 \Delta z_{k-2} - \gamma_2 \Delta z_{k-3} - \gamma_1 \Delta z_{k-4} - \gamma_0 \Delta z_{k-5} + \hat{K}_C K_a T^5 A \lambda_0 \quad (4.793)$$

“Ch4_CaseStudy1.m” contains statements to implement Equations 4.788, 4.789, 4.792, and 4.793. The simulated altitude response of the closed-loop system with $\hat{K}_C = 0.003$ to the altitude command previously considered ($\Delta z_{\text{com}} = 927.5$ ft) is shown in Figure 4.111. The analytical solution previously plotted in Figure 4.109 is also presented. For purposes of clarity, the simulated points are plotted 1 s apart, that is, every 10th point is plotted. The exact and simulated responses are in close agreement.

Analytical and simulated (Euler $T = 0.1$ s) closed-loop system altitude response.

4.11.2 SIMULATION OF STATE VARIABLE MODEL

The linearized model describing the longitudinal dynamics of an aircraft was given in state variable form in Equation 4.735. Subsequent analysis of dynamic response, however, was done using transfer function descriptions relating a specific input, namely, $\Delta \delta_a(t)$, and a certain output, for example, $\Delta \theta(t)$, $\dot{z}(t)$, and $\Delta z(t)$. The conversion from a state-space description to input–output models is accomplished using Equation 4.736 or the MATLAB function “ss2tf” available in the control system toolbox. The remainder of this section is devoted to simulation of the aircraft dynamics based on the continuous-time state-space model

$$\Delta \underline{x} = A \Delta \underline{x} + B \Delta \underline{u}, \Delta \underline{y} = C \Delta \underline{x} + D \Delta \underline{u} \quad (4.794)$$

where

$A, B, \Delta x, \Delta u$ are evident from Equation 4.735

Δy is the output vector, which determines C and D

Suppose a simulation of the state equations using trapezoidal integration is required. Equation 3.121 is the difference equation for updating the discrete-time state based on trapezoidal

integration. It is repeated below (using the deviation variable notation) along with the equation for computing the output vector.

$$\underline{\Delta x}(n+1) = \left(I - \frac{1}{2} TA \right)^{-1} \left(I + \frac{1}{2} TA \right) \underline{\Delta x}(n) + \frac{1}{2} \left(I - \frac{1}{2} TA \right)^{-1} TB [\underline{\Delta u}(n) + \underline{\Delta u}(n+1)] \quad (4.795)$$

$$\underline{\Delta y}(n) = C \underline{\Delta x}(n) + D \underline{\Delta u}(n) \quad (4.796)$$

Equations 4.795 and 4.796 represent a straightforward approach to simulation of the state equations using trapezoidal integration. The equations are implemented in the script file “Ch4_CaseStudy1.m” for the case where $\underline{\Delta u} = [\Delta\delta_e \Delta\delta_T]^T = [5^\circ 0 \text{ lb}]^T$, $\underline{\Delta y} = [\Delta u \Delta a \Delta q \Delta\theta]^T$. Accordingly, C is the 4×4 identity matrix and D is a 4×2 matrix of zeros. The simulated output $\underline{\Delta y}(n) = [\Delta u(n) \Delta a(n) \Delta q(n) \Delta\theta(n)]^T$ was recorded for $T = 1, 5, 10$ s and the results graphed for $T = 1$ and 10 s in Figure 4.112.

There was very little difference in the outputs for $T = 1$ and 5 s suggesting that the higher value is appropriate for further simulation studies using trapezoidal integration.

Setting $\underline{\Delta \dot{x}} = 0$ in Equation 4.794 and solving for $\underline{\Delta x}$ at steady state give

$$\underline{\Delta x}_{ss} = -A^{-1}B\underline{\Delta u} \quad (4.797)$$

$$\begin{aligned} \Rightarrow \begin{bmatrix} \Delta u_{ss} \\ \Delta \alpha_{ss} \\ \Delta q_{ss} \\ \Delta \theta_{ss} \end{bmatrix} &= - \begin{bmatrix} -0.04 & 11.59 & 0 & -32.2 \\ -0.00073 & -0.65 & 1 & 0 \\ 0.000048 & -0.49 & -0.58 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0.1 \\ 0 & 0 \\ -0.014 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 117.83 \text{ ft/s} \\ -0.13 \text{ rad} \\ 0 \text{ rad/s} \\ -0.19 \text{ rad} \end{bmatrix} \end{aligned} \quad (4.798)$$

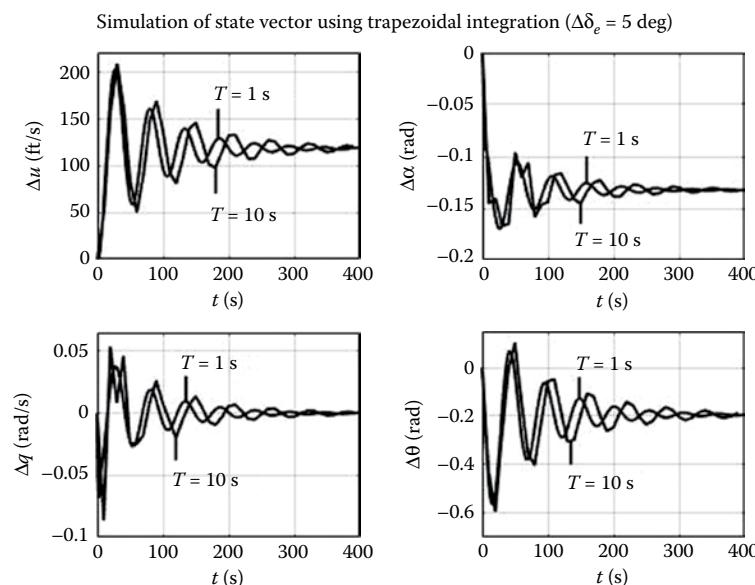


FIGURE 4.112 Simulation of state vector using trapezoidal integration ($\delta_e = 5$ deg).

Setting $\Delta\underline{x}(n+1) = \Delta\underline{x}(n) = \Delta\underline{x}(\infty)$ in Equation 4.795,

$$\begin{aligned}\Delta\underline{x}(\infty) &= \left(I - \frac{1}{2}TA \right)^{-1} \left(I + \frac{1}{2}TA \right) \Delta\underline{x}(\infty) \\ &\quad + \frac{1}{2} \left(I - \frac{1}{2}TA \right)^{-1} TB[\Delta\underline{u}(\infty) + \Delta\underline{u}(\infty)]\end{aligned}\tag{4.799}$$

Solving for the steady-state vector $\Delta\underline{x}(\infty)$ gives

$$\begin{aligned}\Delta\underline{x}(\infty) &= \left[I - \left(I - \frac{1}{2}TA \right)^{-1} \left(I + \frac{1}{2}TA \right) \right]^{-1} \left(I + \frac{1}{2}TA \right)^{-1} TB\Delta\underline{u}(\infty) \\ &= [117.83 \text{ ft/s} \quad -0.13 \text{ rad} \quad 0 \text{ rad/s} \quad -0.19 \text{ rad}]^T\end{aligned}\tag{4.800}$$

The continuous-time Δx_{ss} and discrete-time (simulated) $\Delta x(\infty)$ are identical, in agreement with the values observed in [Figure 4.112](#).

EXERCISES

- 4.86 Prove the relationship in Equation 4.742 involving complex numbers.
- 4.87 Use the control system toolbox to
- Find the transfer functions $\Delta u(s)/\Delta\delta_T(s)$, $\Delta\alpha(s)/\Delta\delta_T(s)$, $q(s)/\Delta\delta_T(s)$, $\Delta\theta(s)/\Delta\delta_T(s)$.
 - Plot the unit step responses for the linearized model in Equation 4.735 with $\Delta y = \Delta\underline{x}$.
- 4.88 Find $\Delta z'(t)$ by inversion of $\Delta z'(s)$ in Equation 4.768.
- 4.89
 - Use a similar approach to the one for finding $\Delta z'(s)/\Delta\delta_e(s)$ to determine $\Delta x'(s)/\Delta\delta_e(s)$.
 - Use the control system toolbox to plot $\Delta x'(t)$ in response to a step change in elevator input of 5° .
 - Find the response $\Delta z'(t)$ to the same input.
 - Plot the aircraft's flight trajectory $\Delta z'$ vs. $\Delta x'$ for $(0 \leq \Delta x' \leq 25,000 \text{ ft})$.
- 4.90 Find the time duration of a -5° elevator pulse input required to increase the plane's elevation by 1500 ft.
- 4.91 The actuator that controls elevator deflection was assumed to exhibit negligible dynamics in the typical range of frequencies encountered. The actuator transfer function is first order with gain $K_a = 1^\circ/\text{V}$ and time constant $\tau_a = 0.4 \text{ s}$.
- Find the closed-loop transfer functions $\Delta z(s)/\Delta z_{\text{com}}(s)$ and $\Delta\delta_e(s)/\Delta z_{\text{com}}(s)$ with the actuator dynamics included. Express both transfer functions as a ratio of polynomials similar to Equations 4.776 and 4.781.
 - Find the closed-loop system ($\hat{K}_C = 0.005$) poles with and without the actuator dynamics. Comment on the results.
With $\hat{K}_C = 0.005$, verify the assumption of negligible actuator dynamics by
 - Plotting the frequency response of the open-loop transfer function with and without actuator dynamics.
 - Comparing the elevator deflection response when $\Delta z_{\text{com}} = 500 \text{ ft}$ with and without the actuator dynamics.
 - Comparing the aircraft altitude response when $\Delta z_{\text{com}} = 500 \text{ ft}$ with and without the actuator dynamics.

- 4.92 For the conditions in [Figure 4.111](#), find the maximum deviation between the analytical and simulated altitude responses when using Euler integration with the step sizes shown in [Table E4.92](#). Fill in [Table E4.92](#).

TABLE E4.92

Step Size	$T = 0.025 \text{ s}$	$T = 0.05 \text{ s}$	$T = 0.1 \text{ s}$	$T = 0.25 \text{ s}$
Max $ \Delta z_{\text{anal}} - \Delta z_{\text{sim}} $				

- 4.93 Starting with the open-loop transfer function $G_{\Delta\delta_e}^{\Delta\theta}(s) = \Delta\theta(s)/\Delta\delta_e(s)$ in [Equation 4.738](#),
- Use Tustin's method with a sample time of $T = 1 \text{ s}$ to obtain a discrete-time system approximation $G_{\Delta\delta_e}^{\Delta\theta}(z)$. Use the control system toolbox function "c2d" if available, otherwise be prepared for some tedious algebraic work.
 - Use the pulse transfer function $G_{\Delta\delta_e}^{\Delta\theta}(z)$ to find the difference equation relating $\Delta\theta_k$ and $(\Delta\delta_e)_k$.
 - Find the aircraft's pitch response to a unit step change in elevator position by recursive solution of the difference equation.
 - Compare the simulated pitch step response in part (c) to the continuous-time pitch step response shown in [Figure 4.103](#).

4.12 CASE STUDY: NOTCH FILTER FOR ELECTROCARDIOGRAPH WAVEFORM

An electrocardiograph (ECG) signal is corrupted with 60 Hz noise from an electrical power source. A portion of the noisy signal, sampled regularly at 0.004 s intervals, is shown in [Figure 4.113](#).

A notch filter is needed to remove the noise. One realization of a second-order filter transfer function is given by (Orfanidis 1996)

$$H(z) = \frac{Y(z)}{U(z)} = b \left[\frac{1 - 2(\cos \omega_0 T)z^{-1} + z^{-2}}{1 - 2b(\cos \omega_0 T) + (2b - 1)z^{-2}} \right] \quad (4.801)$$

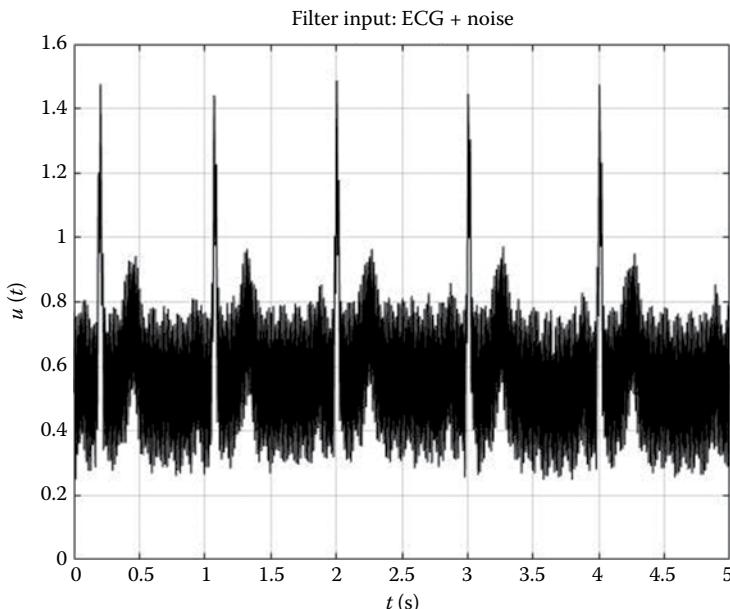


FIGURE 4.113 ECG signal corrupted with 60 Hz noise sampled at $T = 0.004 \text{ s}$ intervals.

where ω_0 is the notch frequency (in rad/s). The filter parameter Q relates the notch frequency ω_0 to the width of the 3 db interval $\Delta\omega$ on a plot of $|H(e^{j\omega T})|^2$ vs. ω .

$$Q = \frac{\omega_0}{\Delta\omega} \quad (4.802)$$

The higher Q is, the narrower is the 3 db interval $\Delta\omega$. The filter parameter b is obtained from

$$b = \frac{1}{1 + \tan(\omega_0 T / 2Q)} \quad (4.803)$$

Two notch filters will be investigated. One with $Q = 10$ and the other with $Q = 50$. The M-file "Ch4_CaseStudy2.m" computes the filter coefficients and plots both $|H(e^{j\omega T})|^2$ vs. ω and the magnitude function (in db), $|H(e^{j\omega T})|$ vs. ω (see Figures 4.114 through 4.117).

Note, when $|H(e^{j\omega T})|^2 = 0.5$ it is 3 db below the DC value $|H(e^{j0T})|^2 = 1$.

The filtered outputs are shown in Figures 4.118 and 4.119. There is little difference in the outputs of the two filters except for the longer transient period of the filter with $Q = 50$.

4.12.1 MULTINOTCH FILTERS

When more than one notch frequency exists, a multinotch filter design is required. The previous reference includes several methods of designing a multinotch filter. One approach is to simply use the singlenotch design for each notch frequency and cascade the respective filters. To illustrate, suppose the ECG signal contains a 25 Hz square wave noise signal like the one shown in Figure 4.120.

The noise $n(t)$ contains harmonics at multiples of the fundamental frequency $\omega_0 = 2\pi f_0 = 50\pi$ rad/s. The Fourier Series expansion of $n(t)$ is given by (see Exercise 4.95)

$$n(t) = \frac{1}{\pi} \sin \omega_0 t + \frac{1}{3\pi} \sin 3\omega_0 t + \frac{1}{5\pi} \sin 5\omega_0 t + \dots \quad (4.804)$$

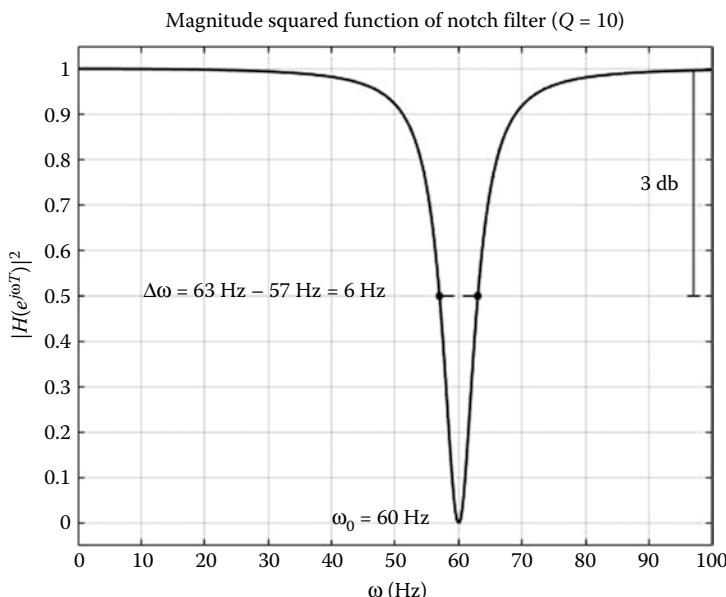


FIGURE 4.114 Magnitude squared function for notch filter ($Q = 10$).

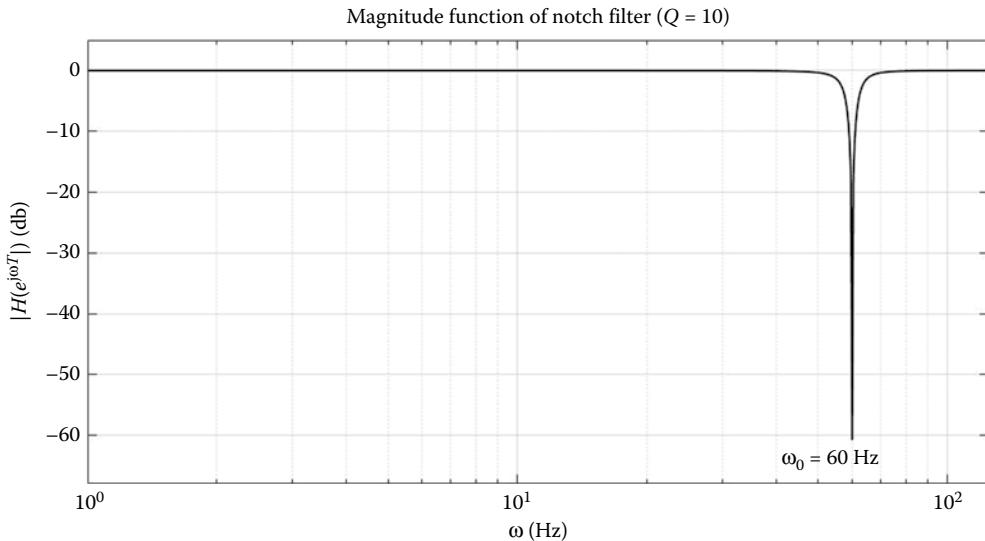


FIGURE 4.115 Magnitude function (in db) for notch filter ($Q = 10$).

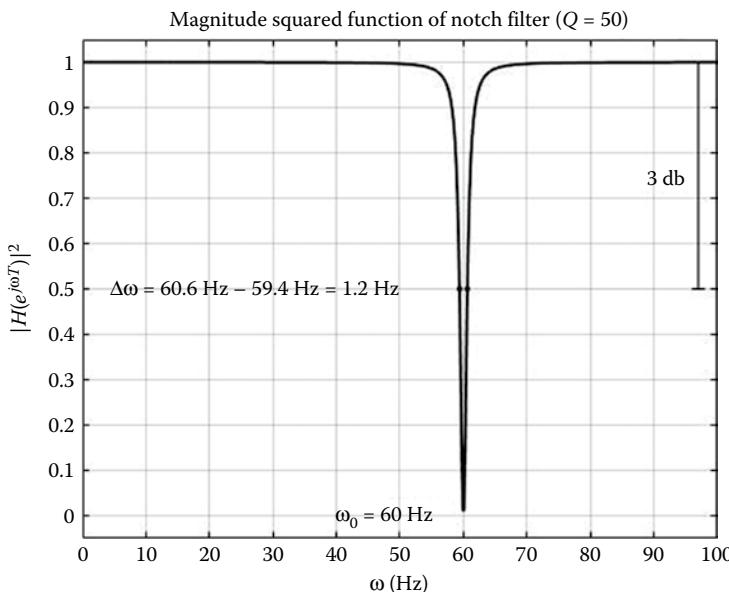


FIGURE 4.116 Magnitude squared function for notch filter ($Q = 50$).

EXAMPLE 4.37

A clean ECG signal, 10 s in duration, is sampled every $T_s = 0.004$ s and stored in the data file "Ch4_clean_ecg_10sec.mat." The time and signal data are stored in arrays "t" and "s."

- Sample the square wave noise shown in Figure 4.120 at the sampling frequency $\omega_s = 1/T_s$ and plot the sampled noise $n(t)$ and the noisy ECG signal $s(t) + n(t)$.
- Design notch filters:
 - $H\omega_0(z)$ to remove the fundamental frequency
 - $H_3\omega_3(z)$ to remove the first nonzero harmonic term
 - $H_5\omega_0(z)$ to remove the second nonzero harmonic term.

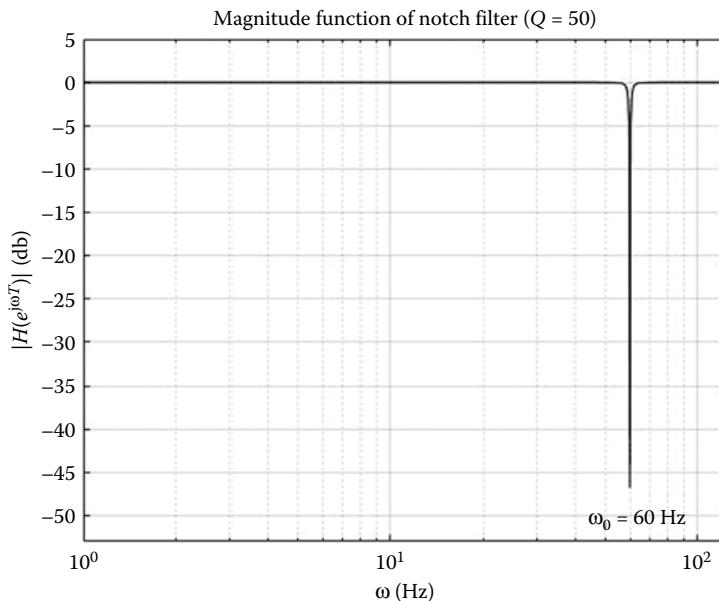


FIGURE 4.117 Magnitude function (in db) for notch filter ($Q = 50$).

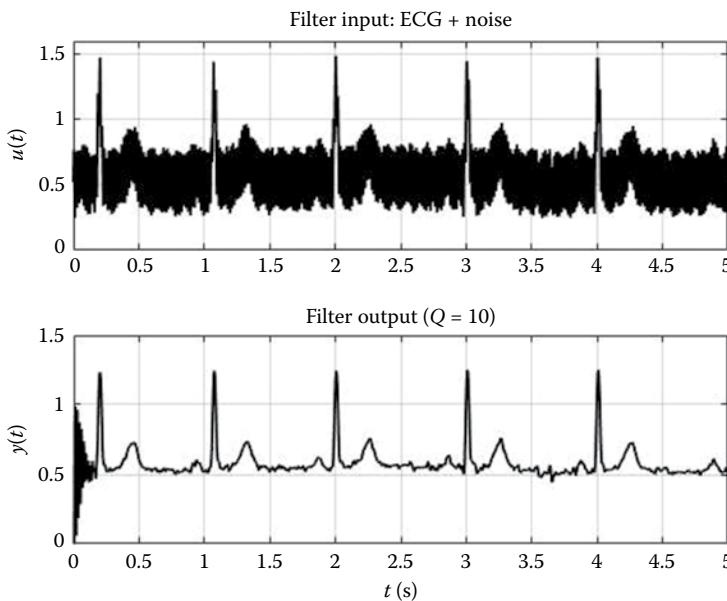


FIGURE 4.118 Output of notch filter ($Q = 10$).

Choose the Q values such that the 3 db width $\Delta\omega$ for $|H(e^{j\omega T})|^2$ versus ω is the same for each filter.

- Draw the magnitude function (in db) for the following filters:
 - $H_{\omega_0}(z)$
 - $H_{\omega_0}(z)H_{3\omega_0}(z)$
 - $H_{\omega_0}(z)H_{3\omega_0}(z)H_{5\omega_0}(z)$
- Filter the noisy ECG signal in part (a) using the three filters in part (c) and graph the results.

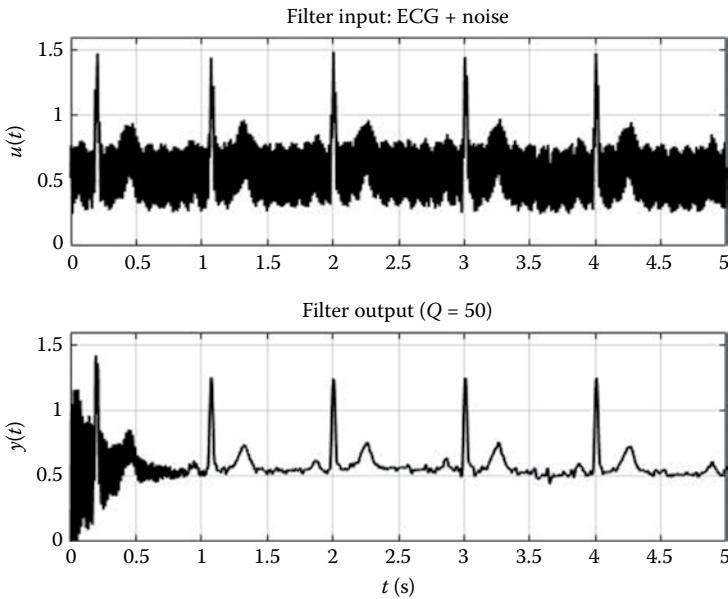


FIGURE 4.119 Output of notch filter ($Q = 50$).

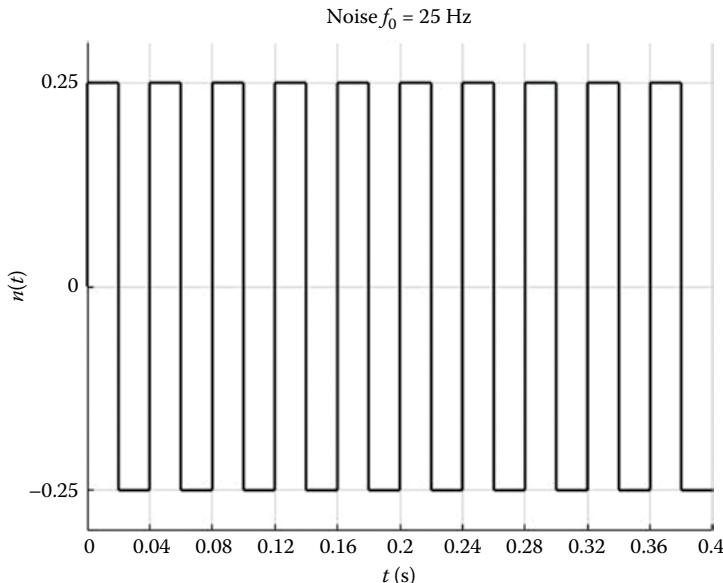


FIGURE 4.120 Square wave noise component of ECG signal.

- Figure 4.121 shows 5 s of the noise square wave $n(t)$ and the combined signal plus noise $s(t) + n(t)$.
- The filter parameter Q was chosen as 10 for the first filter. From Equation 4.802 the 3 db width $\Delta\omega = 2.5$ Hz. Using this value for notch frequencies $3\omega_0 = 75$ Hz and $5\omega_0 = 125$ Hz in Equation 4.802 gives

$$Q = \frac{3\omega_0}{\Delta\omega} = \frac{3(25)}{2.5} = 30, \quad Q = \frac{5\omega_0}{\Delta\omega} = \frac{5(25)}{2.5} = 50 \quad (4.805)$$

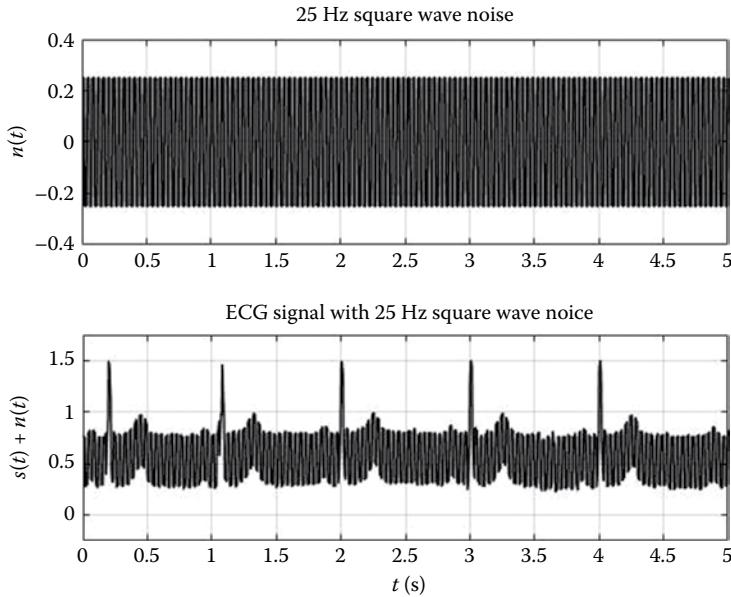


FIGURE 4.121 Square wave noise and noise-corrupted ECG signal.

The M-file “*Ch4_Ex4_37.m*” computes the filter coefficients for the three notch filters with Q values 10, 30, and 50 using Equations 4.801 and 4.803. The results are

$$H_{\omega_0}(z) = 0.9695 \left(\frac{1 - 1.6180z^{-1} + z^{-2}}{1 - 1.5687z^{-1} + 0.9391} \right) \quad (Q = 10) \quad (4.806)$$

$$H_{3\omega_0}(z) = 0.9695 \left(\frac{1 + 1.6180z^{-1} + z^{-2}}{1 + 0.5992z^{-1} + 0.9391} \right) \quad (Q = 30) \quad (4.807)$$

$$H_{5\omega_0}(z) = 0.9695 \left(\frac{1 + 2z^{-1} + z^{-2}}{1 + 1.9391z^{-1} + 0.9391} \right) \quad (Q = 50) \quad (4.808)$$

- c. “*Ch4_Ex4_37.m*” includes statements to plot the magnitude functions of $H_{\omega_0}(z)$ and the cascaded filters $H_{\omega_0}(z) H_{3\omega_0}(z)$ and $H_{\omega_0}(z) H_{5\omega_0}(z)$. The results are shown in Figures 4.122 through 4.124.
- d. The three filters are shown in Figure 4.125 with their corresponding inputs and outputs. Output of the filter with transfer function $H_{\omega_0}(z)$ in Equation 4.806 is shown in Figure 4.126. The simple notch filter was designed to remove the fundamental frequency term in Equation 4.804.

Output of the first filter $y_1(k)$ is passed to the notch filter with transfer function $H_{3\omega_0}(z)$ in Equation 4.807. Output $y_2(k)$ of the multinoth filter $H_{\omega_0}(z)H_{3\omega_0}(z)$ is shown in Figure 4.127. Finally, the output of the middle filter in Figure 4.125 is the input to the third filter in the series of cascaded filters. The output of the last filter $y_3(k)$ is plotted as $y_3(t)$ in Figure 4.128.

The multinoth filter output in Figure 4.128 is similar in appearance to the single notch filter outputs shown in Figures 4.118 and 4.119 (after the transient response has vanished) when the noise was a pure sinusoid at 60 Hz. Even though the square wave noise contains an infinite number of harmonics, that is, odd multiples of the fundamental

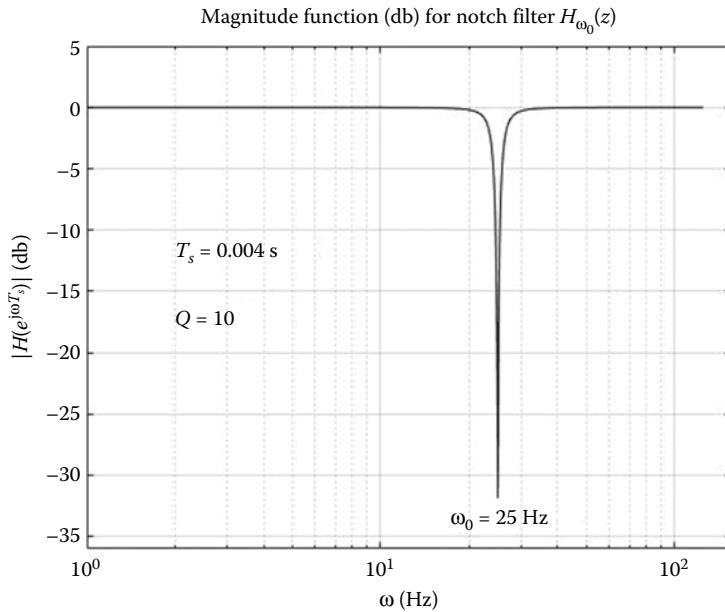


FIGURE 4.122 Magnitude function (db) for notch filter $H_{\omega_0}(z)$.

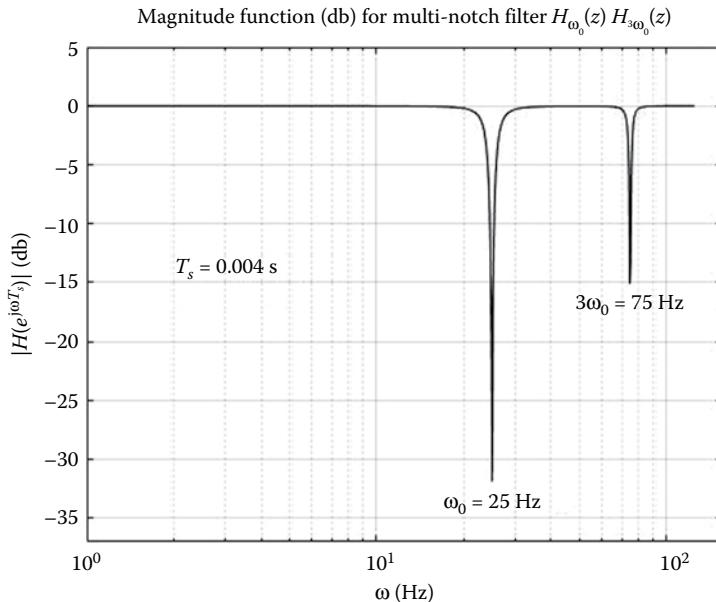


FIGURE 4.123 Magnitude function (db) for multinoth filter $H_{\omega_0}(z)H_{3\omega_0}(z)$.

frequency (see Equation 4.804), all but the first two nonzero harmonics $3\omega_0 = 75$ Hz and $5\omega_0 = 125$ Hz are above the Nyquist frequency $\omega_{nyq} = 0.5\omega_s = 0.5 \times (1/T_s) = 125$ Hz. Consequently, the harmonics at $7\omega_0 = 175$ Hz, $9\omega_0 = 225$ Hz, and so forth, are aliased back to the lower frequencies which are effectively removed by the multinoth filter in Figure 4.125.

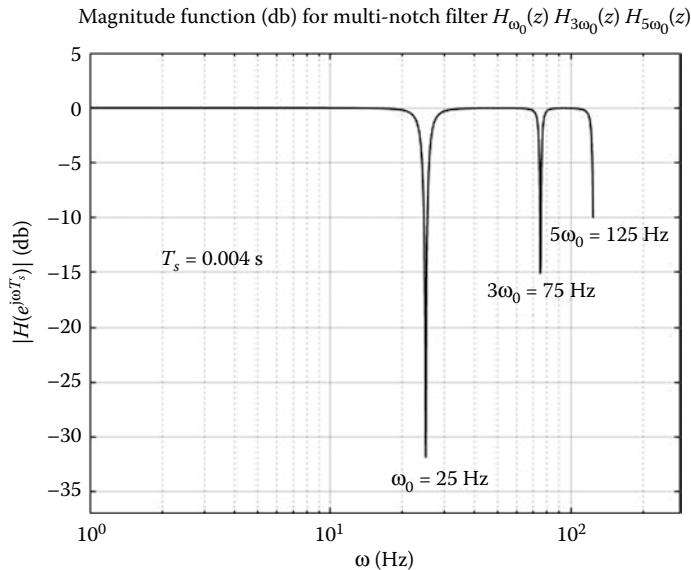


FIGURE 4.124 Magnitude function (db) for multinoth filter $H_{\omega_0}(z)H_{3\omega_0}(z)H_{5\omega_0}(z)$.

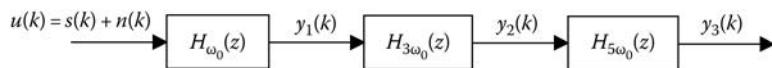


FIGURE 4.125 Multinoth filter for removing fundamental frequency and first two nonzero harmonics.

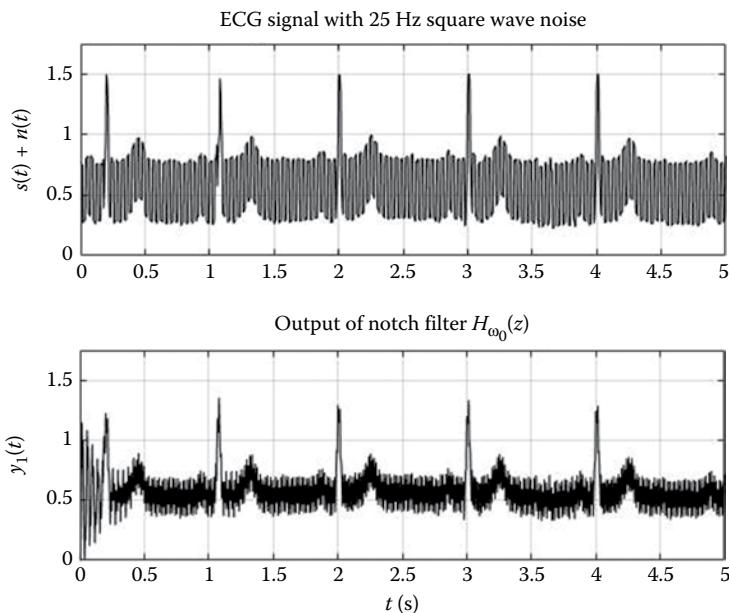


FIGURE 4.126 Input and output of notch filter $H_{\omega_0}(z)$.

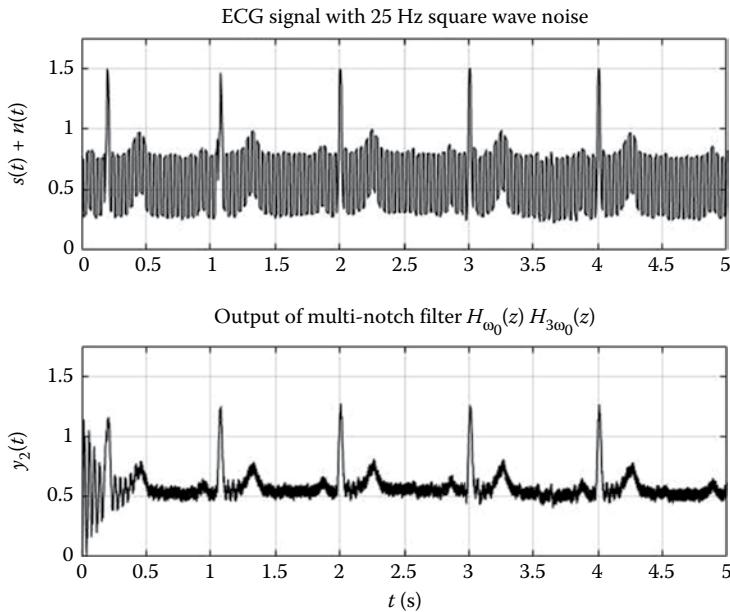


FIGURE 4.127 Input and output of multinoth filter $H_{\omega_0}(z)H_{3\omega_0}(z)$.

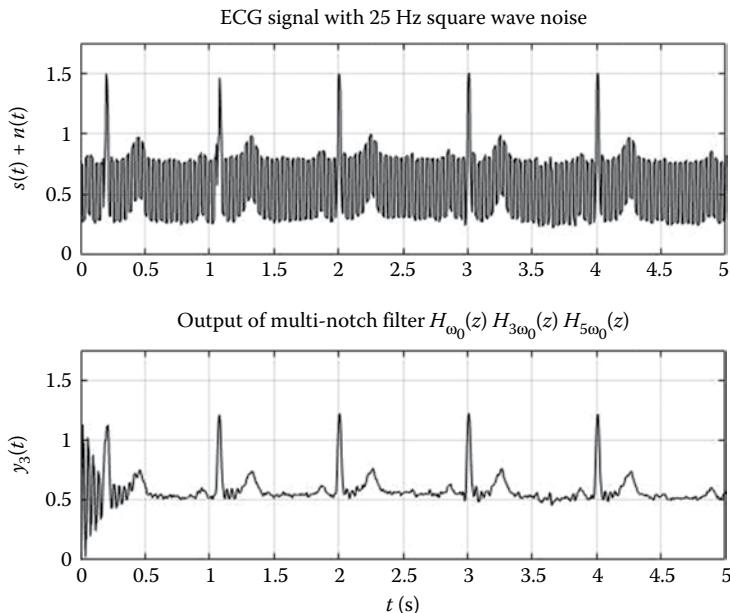
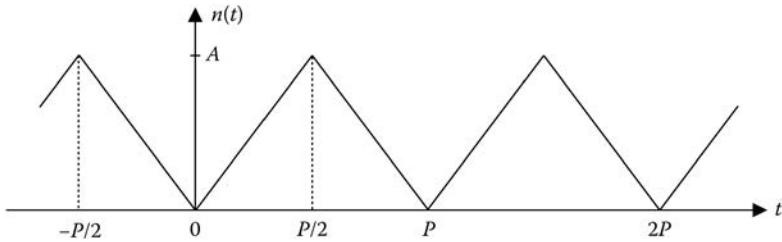


FIGURE 4.128 Input and output of multinoth filter $H_{\omega_0}(z)H_{3\omega_0}(z)H_{5\omega_0}(z)$.

EXERCISES

- 4.94 Create a noisy ECG signal $u(t_k)$ by starting with the clean signal $s(t_k)$, where $t_k = kT_s$, $k = 0, 1, 2, \dots$ ($T_s = 0.004$ s) in “Ch4_clean_ecg_10sec.mat.” Add a 50 Hz sinusoidal noise $n(t_k)$ with amplitude of 0.75.
- Design and implement an appropriate notch filter to remove the noise.

- b. Graph the filter input $u(t_k)$ and its output $y(t_k)$ below it.
c. Compare the clean ECG signal $s(t_k)$ and the filter output $y(t_k)$.
- 4.95 The clean ECG signal described in Exercise 4.94 is corrupted by the periodic noise $n(t)$ shown in [Figure E4.95](#). The period $P = 1/30$ s ($\omega_0 = 30$ Hz) and amplitude $A = 1$.

**FIGURE E4.95**

- a. Sample the noise at the frequency $\omega_s = 250$ Hz ($T_s = 0.004$ s), and add it to the clean ECG signal. Denote the corrupted signal by $u(t_k)$, where $t_k = kT_s$, $k = 0, 1, 2, 3, \dots$
b. Expand the noise in a Fourier series expansion,

$$n(t) = \frac{a_0}{2} + \sum_{k=1,2,\dots} (a_k \cos k\omega_0 t + b_k \sin k\omega_0 t),$$

$$\omega_0 = \frac{2\pi}{P} = \frac{2\pi}{1/30} = 60\pi \text{ rad/s}$$

$$a_k = \frac{2}{P} \int_{-P/2}^{P/2} n(t) \cos k\omega_0 t \, dt, \quad k = 0, 1, 2, \dots$$

$$b_k = \frac{2}{P} \int_{-P/2}^{P/2} n(t) \sin k\omega_0 t \, dt, \quad k = 1, 2, \dots$$

- c. Design and implement a multinotch filter to remove all the frequency components (except DC) below the Nyquist frequency $\omega_{nyq} = 0.5\omega_s = 125$ Hz.
d. Graph the filter input $u(t_k)$ and its output $y(t_k)$ below it.
e. Compare the clean ECG signal $s(t_k)$ and the filter output $y(t_k)$.



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

5 Simulink®

5.1 INTRODUCTION

This chapter serves as an introduction to the continuous simulation program, Simulink®. It is similar in many ways to its predecessors such as CSMP (Continuous System Modeling Program), ACSL (Advanced Continuous Simulation Language), TUTSIM (Twente University of Technology Simulator), MATRIX-X, STELLA, and EASY5. The major advantage of Simulink stems from its tight integration with MATLAB®, the data analysis and visualization program with its own structured programming language. The numerous (37 at the time of this printing) MATLAB toolboxes in diverse areas of engineering, science, and business extend the capabilities of Simulink.

In addition to the toolboxes, there are a number of Simulink blocksets that extend Simulink into various disciplines such as aerospace, communications, signal processing, image processing, and so forth. A complete list of toolboxes and blocksets with descriptions of each can be found at http://www.mathworks.com/products/product_listing/index.html.

Chapters 1 through 4 cover some basic essentials of linear continuous-and discrete-time systems. Elementary simulation techniques based on numerical integration are also introduced. In all but the simplest cases, the simulated solutions were programmed in MATLAB M-files.

The early continuous-time system simulation languages (CSSLs) consisted of individual sections, for example, “Initial,” “Dynamic,” “Derivative,” and “Terminal” with special demarcation headers for inputting constants and system parameters, calculating new parameters, setting initial conditions for the states, evaluating inputs over time, numerically integrating the state derivative vector, and computing the system outputs (Korn and Wait 1978). The continuous-time system dynamics were confined to a section containing expressions for the state derivatives. Lookup tables (in one or more dimensions) were often included in the section to evaluate the state derivatives. Crucial savings in simulation development time resulted from the built-in numerical integration routines and graphing capabilities.

Despite minor variations among the CSSLs, they were classified as “equation-oriented” because expressions for the state derivatives, difference equations, and outputs were entered on one or more lines in equation format. Later, general-purpose, block-oriented simulation programs emerged with powerful graphical user interfaces (GUIs). Dragging and dropping blocks from libraries containing blocks of similar functionality is the most intuitive way for creating a simulation model. Even more so than equation-oriented CSSLs, block-oriented simulation programs such as Simulink free the simulationist from the tedious grunt work required to develop a model structure, implement numerical integration, and produce useful output.

Our initial exploration of Simulink in this chapter is merely the “tip of the iceberg.” Later chapters will delve further into the world of Simulink and its capabilities.

5.2 BUILDING A SIMULINK MODEL

To begin our introduction to Simulink we will demonstrate the procedure for creating a model of a simple system and run the model to obtain useful information about its dynamic response. Our purpose here is to get comfortable with the Simulink user interface at a macroscopic level. The Math Works web site at mathworks.com provides excellent tutorials for the beginner interested in getting started with Simulink.

5.2.1 THE SIMULINK LIBRARY

The Simulink library contains blocks for representing the mathematical models of commonly occurring components in dynamic systems. The blocks are grouped in sublibraries according to function.

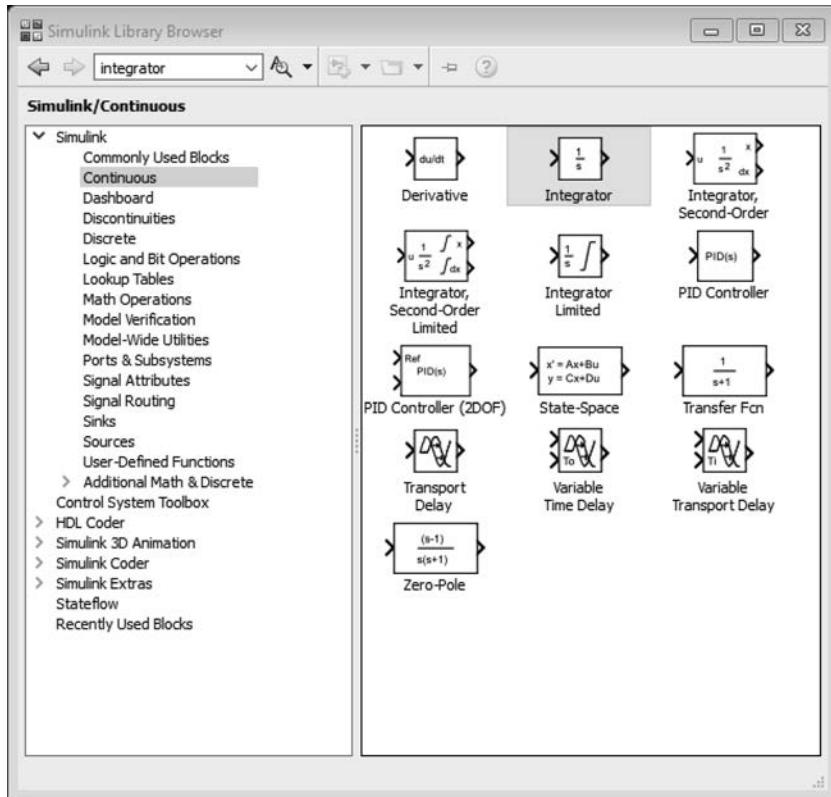


FIGURE 5.1 The Simulink Library Browser.

The standard Simulink sublibraries are shown in the left pane of Figure 5.1. The blocks residing in the selected “Continuous” sublibrary are shown in the right pane. The “Integrator” block is selected and the transfer function, $1/s$, is used to designate the integrator.

Building a Simulink model of a system consists of selecting the appropriate blocks and connecting them in a way that represents the mathematical model. Inputs, when present, are implemented using blocks from the “Sources” sublibrary which can generate a host of input signals. Simulation output is saved and displayed using various blocks such as “Scopes,” “XY Graphs,” and “Displays” from the “Sinks” sublibrary.

Our first Simulink model will simulate the dynamics of the linear second-order system model introduced in Chapter 2. The differential equation is

$$\frac{d^2}{dt^2}y(t) + 2\zeta\omega_n \frac{d}{dt}y(t) + \omega_n^2 y(t) = K\omega_n^2 u(t) \quad (5.1)$$

Assuming for the moment that the second derivative term d^2y/dt^2 is present in a new model window, it can be twice integrated as shown in Figure 5.2 where “ydd,” “yd,” and “y” are the Simulink variable names. The “Integrator” blocks are dragged or copied from the “Continuous” sublibrary into the model window.

By inspection of Equation 5.1, the second derivative term is a linear combination of the input $u(t)$, the output $y(t)$, and its first derivative dy/dt . The Simulink Library Browser allows us to search the standard sublibraries for the blocks needed to “build” the second derivative and thus complete the Simulink model.

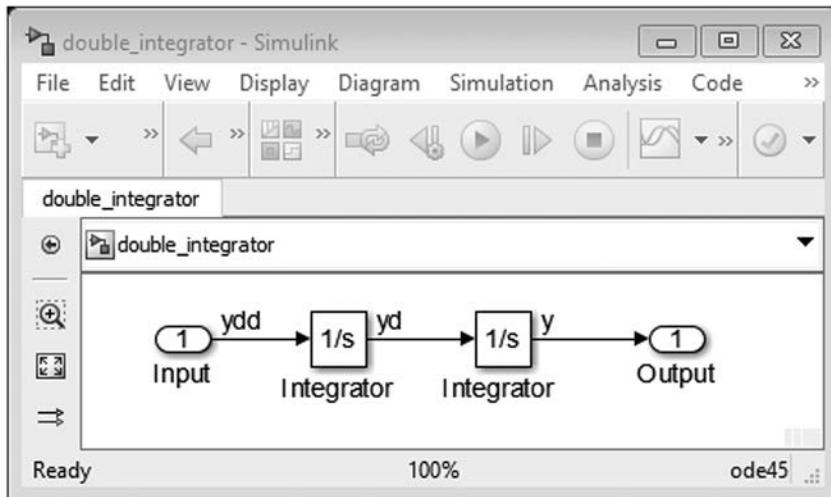


FIGURE 5.2 Integrating the second derivative “ydd” twice to obtain the output “y”.

The system parameters K, ω_n, ζ and the literal constant “2” are generated using a “Constant” block found in the “Sources” sublibrary. The “Math” sublibrary provides the additional blocks for addition and multiplication of the signals.

We have yet to specify an input or forcing function, assuming there is one. For now, let’s pick a simple step input applied at $t = 0$. Looking in the “Sources” sublibrary, the step input can be implemented with a “Constant” or “Step” block, however the latter is more flexible should we later decide to delay the time at which the step is applied.

Numerical values of the system parameters are set by selecting the individual blocks and typing in the appropriate values in a properties dialog box. Some Simulink blocks contain several parameters, all of which should be specified or else the default values will be used. For example, the “Step” block generally requires values for “Step time,” “Initial value,” and “Final value” as shown in [Figure 5.3](#) and the “Integrator” block requires an “Initial condition”.

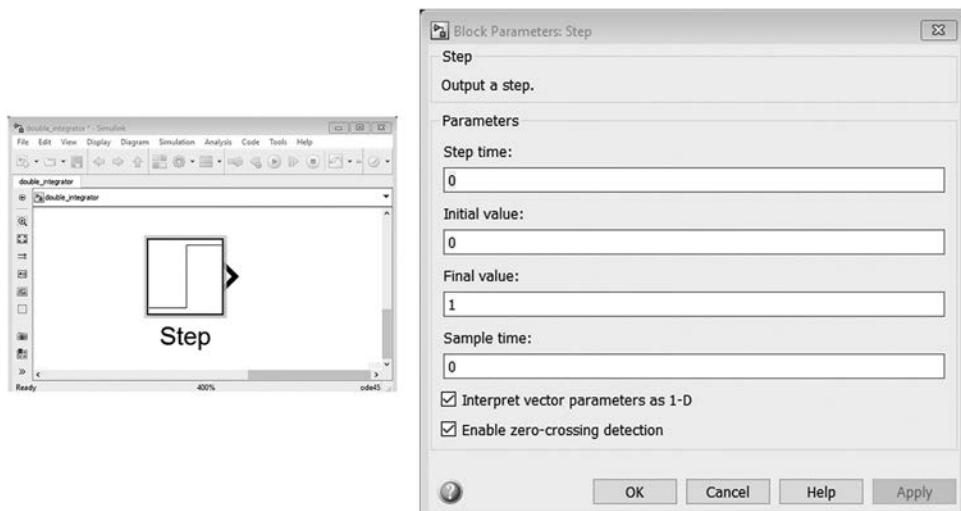


FIGURE 5.3 Dialog box for specifying input step parameter values.

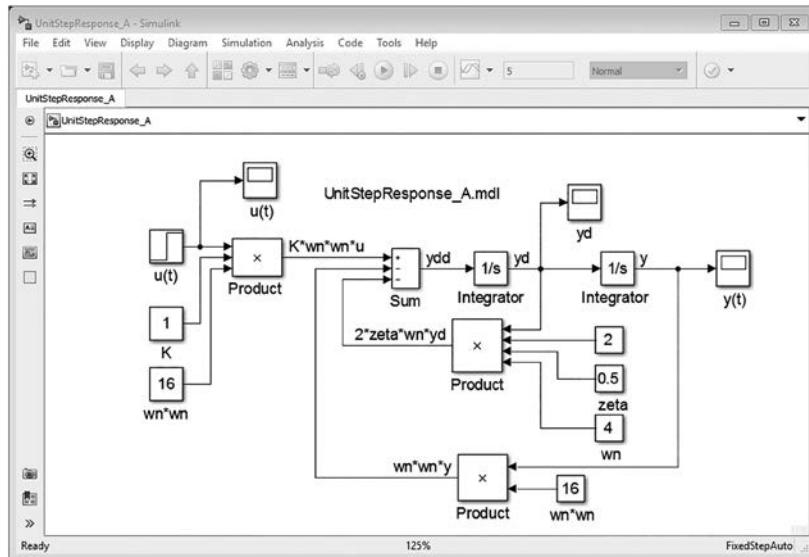


FIGURE 5.4 Simulink model for step response of a second-order system.

Figure 5.4 shows a Simulink diagram for simulation of the unit step response of the second-order system. The choice of Simulink blocks and their location in a Simulink diagram is not unique. The appearance or layout of blocks depends to a large extent on individual user preferences. Some prefer the diagram be the most economical in terms of Simulink blocks used. Others are more concerned with layout style striving to make the diagram visually appealing.

Often times, the mathematical model of the system is available in block diagram form, as in the case of a control system. A Simulink diagram of the system will be strikingly similar, especially when Simulink blocks for modeling actual system components are available.

An alternate Simulink diagram for the second order-system in Equation 5.1 is shown in Figure 5.5. A “Gain” block with parameter value equal to the product $2\zeta\omega_n$ replaces the “Product” block in

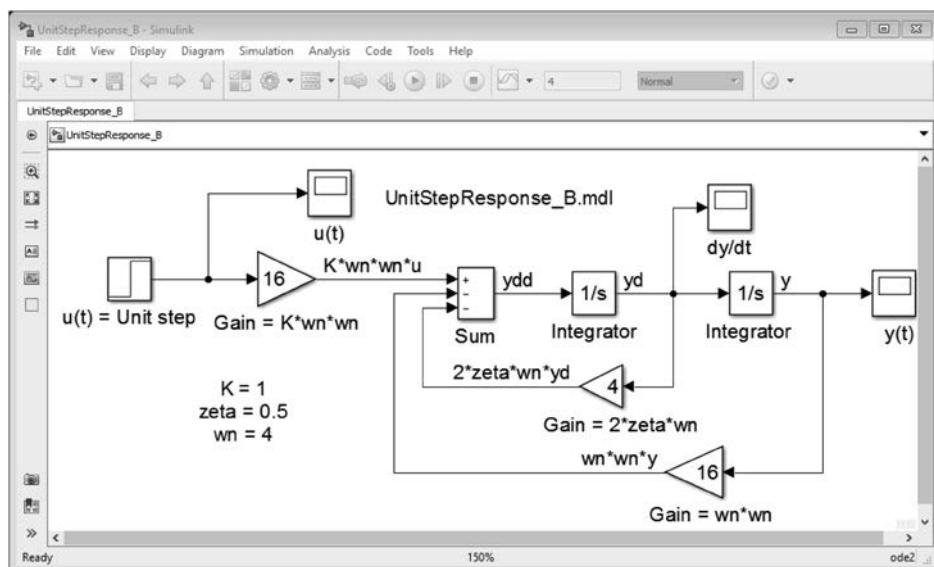


FIGURE 5.5 Alternate Simulink model for a second-order system step response.

the inner feedback loop and the three constant blocks feeding it. Another “Gain” block is inserted in the outer feedback loop with parameter value numerically equal to ω_n^2 replacing the “Product” and “Constant” blocks in [Figure 5.4](#). A third “Gain” block is employed to multiply the input $u(t)$ by $K\omega_n^2$, further reducing the number of blocks required.

Note the similarity between the Simulink diagram in [Figure 5.5](#) and the simulation diagram of the system in [Figure 2.13](#). In fact, the thought process for preparing a simulation diagram of a system is nearly identical to the steps required to arrive at a Simulink diagram.

Before we delve further into the Simulink library, let’s run one of the Simulink models for simulating the step response of the second-order system.

5.2.2 RUNNING A SIMULINK MODEL

The Simulink model is similar to a conventional block diagram of a system. For a system with analog components, it embodies the algebraic and differential equations of the continuous-time math model. For inherently discrete-time systems, the Simulink model encapsulates algebraic and difference equations governing the system’s behavior. Simulink models of hybrid systems containing analog and discrete-time components implement solutions to algebraic, differential and difference equations.

A computer program is created from the Simulink model to solve the equations that comprise the mathematical model of the system. Some of its functions include initialization of state variables, calculation of state derivatives, solution of algebraic equations, updating the state variables and calculation of the system’s outputs. Simulink offers a variety of numerical integrators to advance the continuous-time state vector over an integration step. The user has the option of choosing a particular integrator and step size (applicable for fixed-step size algorithms), tolerances for satisfying accuracy requirements, the simulation start and stop times, and exchanging simulation data with MATLAB via The MATLAB Workspace.

Clicking on the “Model Configuration Parameters” icon located directly under “Simulation” in the Simulink tool bar leads to a dialog box like the one shown in [Figure 5.6](#) where the simulation is configured according to the user’s preferences as previously described. The improved Euler integrator (Heun’s method) with a fixed step size of 0.01 s and simulation time of 5 s has been selected.

After configuring the simulation, hitting the “Start” icon on the Simulink tool bar begins the simulation. The simulation terminates when the simulation time reaches the selected “stop time” of 5 s.

The simplest way to view simulation output is to select one of the scopes, double click it, and observe the time history of its input. The output of the second integrator “y” is displayed in [Figure 5.7](#).

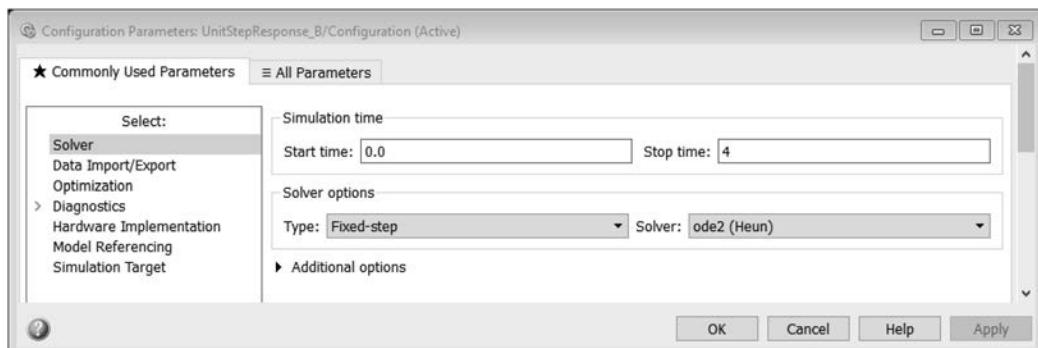


FIGURE 5.6 Dialog box for configuring simulation.

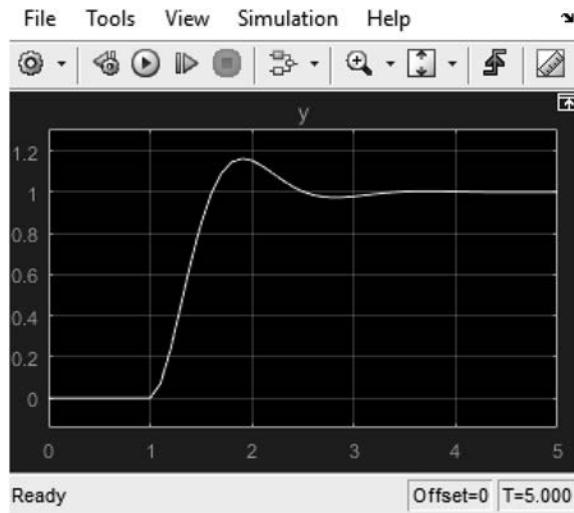


FIGURE 5.7 Screen capture of ‘ $y(t)$ ’ scope output.

A scope output graph can be edited in Simulink by hitting the drop down arrow located under “File” at the top of [Figure 5.7](#). Properties related to the overall appearance of the figure such as colors, axis scales and limits, line thickness, labels, etc. are under the control of the scope property editor. [Figure 5.8](#) shows the result of editing the scope output in [Figure 5.7](#).

The simulation results can also be imported to the MATLAB Workspace several different ways. For simulation data displayed in scopes, the process consists of hitting the “Configuration Properties” icon directly under “File” in the Simulink scope menu bar. A “Configuration Properties” dialog box opens and the “Logging” option at the top is selected. A second dialog box opens and the variable (or variables) to be saved are named. Additional properties of the data are specified as shown in [Figure 5.9](#).

Once in the MATLAB Workspace, the logged signals can be graphed from within the MATLAB environment. [Figures 5.10](#) through [5.12](#) illustrate the results of data logging “ ydd ,” “ yd ” and “ y ,”

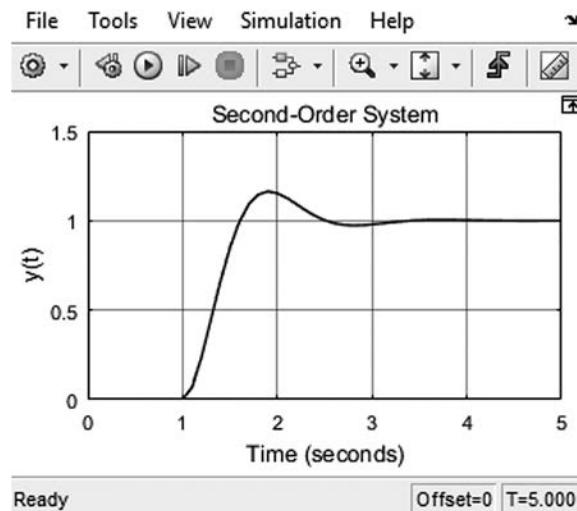


FIGURE 5.8 Screen capture after editing of [Figure 5.7](#).

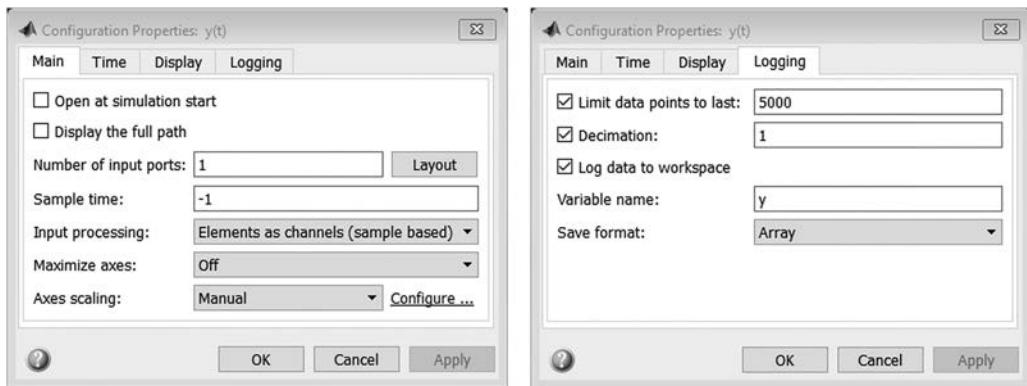


FIGURE 5.9 Dialog boxes for enabling data logging of scope signals.

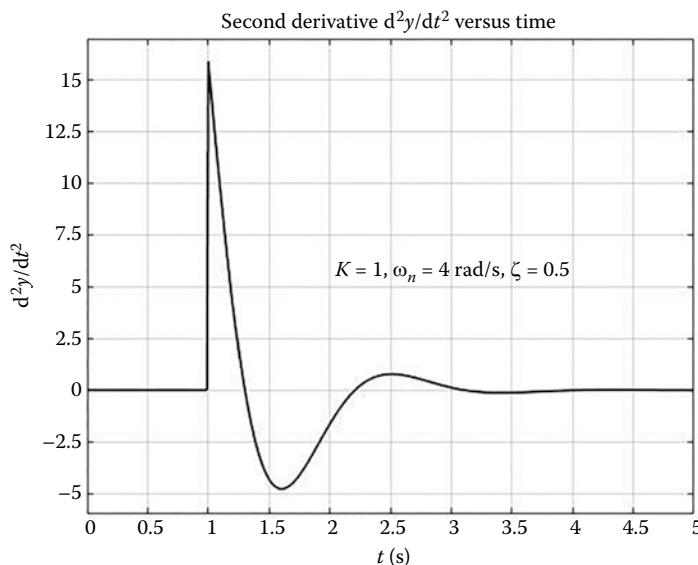


FIGURE 5.10 MATLAB plot of second-order system response dy^2/dt^2 .

respectively, for subsequent plotting in MATLAB. An additional scope was added to the model to capture the second derivative “ydd.”

EXERCISES

- 5.1 Suppose the flow out of Tank 1 enters Tank 2 at the bottom instead of the top.
 - a. Amend the cascaded tank model equations to account for the interaction between the tanks.
 - b. Run the simulation for the baseline conditions given in the text and prepare similar graphs to those in [Figures 5.129–5.132](#).
 - c. Run the simulation for the parameter values in Cases I–V given in the text for
 - i. $t_{final} = 30 \text{ min}$
 - ii. t_{final} sufficient for steady-state to be achieved and prepare a tables similar to [Table 5.5](#).

- 5.2 For the case of non-interacting tanks (outflow from Tank 1 flows in to top of Tank 2), determine the constant flowrate “A” from the external source which results in Tank 1 filling up without overflowing. Use baseline conditions for all parameters other than “A”.
- 5.3 Modify the simulation to allow the value of K_2 to vary continuously between zero and 1, i.e. $f_{1,2} = K_2 f_{0,2}$, $0 \leq K_2 \leq 1$
 Run baseline simulations for 3 values of K_2 , between zero and 1, until steady-state is achieved.
- Prepare plots similar to [Figures 5.129–5.132](#).
 - Prepare a table similar to [Table 5.5](#).

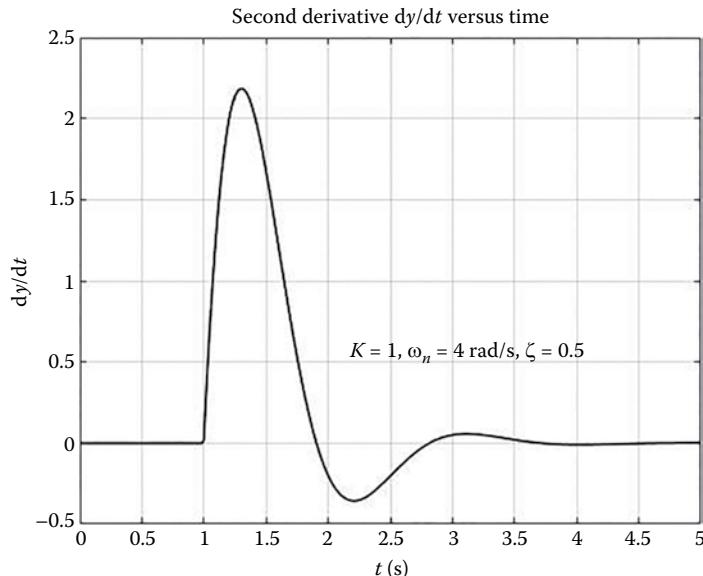


FIGURE 5.11 Parameter dialog box for saving output to the MATLAB Workspace.

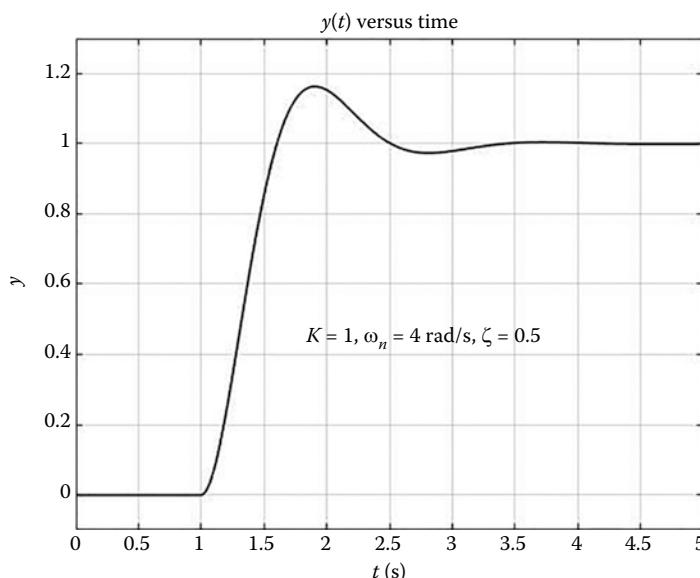


FIGURE 5.12 MATLAB plot of a second-order system unit step response.

5.3 SIMULATION OF LINEAR SYSTEMS

Simulink offers the user a variety of approaches when it comes to simulation of linear continuous-time systems. The form of the system model generally dictates the choice of blocks from the “Continuous” sublibrary to be used in the Simulink model. For example, a linear second-order system comprising two first-order systems in series like that shown in Figure 5.13 suggests an overall Simulink model constructed using Simulink models of the individual first-order systems.

The Simulink diagram of the system is shown in Figure 5.14. Note that the two integrators are not in series like they were when the system model was a second-order differential equation. The state variables are x and y .

A Simulink model of the cascaded first-order systems employing consecutive “Integrator” blocks is easily obtained once the variable x is eliminated from the coupled first-order differential equations in Figure 5.13. The resulting second-order differential equation in y and the Simulink diagram is left as an exercise.

5.3.1 TRANSFER FCN BLOCK

A glimpse of the Simulink blocks in the “Continuous” sublibrary reveals additional options for simulation of linear system models. The “Transfer Fcn” and “Zero-Pole” blocks provide alternative representations for the dynamics of a linear continuous-time component. The n individual integrators and arithmetic blocks for a system component with n th-order dynamics are collapsed into a single block, incorporating the higher-order dynamics. The “Transfer Fcn” and “Zero-Pole” blocks correspond to transfer function models in polynomial and factored form, respectively.

To illustrate the use of the “Transfer Fcn” block, consider a variation of the case study in Section 2.8 for the submarine depth control system. The reference signal $v_{\text{com}}(t)$ for the control loop is the command depth rate and the controlled variable is the actual depth rate $v(t)$ as shown in Figure 5.15. The depth $y(t)$ is obtained by integrating the depth rate $v(t)$.

The submarine is assumed initially to be in steady state at the surface when the command depth rate is suddenly increased to 25 ft/s and held constant for 30 s and then returned to zero. The transfer function for the controller and stern plane actuator is

$$G_C(s) = \frac{\theta(s)}{E(s)} = \frac{K_C s + K_I}{s} \quad (K_C = 0.6, K_I = 0.1) \quad (5.2)$$

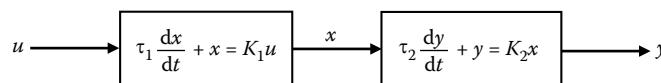


FIGURE 5.13 A second-order system comprised of two cascaded first-order systems.

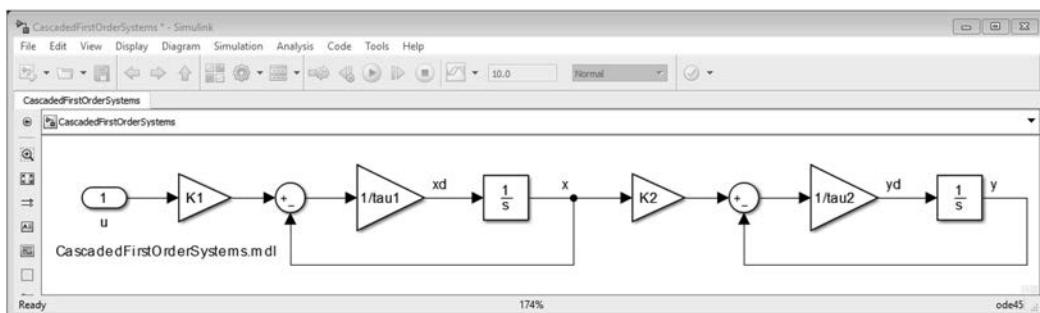


FIGURE 5.14 Simulink diagram of a second-order system shown in Figure 5.13.

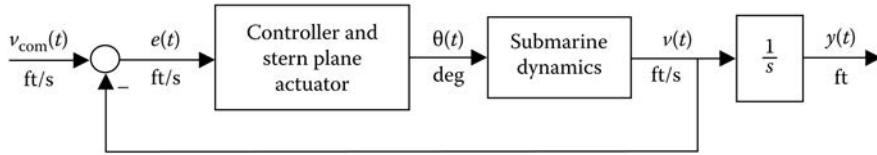


FIGURE 5.15 Submarine depth rate control system.

and the submarine dynamics is modeled by the transfer function

$$G_P(s) = \frac{V(s)}{\theta(s)} = \frac{K_{\dot{\theta}}s + K_{\theta}}{\tau s + 1} \quad (K_{\dot{\theta}} = 20, K_{\theta} = 10, \tau = 10) \quad (5.3)$$

The Simulink diagram is shown in [Figure 5.16](#). A “Transfer Fcn” block was used to model the controller and submarine dynamics.

Note the use of two step blocks with the same amplitude (25 ft/s), the first commencing at $t = 0$ and the second starting at $t = 30$ s along with the summation block to implement the overall command depth rate signal. The command and actual depth rates are multiplexed and fed to the scope in the upper right corner of the diagram. The submarine depth is captured by the scope directly below. The simulation was configured using Simulink’s fixed-step “ode4” numerical integrator with step size 0.01 s. The “ode4” numerical integrator belongs to a family of numerical integrators collectively referred to as Runga–Kutta. [Chapter 6](#) includes a discussion of Runga–Kutta integration.

The command and actual depth rate signals are shown in [Figure 5.17](#). Note the discontinuity in the actual depth rate at $t = 0$ and $t = 30$ s. This implies the existence of a direct path from the command depth rate v_{com} to the actual depth rate v without integrators present. The direct path is not apparent in [Figure 5.16](#); however, it would be evident on a simulation diagram of the system.

The stern plane angle ($^{\circ}$) and the actual submarine depth (ft) are shown in [Figure 5.18](#).

The presence of a direct path with only algebraic blocks from command input v_{com} to the actual submarine depth rate v is easier to visualize if we express the transfer functions in [Figure 5.16](#) differently, that is,

$$G_C(s) = \frac{\theta(s)}{E(s)} = K_C + \frac{K_I}{s} \quad (5.4)$$

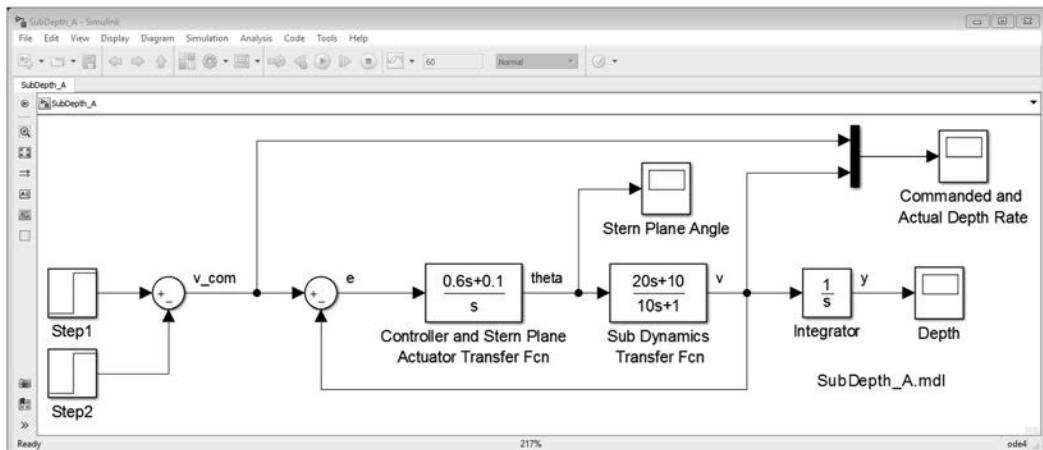


FIGURE 5.16 Simulink diagram for sub depth control using transfer function blocks.

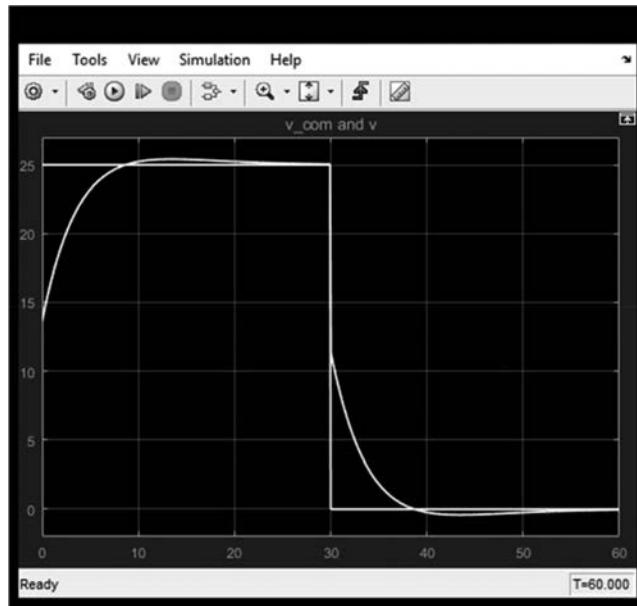


FIGURE 5.17 Command and actual submarine depth rates.

$$G_P(s) = \frac{V(s)}{\theta(s)} = \frac{K_{\dot{\theta}}s + K_{\theta}}{\tau s + 1} \quad (5.5)$$

$$= \frac{K_{\dot{\theta}}}{\tau} + \frac{(K_{\theta} - (K_{\dot{\theta}}/\tau))}{\tau s + 1} \quad (5.6)$$

Hence, the direct path starts from “v – com” through the summer and on through constant blocks with gains “KC” and “Kthd/tau” to the output “v.” The Simulink diagram in Figure 5.16 can be modified to implement the controller and submarine dynamics transfer functions as given in Equations 5.4 and 5.6 (see Exercise 5.5).

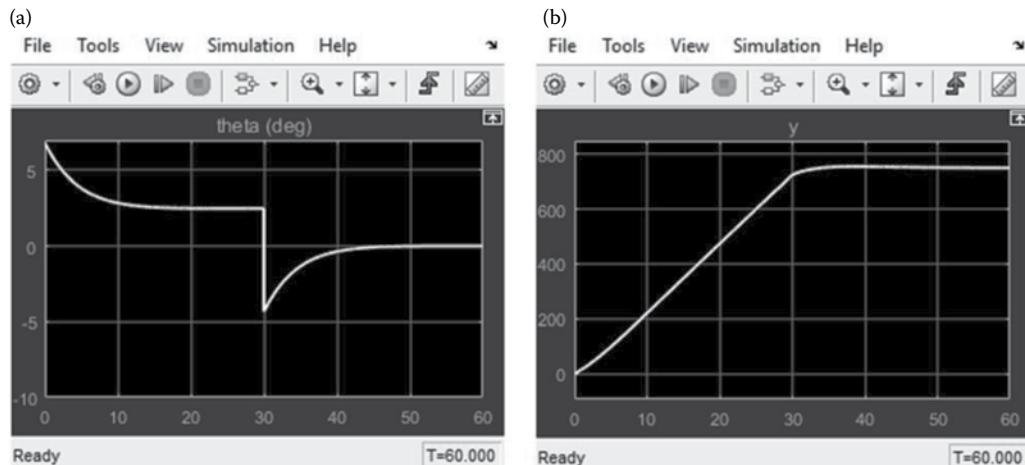


FIGURE 5.18 Simulated (a) stern plane angle (deg) and (b) sub depth (ft).

The submarine depth, shown in [Figure 5.18](#), is continuous at $t = 0$ due to the presence of the integrator between “v” and “y.”

Referring to [Figure 5.15](#), the closed-loop transfer function is

$$\frac{V(s)}{V_{\text{com}(s)}} = \frac{G_C(s)G_P(s)}{1 + G_C(s)G_P(s)} \quad (5.7)$$

$$= \frac{((K_C s + K_I)/s)((K_{\dot{\theta}} s + K_{\theta})/(\tau s + 1))}{1 + ((K_C s + K_I)/s)((K_{\dot{\theta}} s + K_{\theta})/(\tau s + 1))} \quad (5.8)$$

$$= \frac{(K_C s + K_I)(K_{\dot{\theta}} s + K_{\theta})}{s(\tau s + 1) + (K_C s + K_I)(K_{\dot{\theta}} s + K_{\theta})} \quad (5.9)$$

The steady-state value $v(\infty)$ resulting from the step input $v_{\text{com}}(t) = 25, t \geq 0$ is obtained from the final value theorem (Section 4.2),

$$v(\infty) = \lim_{s \rightarrow 0} s V(s) = \lim_{s \rightarrow 0} s \left[\frac{(K_C s + K_I)(K_{\dot{\theta}} s + K_{\theta})}{s(\tau s + 1) + (K_C s + K_I)(K_{\dot{\theta}} s + K_{\theta})} \right] \frac{25}{s} = 25 \quad (5.10)$$

confirmed in [Figure 5.17](#), which shows the depth rate $v(t)$ approaching the commanded 25 ft/s once the transient response has vanished.

The discontinuity in depth rate at $t = 0$ shown in [Figure 5.17](#) can also be verified. According to the initial value theorem,

$$v(0^+) = \lim_{s \rightarrow \infty} s V(s) = \lim_{s \rightarrow \infty} s \left[\frac{(K_C s + K_I)(K_{\dot{\theta}} s + K_{\theta})}{s(\tau s + 1) + (K_C s + K_I)(K_{\dot{\theta}} s + K_{\theta})} \right] \frac{25}{s} \quad (5.11)$$

$$= \lim_{s \rightarrow \infty} \left[\frac{(K_C + (K_I/s))(K_{\dot{\theta}} + (K_{\theta}/s))}{(\tau + (1/s)) + (K_C + (K_I/s))(K_{\dot{\theta}} + (K_{\theta}/s))} \right] 25 \quad (5.12)$$

$$= \frac{25 K_C K_{\dot{\theta}}}{\tau + K_C K_{\dot{\theta}}} = \frac{25(0.6)(20)}{10 + (0.6)(20)} = 13.64 \text{ ft/s} \quad (5.13)$$

is in agreement with the graph of $v(t)$ shown in [Figure 5.17](#).

The following example further illustrates the use of the “Transfer fcn” block.

EXAMPLE 5.1

For the submarine depth rate control system shown in [Figure 5.15](#),

- Find the analytical solution for the submarine depth rate $v(t), 0 < t \leq 30$ in response to the command input $v_{\text{com}}(t) = 25, t \geq 0$.
- Model the closed-loop control system dynamics using a “Transfer fcn” block for $V(s)/V_{\text{com}}(s)$ and use it to simulate the depth rate response to the command depth rate shown in [Figure 5.17](#). Compare the simulated and analytical depth rate responses for $v(t), 0 < t \leq 30$.

a. From Equation 5.9,

$$V(s) = \left[\frac{(K_C s + K_I)(K_0 s + K_\theta)}{s(\tau s + 1) + (K_C s + K_I)(K_0 s + K_\theta)} \right] V_{\text{com}}(s) \quad (5.14)$$

$$= \left[\frac{(K_C K_\theta s^2 + (K_C K_\theta + K_I K_\theta)s + K_I K_\theta)}{(\tau + K_C K_\theta)s^2 + (1 + K_C K_\theta + K_I K_\theta)s + K_I K_\theta} \right] \frac{25}{s} \quad (5.15)$$

$$= \left[\frac{12s^2 + 8s + 1}{22s^2 + 9s + 1} \right] \frac{25}{s} \quad (5.16)$$

$$= \frac{25}{22} \left[\frac{12s^2 + 8s + 1}{s^2 + (9/22)s + 1/22} \right] \quad (5.17)$$

Using partial fraction expansion of the right-hand side of Equation 5.17 followed by inverse Laplace transformation, the solution for $v(t)$, $0 < t \leq 30$ becomes

$$v(t) = 25 - \frac{25}{22} e^{-9t/44} \left[10 \cos\left(\frac{\sqrt{7}}{44}t\right) - \frac{47}{\sqrt{4}} \sin\left(\frac{\sqrt{7}}{44}t\right) \right], \quad 0 < t \leq 30 \quad (5.18)$$

b. The analytical solution for $v(t)$ in Equation 5.18 is incorporated in Simulink using a "Sine wave" block from the "Sources" sublibrary and a "Math Function" block from the "Math" sublibrary for the exponential term. The Simulink diagram appears in [Figure 5.19](#).

The "Sine Wave" parameters dialog box for the cosine term $\cos(\sqrt{7}/44 t)$ in the analytical solution, Equation 5.18, is shown in [Figure 5.20](#). Note that the phase angle is $\pi/2$ rad to produce the cosine function.

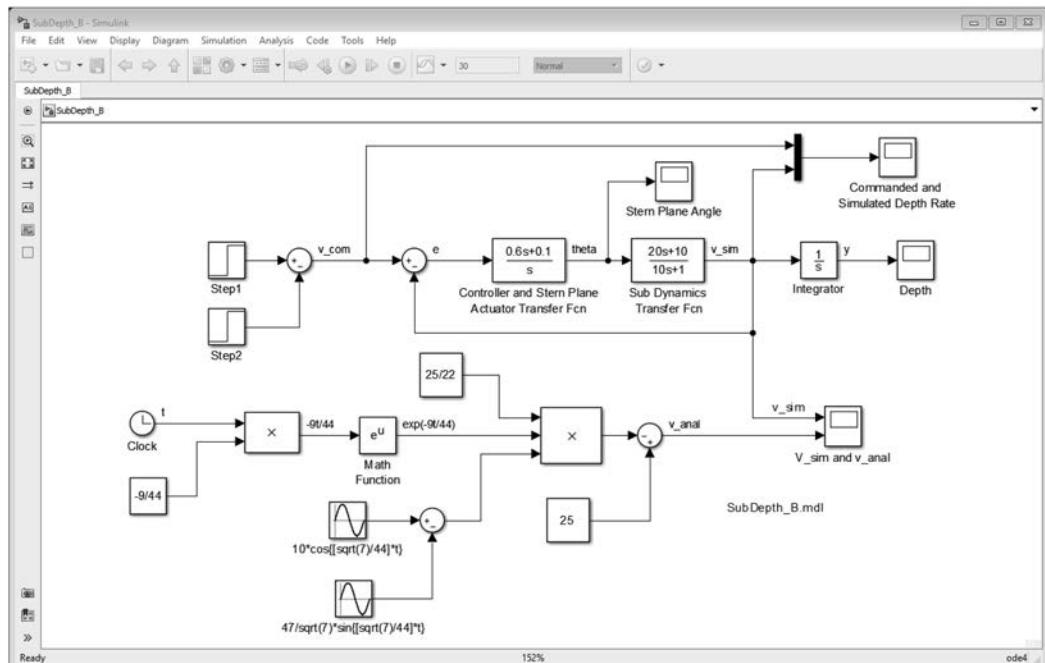


FIGURE 5.19 Simulink diagram with simulated and analytical submarine depth rate.

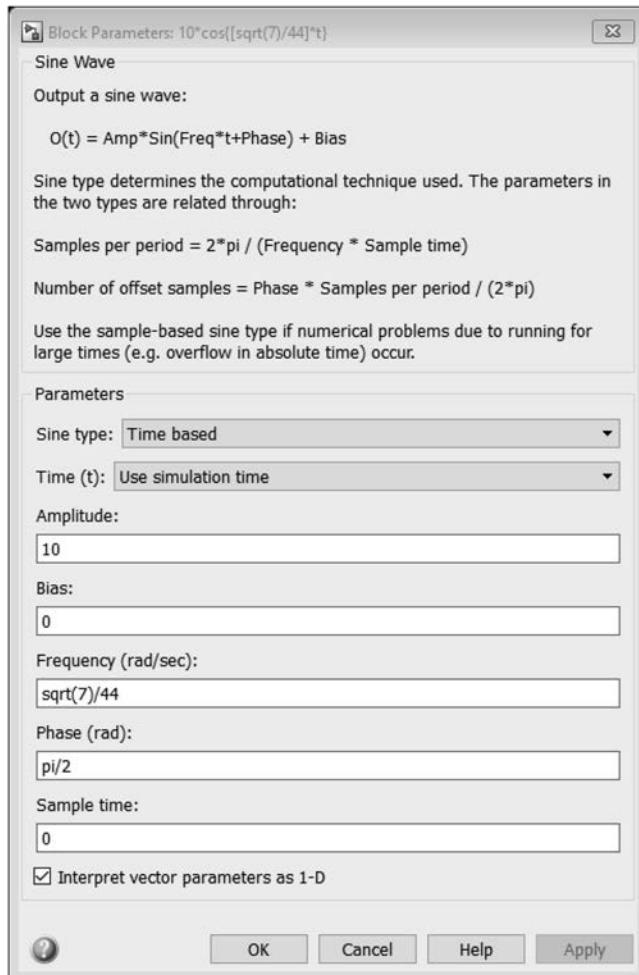


FIGURE 5.20 “Sine Wave” parameter box to generate cosine term in analytical solution.

The control system loop with input $v_{\text{com}}(t)$ and output $v(t)$ in Figure 5.15 is replaced with the equivalent closed-loop transfer function $V(s)/V_{\text{com}}(s)$ in Figure 5.21.

The Simulink diagram shown in Figure 5.22 includes a “Transfer fcn” block for implementing the closed-loop transfer function. The simulated and analytical depth rates for a time period $0 < t \leq 12$ s are shown in Figure 5.23. The graphs were generated in the MATLAB M-file “Ch5_Ex5_1.m” by saving the data in the scope shown with the heavy line multiplexed input in Figure 5.22. The complete set of time values along with the simulated and analytical results is saved in the MATLAB Workspace in a named array set in the scope dialog box. Also shown is the difference between the two depth rates. It is clear from looking at the difference that the simulated depth rate is nearly identical to the analytical solution.

$$V_{\text{com}}(s) \xrightarrow{\frac{V(s)}{V_{\text{com}}(s)} = \frac{(K_C s + K_I)(K_\theta s + K_\theta)}{s(\tau s + 1) + (K_C s + K_I)(K_\theta s + K_\theta)}} V(s)$$

FIGURE 5.21 Closed-loop transfer function of submarine depth rate control system.

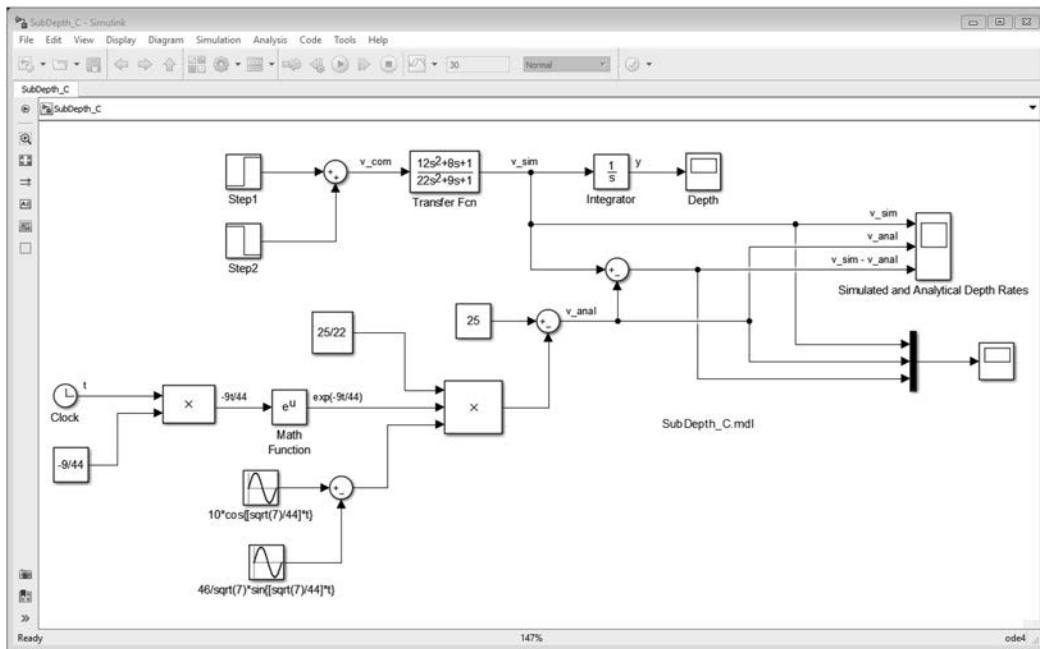


FIGURE 5.22 Simulink diagram using “Transfer fcn” block for submarine closed-loop depth rate control system dynamics.

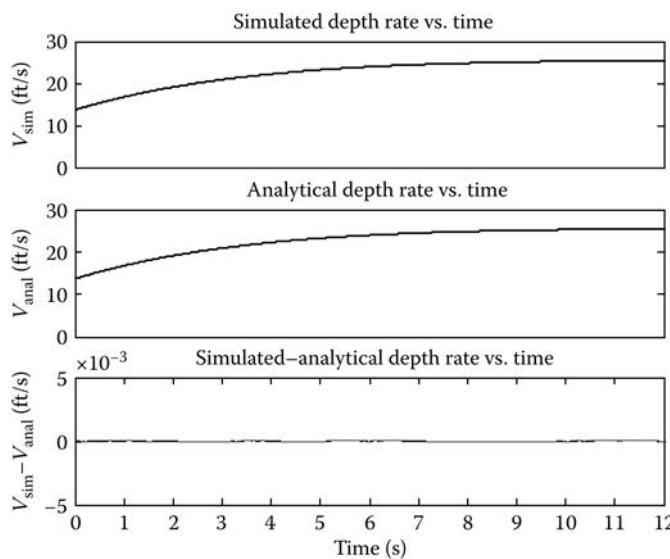


FIGURE 5.23 Analytical and simulated depth rate using “Transfer fcn” for $V(s)/V_{\text{com}}(s)$.

5.3.2 STATE-SPACE BLOCK

The process of transforming models consisting of linear algebraic and differential equations into state variable form was demonstrated in Section 2.6. Conversion of SISO (single input–single output) or MIMO (multiple input–multiple output) system transfer functions to state-space models and vice versa was illustrated using the control system toolbox in Section 4.10. Simulink provides



FIGURE 5.24 The Simulink “State-Space” block.

a mechanism for incorporating state variable models of system components using the “State-Space” block located in the “Continuous” sublibrary. A partial description of the “State-Space” block is shown in Figure 5.24. The next example illustrates its use.

EXAMPLE 5.2

An automobile traveling along a level road at a constant speed v_0 encounters a speed bump shown in Figure 5.25. The vehicle’s suspension system (front and rear springs and shock absorbers) is modeled by linear springs and dampers, and the compliance of the tires is modeled by front and rear springs. The vehicle cab motion is limited to heave in the y -direction and a small amount of pitch θ of the vehicle’s longitudinal axis. The tires are assumed to remain in contact with the road surface at all times.

The road profile is responsible for the system’s input $u = [u_f \ u_r]^T$, where u_f and u_r are the height of the road (with respect to some reference) underneath the front and rear tires, respectively. The system has three translational degrees of freedom, y , y_f , y_r , which are the vertical displacements of the vehicle cab and both front and rear axles from their equilibrium positions. The lone rotational degree of freedom is the pitch angle θ .

Three of the four model equations are obtained by equating the sum of suspension and tire forces acting on the three masses to the appropriate acceleration term, $M\ddot{y}$, $M_f\ddot{y}_f$, and $M_r\ddot{y}_r$. The fourth equation sets the torques about the vehicle cab’s center of gravity created by the suspension forces equal to the inertial acceleration $I\ddot{\theta}$.

The model equations are listed as follows:

$$M\ddot{y} = K_{fs}[y_f - (y + L_f\theta)] + B_f[\dot{y}_f - (\dot{y} + L_f\dot{\theta})] + K_{rs}[y_r - (y - L_r\theta)] + B_r[\dot{y}_r - (\dot{y} - L_r\dot{\theta})] \quad (5.19)$$

$$= -(K_{fs} + K_{rs})y - (B_f + B_r)\dot{y} + K_{fs}y_f + B_f\dot{y}_f + K_{rs}y_r + B_r\dot{y}_r + (K_{rs}L_r - K_{fs}L_f)\theta + (B_rL_r - B_fL_f)\dot{\theta} \quad (5.20)$$

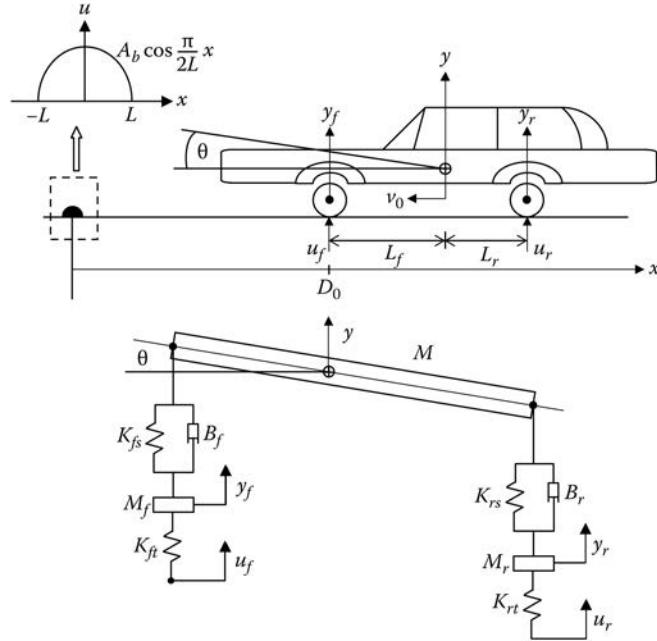


FIGURE 5.25 Moving vehicle and suspension system model.

$$M_f \ddot{y}_f = -K_{fs}[y_f - (y + L_f \theta)] + B_f[\dot{y}_f - (\dot{y} + L_f \dot{\theta})] + K_{ft}(u_f - y_f) \quad (5.21)$$

$$= -(K_{fs} + K_{ft})y_f - B_f \dot{y}_f + K_{fs}y + B_f \dot{y} + K_{fs}L_f \theta + B_f L_f \dot{\theta} + K_{ft}u_f \quad (5.22)$$

$$M_r \ddot{y}_r = -K_{rs}[y_r - (y + L_r \theta)] + B_r[\dot{y}_r - (\dot{y} + L_r \dot{\theta})] + K_{rt}(u_r - y_r) \quad (5.23)$$

$$= -(K_{rs} + K_{rt})y_r - B_r \dot{y}_r + K_{rs}y + B_r \dot{y} + K_{rs}L_r \theta + B_r L_r \dot{\theta} + K_{rt}u_r \quad (5.24)$$

$$I\ddot{\theta} = \{K_{fs}[y_f - (y + L_f \theta)] + B_f[\dot{y}_f - (\dot{y} + L_f \dot{\theta})]\}L_f$$

$$- \{K_{rs}[y_r - (y + L_r \theta)] + B_r[\dot{y}_r - (\dot{y} + L_r \dot{\theta})]\}L_r \quad (5.25)$$

$$\begin{aligned} &= -\{K_{fs}L_f^2 + K_{rs}L_r^2\}\theta - (B_f L_f^2 + B_r L_r^2)\dot{\theta} + (K_{rs}L_r + K_{fs}L_f)y \\ &\quad + (B_r L_r - B_f L_f)\dot{y} + K_{fs}L_f y_f - K_{rs}L_r y_r + B_f L_f \dot{y}_f - B_r L_r \dot{y}_r \end{aligned} \quad (5.26)$$

Note that the equations are linear as a result of assuming small pitch angles, allowing the approximations $\sin \theta \approx \theta$ and $\cos \theta \approx 1$.

a. Introduce state variables

$$x_1 = y, x_3 = y_f, x_5 = y_r, x_7 = \theta,$$

$$x_2 = \dot{y}, x_4 = \dot{y}_f, x_6 = \dot{y}_r, x_8 = \dot{\theta},$$

and solve for the state derivatives, that is, find the matrices A and B in $\dot{x} = Ax + Bu$.

- b. Define the outputs as $y_1 = y$, $y_2 = y_f$, $y_3 = y_r$, and $y_4 = \theta$ and find matrices C and D in $y = Cx + Du$.
- c. Simulate and plot the vehicle dynamics using the following values for the weight of the vehicle and tires, suspension parameters, forward speed, and speed bump profile.

$$\begin{aligned} W &= 4200 \text{ lb}, W_f = 125 \text{ lb}, W_r = 125 \text{ lb}, K_{fs} = 120 \text{ lb/in}, K_{rs} = 180 \text{ lb/in}, \\ B_f &= 25 \text{ lb s/in}, B_r = 35 \text{ lb s/in}, K_{ft} = 1100 \text{ lb/in}, K_{rt} = 1100 \text{ lb/in}, \\ I &= 40,000 \text{ in.lb s}^2, L_f = 55 \text{ in.}, L_r = 65 \text{ in.}, v_0 = 20 \text{ mph}, A_b = 4 \text{ in.}, L = 1 \text{ ft} \end{aligned}$$

- a. Using the definition of the state variables and solving for the state derivatives in Equations 5.19 through 5.26 give

$$\dot{x}_1 = x_2 \quad (5.27)$$

$$\begin{aligned} \dot{x}_2 &= \frac{-(K_{fs} + K_{rs})}{M} x_1 - \frac{(B_f + B_r)}{M} x_2 + \frac{K_{fs}}{M} x_3 + \frac{B_f}{M} x_4 + \frac{K_{rs}}{M} x_5 + \frac{B_r}{M} x_6 \\ &\quad + \frac{(K_{rs}L_r + K_{fs}L_f)}{M} x_7 + \frac{(B_rL_r + B_fL_f)}{M} x_8 \end{aligned} \quad (5.28)$$

$$\dot{x}_3 = x_4 \quad (5.29)$$

$$\dot{x}_4 = \frac{K_{fs}}{M_f} x_1 + \frac{B_f}{M_f} x_2 - \frac{(K_{fs} + K_{ft})}{M_f} x_3 - \frac{B_f}{M_f} x_4 + \frac{K_{fs}L_f}{M_f} x_7 + \frac{B_fL_f}{M_f} x_8 + \frac{K_{ft}}{M_f} u_f \quad (5.30)$$

$$\dot{x}_5 = x_6 \quad (5.31)$$

$$\dot{x}_6 = \frac{K_{rs}}{M_r} x_1 + \frac{B_r}{M_r} x_2 - \frac{(K_{rs} + K_{rt})}{M_r} x_5 - \frac{B_r}{M_r} x_6 - \frac{K_{rs}L_r}{M_r} x_7 - \frac{B_rL_r}{M_r} x_8 + \frac{K_{rt}}{M_r} u_r \quad (5.32)$$

$$\dot{x}_7 = \dot{x}_8 \quad (5.33)$$

$$\begin{aligned} \dot{x}_8 &= \frac{(K_{rs}L_r - K_{fs}L_f)}{I} x_1 + \frac{(B_rL_r - B_fL_f)}{I} x_2 + \frac{K_{fs}L_f}{I} x_3 + \frac{B_fL_f}{I} x_4 \\ &\quad - \frac{K_{rs}L_r}{I} x_5 - \frac{B_rL_r}{I} x_6 - \frac{(K_{fs}L_f^2 - K_{rs}L_r^2)}{I} x_7 - \frac{(B_fL_f^2 + B_rL_r^2)}{I} x_8 \end{aligned} \quad (5.34)$$

The system matrix A and input matrix B are

$$A = \left[\begin{array}{ccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-(K_{fs} + K_{rs})}{M} & \frac{-(B_f + B_r)}{M} & \frac{K_{fs}}{M} & \frac{B_f}{M} & \frac{K_{rs}}{M} & \frac{B_r}{M} & \frac{K_{rs}L_r - K_{fs}L_f}{M} & \frac{B_fL_r - B_fL_f}{M} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{K_{fs}}{M_f} & \frac{B_f}{M_f} & \frac{-(K_{fs} + K_{ft})}{M_f} & \frac{-B_f}{M_f} & 0 & 0 & \frac{K_{fs}L_f}{M_f} & \frac{B_fL_f}{M_f} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{K_{rs}}{M_r} & \frac{B_r}{M_r} & 0 & 0 & \frac{-(K_{rs} + K_{rt})}{M_r} & \frac{-B_r}{M_r} & \frac{-K_{rs}L_r}{M_r} & \frac{-B_rL_r}{M_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{(K_{rs}L_r - K_{fs}L_f)}{I} & \frac{(B_rL_r - B_fL_f)}{I} & \frac{K_{fs}L_f}{I} & \frac{B_fL_f}{I} & \frac{-K_{rs}L_r}{I} & \frac{-B_rL_r}{I} & \frac{-(K_{fs}L_f^2 + K_{rs}L_r^2)}{I} & \frac{-(B_fL_f^2 + B_rL_r^2)}{I} \end{array} \right] \quad (5.35)$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{K_{ft}}{M_f} & 0 \\ 0 & 0 \\ 0 & \frac{K_{fr}}{M_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.36)$$

b. The output matrix C and direct transmission matrix D are given by

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.37)$$

The direct transmission matrix D is all zeros, since the system inputs u_f and u_r are not directly coupled to the outputs, that is, step changes in either input are integrated before influencing the outputs, and, hence, the outputs are continuous at the time the step input(s) is applied.

c. The Simulink diagram for simulating the vehicle's response as it travels over the speed bump is shown in Figure 5.26. The "state-space" block parameters are the matrices

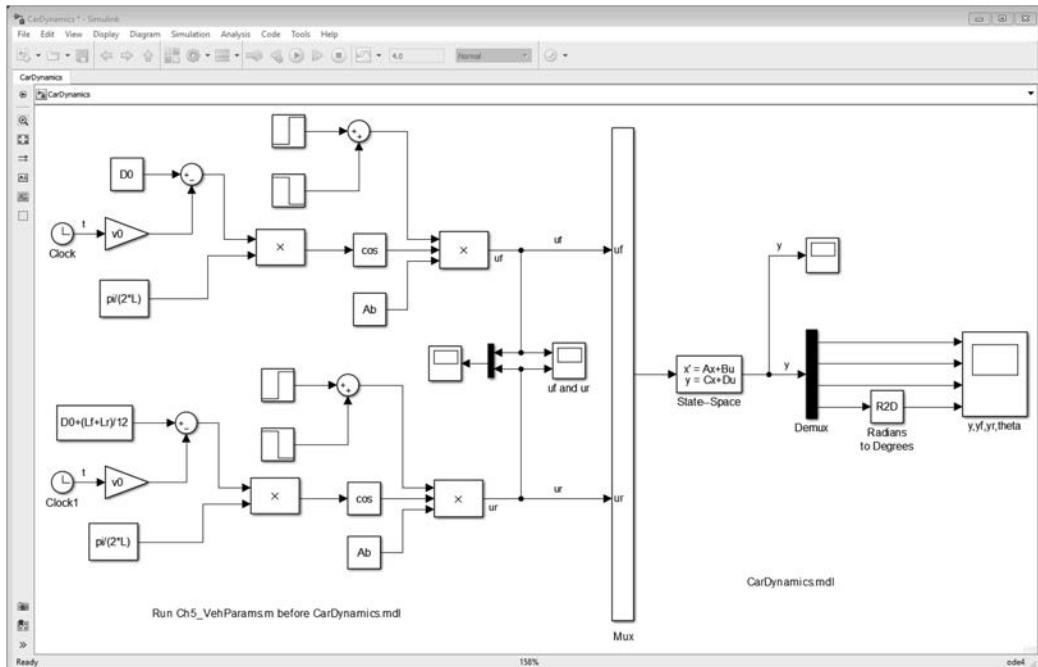


FIGURE 5.26 Simulink diagram for vehicle response traveling over a speed bump.

A , B , C , and D of Equations 5.35 through 5.37, which have been defined in a MATLAB M-file “*Ch5_VehParams.m*” for convenience.

The input displacements u_f and u_r are based on the speed bump profile shown in Figure 5.25 and the forward speed of the car. The front tire displacement is given by

$$u_f = \begin{cases} 0, & t < \frac{D_0 - L}{v_0} \\ A_b \cos \frac{\pi}{2L} (D_0 - v_0 t), & \frac{D_0 - L}{v_0} \leq t \leq \frac{D_0 + L}{v_0} \\ 0, & t > \frac{D_0 + L}{v_0} \end{cases} \quad (5.38)$$

The Simulink blocks to implement u_f (and u_r) are shown in the top left (and lower left) corner of Figure 5.26. Note the use of the “Clock” from the “sources” sublibrary to generate the simulation time variable “ t .” Also, the wider (and heavier) arrows in and out of the “state-Space” block designate the presence of nonscalar signals, and the “2” and “4” indicate the number of components in each.

The inputs u_f and u_r are captured in a scope and plotted for $1.75 < t \leq 2.5$ s in the M-file “*Ch5_Ex5_2.m*” (see Figure 5.27).

The output vector “ y ” of the “state-space” block is decomposed in a “Demux” block and sent to a scope with four input channels (one for each output). It is also saved for use by the M-file “*Ch5_Ex5_2.m*.” The results are plotted for the interval $1.5 \leq t \leq 3.5$ s in Figure 5.28.

The vehicle cab displacement varies from -0.189 to 0.627 in. despite the 4 in. height of the speed bump. Also, the pitch of the vehicle is constrained to $-0.403^\circ \leq \theta \leq 0.358^\circ$.

The “Data history” tab in the “scope” with multiplexed input containing “ u_f ” and “ u_r ” is shown in Figure 5.29. Simulation time values and front and rear tire displacements are saved to the MATLAB Workspace in array “ uF_ur .”

The following MATLAB statements placed at the beginning of M-file “*Ch5_Ex5_2.m*” store the saved values of the time array and tire displacements in arrays “ t ,” “ uF ,” and “ ur ” and produce the graph shown in Figure 5.27.

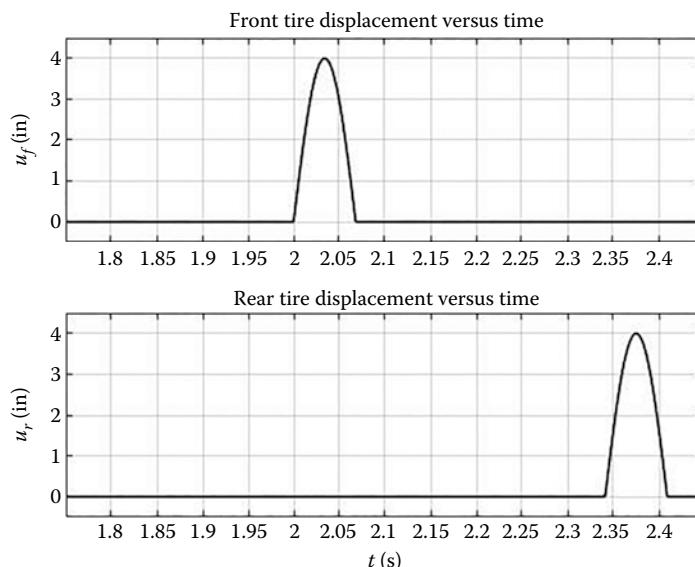


FIGURE 5.27 Inputs u_f and u_r for vehicle traveling at constant speed v_0 .

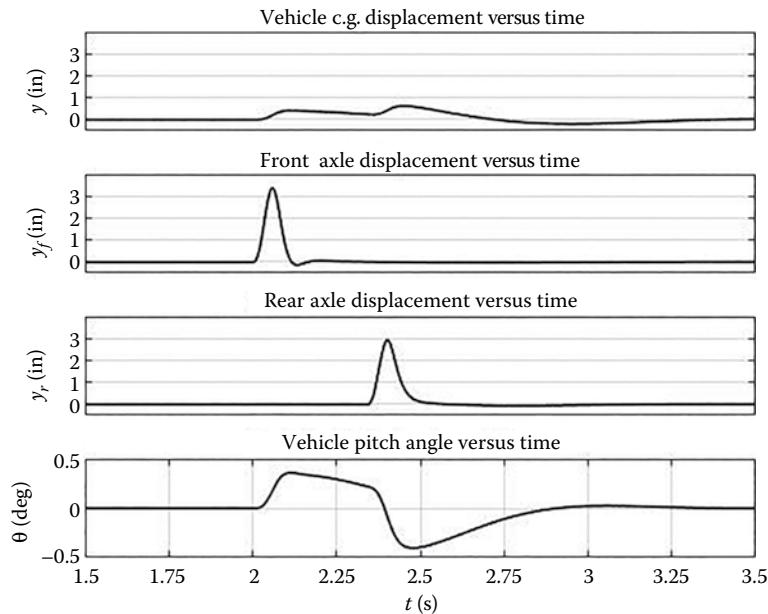


FIGURE 5.28 Outputs y , y_f , y_r , and θ of vehicle suspension system for $1.5 \leq t \leq 4$ s.

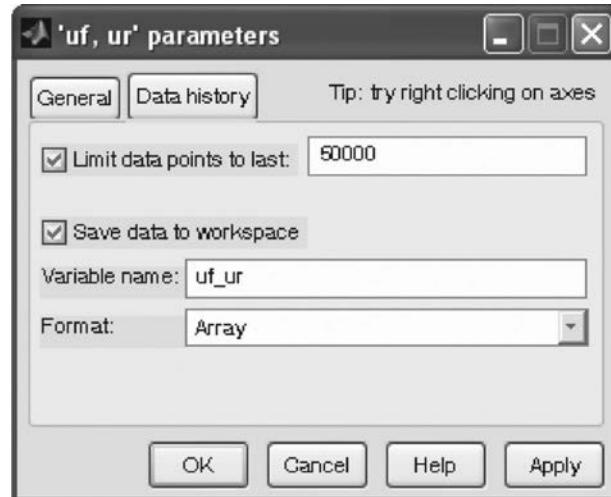


FIGURE 5.29 Saving “uf” and “ur” for plotting in M-file “Ch5_Ex5_2.m.”

```

t=uf_ur(:,1);
uf=uf_ur(:,2);
ur=uf_ur(:,3);
figure(1) % begin Figure 5.27
subplot(2,1,1)
plot(t,uf,'b','LineWidth',1.3)
grid on
set(gca,'XTick',[1.8:0.05:2.4],'YTick',[0:4],'FontSize',11)
ylabel('u_f (in)', 'FontSize',13)
axis([1.75 2.45 -0.5 4.5])
set(gca,'XTick',[1.8:0.05:2.4], 'YTick',[0:4])

```

```

title('Front Tire Displacement vs Time','FontSize',11)
subplot(2,1,2)
plot(t,ur,'b','LineWidth',1.3)
grid on
ylabel('u_r (in)','FontSize',13)
axis([1.75 2.45 -0.5 4.5])
set(gca,'XTick',[1.8:0.05:2.4],'YTick',[0:4],'FontSize',11)
xlabel('t (s)','FontSize',12)
title('Rear Tire Displacement vs Time','FontSize',11))

```

The first statement runs another M-file “Ch5_VehParams.m,” which loads the parameter values. The next command sim(‘CarDynamics’) causes execution of the Simulink model “CarDynamics.mdl.”

EXERCISES

- 5.4 Modify the cascaded tank system by introducing a third tank with area $A_3 = 7.5 \text{ ft}^2$ and sufficiently tall to hold the spillover from both tanks. Run several simulations where spillover from one or both tanks occurs and plot the tank levels $H_1(t)$, $H_2(t)$, and $H_3(t)$ vs. time on the same graph. Stop the simulations when steady-state is achieved. Specify the entire set of parameter values used in each simulation run.
- 5.5 For the submarine depth control system shown in Figure 5.15,
- Draw a simulation diagram. Is there a direct connection from v_{com} to v ?
 - Redraw the Simulink diagram in Figure 5.16 using the alternate expressions for the controller and submarine dynamics transfer functions in Equations 5.4 and 5.6.
 - Run the Simulink model and compare the responses for $v(t)$, $y(t)$, and $\theta(t)$ with those shown in the text.
- 5.6 Two linear tanks are arranged in series as shown in Figure E5.6:

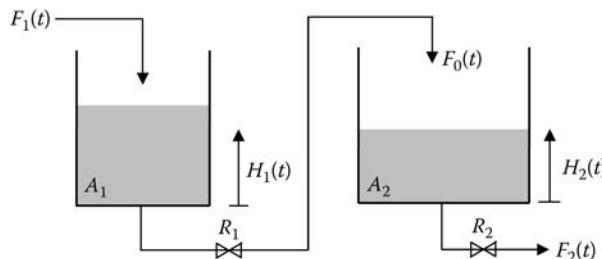


FIGURE E5.6

- Write the differential equation models for the tanks.
- The system parameters are

$$A_1 = 50 \text{ ft}^2, A_2 = 100 \text{ ft}^2, R_1 = 0.2 \text{ ft}/\text{ft}^3/\text{min}, \text{ and } R_2 = 0.3 \text{ ft}/\text{ft}^3/\text{min}$$

Prepare a Simulink diagram of the system, and simulate the response of both tank levels under the following conditions:

- $H_1(0) = 0, H_2(0) = 0, F_1(t) = 40 \text{ ft}^3/\text{min}, t \geq 0$
- $H_1(0) = 0, H_2(0) = 10, F_1(t) = 0, t \geq 0$
- $H_1(0) = 0, H_2(0) = 0, F_1(t) = \begin{cases} 5t, & 0 \leq t \leq 5 \\ -5t + 50, & 5 < t \leq 10 \\ 0, & t > 10 \end{cases}$

- Obtain one graph with time histories of $H_1(t)$ and $H_2(t)$ and a second graph with $F_0(t)$, $F_1(t)$, and $F_2(t)$.
- Eliminate $H_1(t)$ from the two first-order differential equations in part (a) to obtain a second-order differential equation relating $H_2(t)$ and $F_1(t)$.
 - Prepare a Simulink diagram based on the continuous-time model in part (c).
 - Run the Simulink model for the same conditions in part (b), and compare the response for $H_2(t)$ with the one obtained in part (b).
 - Find the analytical solution $[H_2(t)]_{\text{anal}}$ when both tanks are initially empty and $F_1(t) = 40 \text{ ft}^3/\text{min}$, $t \geq 0$. Compare the analytical solution $[H_2(t)]_{\text{anal}}$ with the simulated solution $[H_2(t)]_{\text{sim}}$ obtained in part (b).

Hint: Use Simulink to implement the analytical solution and feed both $[H_2(t)]_{\text{sim}}$ and $[H_2(t)]_{\text{anal}}$ into a summer to obtain the difference.

- Solve Exercise 2.3 using Simulink.
- Solve Exercise 2.4 using Simulink.

5.4 ALGEBRAIC LOOPS

Execution of the Simulink model in this chapter, [Figure 5.16](#), poses a dilemma often encountered when simulating dynamic systems with feedback loops. A runtime warning (default) or error appears in the MATLAB Command Window stating

```
Block diagram 'SubDepth_A' contains 1 algebraic loop(s). To see more
details about the loops use the command
Simulink.BlockDiagram.getAlgebraicLoops('SubDepth_A') or the command
line Simulink debugger by typing sldebug('SubDepth_A') in the MATLAB
command window. To eliminate this message, set the Algebraic loop option
in the Diagnostics page of the Configuration Parameters Dialog to
"None".
Component: Simulink | Category: Block diagram warning
Found algebraic loop containing:
'SubDepth_A/Controller and Stern Plane Actuator Transfer Fcn'
Component: Simulink | Category: Block
'SubDepth_A/Sub Dynamics Transfer Fcn'
Component: Simulink | Category: Block
'SubDepth_A/Sum1' (algebraic variable)
```

An algebraic loop is any closed loop appearing in the Simulink diagram composed of strictly algebraic and implicit blocks such as the implicit discrete-time numerical integrators (discussed in Section 5.6). Consequently, the output of any block in an algebraic loop is ultimately an implicit function of itself. In large scale simulations with several 100 blocks, it is nearly impossible to identify the presence of an algebraic loop by visual inspection. The Simulink diagrams of even relatively simple simulations with only a handful of blocks may contain algebraic loops, which escape detection. Simulink (and other block-oriented continuous simulation languages) detects the presence of an algebraic loop and reports the blocks comprising it.

Before we discuss its implications, let us confirm the existence of an algebraic loop in the Simulink model “*SubDepth_A.mdl*” consisting of the two “Transfer fcn” blocks and the “Sum” block. Referring to [Figure 5.16](#), the controller and stern plane actuator transfer function can be rewritten as follows:

$$G_C(s) = \frac{0.6s + 0.1}{s} = 0.6 + \frac{0.1}{s} \quad (5.39)$$

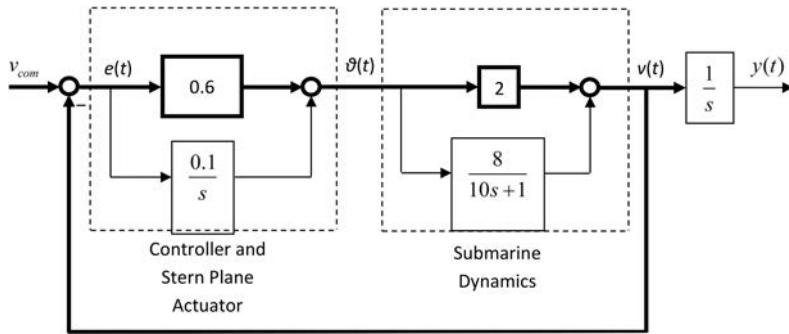


FIGURE 5.30 Block diagram for submarine depth control showing algebraic loop.

and the submarine dynamics transfer function is expressible as

$$G_P(s) = \frac{20s+10}{10s+1} = 2 + \frac{8}{10s+1} \quad (5.40)$$

leading to an equivalent block diagram shown in Figure 5.30.

The algebraic loop is shown in bold, and a similar algebraic loop is present in the Simulink diagram for “*SubDepth_A.mdl*.” Note that if the controller and stern plane actuator transfer function were replaced by a pure gain, the diagram would still have an algebraic loop due to the direct path from the input to the output in the submarine dynamics transfer function.

The dilemma posed by algebraic loops can be demonstrated by looking at the equations the Simulink program is attempting to solve in the submarine example at the time $t = 0$. After initializing the state $\theta(0)$ and $v(0)$ and evaluating the input $v_{com}(0)$, Simulink calculates $e(0)$ according to

$$e(0) = v_{com}(0) - v(0) \quad (5.41)$$

Existence of direct paths, that is, pure gain (zero-order dynamics), from $e(t)$ to $\theta(t)$ and $\theta(t)$ to $v(t)$ implies

$$v(0) = 2\theta(0) \quad (5.42)$$

$$\theta(0) = 0.6e(0) \quad (5.43)$$

Substituting $\theta(0)$ in Equation 5.43 into Equation 5.42 gives

$$v(0) = 2[0.6e(0)] = 1.2e(0) \quad (5.44)$$

Replacing $e(0)$ in Equation 5.44 with $e(0)$ in Equation 5.41 results in

$$v(0) = 1.2[v_{com}(0) - v(0)] \quad (5.45)$$

The circular nature of algebraic loops is demonstrated by Equation 5.45, an implicit equation with $v(0)$ on both sides. In the general case, the implicit equation is nonlinear. Simulink attempts to solve the implicit equations associated with an algebraic loop using the iterative Newton–Raphson method (Chapra and Canale 2002). Solving implicit equation (s) at each iteration, especially nonlinear ones, can dramatically decrease the simulation execution speed. Further, the method can fail to converge to a solution.

The initial depth rate value, more precisely, the value at $t = 0^+$ in the submarine example, is easily verified from Equation 5.45.

$$v(0^+) = 1.2[v_{\text{com}}(0) - v(0^+)] \quad (5.46)$$

$$\Rightarrow v(0^+) = \frac{1.2}{2.2} v_{\text{com}}(0) = \frac{1.2}{2.2} (25) = 13.64 \quad (5.47)$$

in agreement with the value given in Equation 5.13 as well as the graph for $v(t)$ shown in Figure 5.17.

5.4.1 ELIMINATING ALGEBRAIC LOOPS

The most desirable method for eliminating an algebraic loop is by means of algebraic manipulation of the loop equations to produce an equivalent system explicit in nature. It is up to the user to obtain an explicit solution, if one exists, and modify the Simulink diagram accordingly. Simulink does not perform the symbolic math operations necessary to obtain the solution shown in Equation 5.47.

To illustrate, consider the block diagram of a system shown in Figure 5.31. The algebraic loop is shown in bold.

By algebraic manipulation or similar block diagram reduction techniques, the transfer function $Y(s)/R(s)$ is obtained as

$$\frac{Y(s)}{R(s)} = \frac{K + (1+K)G(s)}{(1+K) + (2+K)G(s)} \quad (5.48)$$

Suppose the constant $K = 1$ and the transfer function $G(s) = 1/(s + 10)$. The transfer function $Y(s)/R(s)$ reduces to

$$\frac{Y(s)}{R(s)} = \frac{0.5(s+12)}{s+11.5} \quad (5.49)$$

It is left as an exercise to demonstrate that a Simulink diagram based on the block diagram in Figure 5.31 and one with a single “Transfer Fcn” to implement Equation 5.49 produce identical outputs.

Unfortunately, the dynamic model equations rarely permit this approach. In most cases, the algebraic loop entails nonlinear blocks, making it difficult or impossible to reformulate the equations to produce a new block diagram with the algebraic loop removed. Several algebraic loops with shared blocks may exist, complicating matters even further.

A second approach to dealing with algebraic loops consists of inserting a “Memory” block into the loop. A “Memory” block is equivalent to a one-integration step delay. Its output is the input from the previous time step. This allows Simulink to calculate outputs of all the blocks in the algebraic loop in the proper sequence.

The system shown in Figure 5.32 consists of a cart with an inverted pendulum. The position of the cart $x(t)$ and the angle of the pendulum from the vertical $\theta(t)$ are of interest. The pendulum is

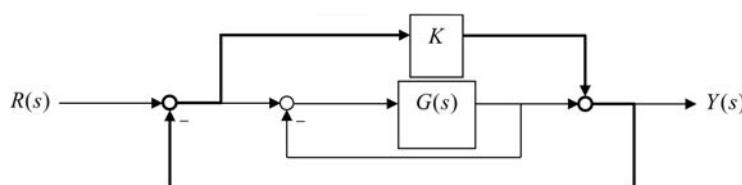


FIGURE 5.31 Block diagram of system with algebraic loop.

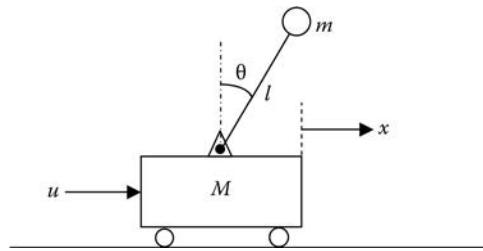


FIGURE 5.32 Inverted pendulum.

free to rotate without friction in a plane, and the cart moves along a frictionless surface. The input is a horizontal force u . The outputs are x and θ .

From Newton's second law (translation and rotation), the equations of motion are

$$(M+m)\ddot{x} - ml\dot{\theta}^2 \sin \theta + ml\ddot{\theta} \cos \theta = u \quad (5.50)$$

$$m\ddot{x} \cos \theta + ml\ddot{\theta} = mg \sin \theta \quad (5.51)$$

where

l is the length of the pendulum

m is the pendulum mass (assumed to be concentrated at the end)

M is the mass of the cart

g is the gravitational constant

Later, in Section 5.6, Equations 5.50 and 5.51 will be converted into a pair of equations, one for \ddot{x} and the other with $\ddot{\theta}$ where both are explicit functions of the state variables θ and $\dot{\theta}$.

A Simulink diagram of the system is shown in [Figure 5.33](#).

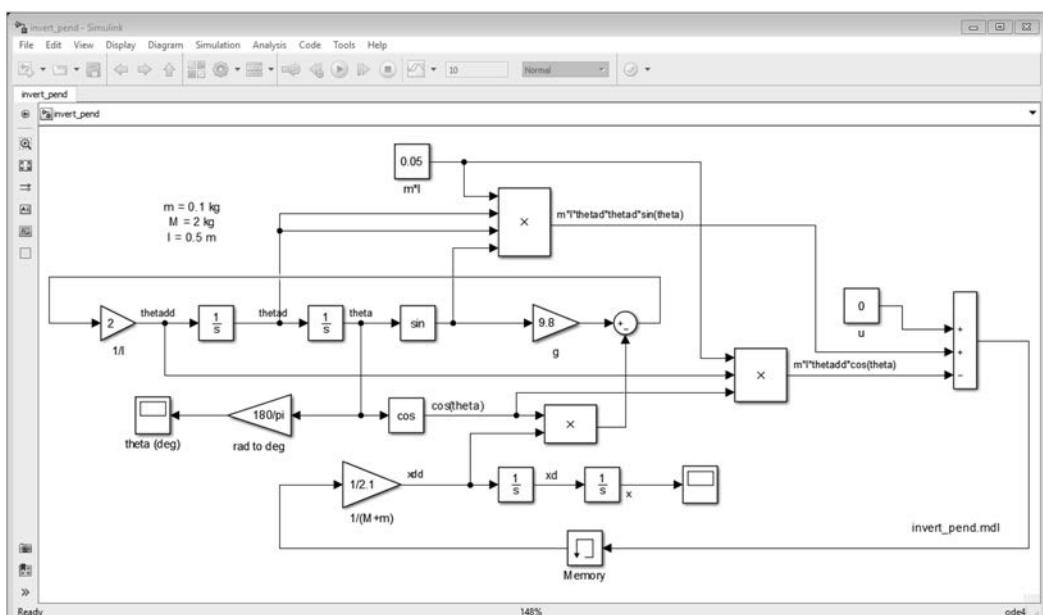


FIGURE 5.33 Simulink model of inverted pendulum with “Memory” block.

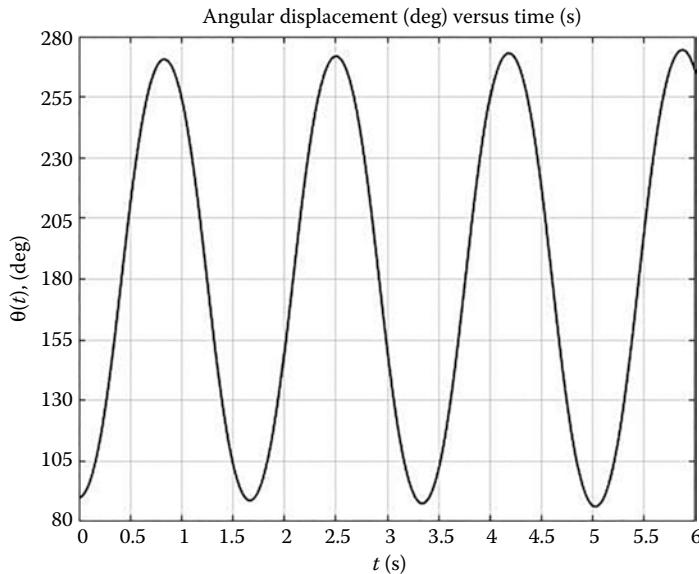


FIGURE 5.34 Simulink output for $\theta(t)$, $t \geq 0$ using a “Memory” block.

The algebraic loop shown in bold is broken by the insertion of a “Memory” block, eliminating the need for the Newton–Raphson iterative root solving at each integration step.

A simulation of the inverted pendulum when $u(t) = 0$, $t \geq 0$ was run for a period of 10 s using a fixed-step numerical integrator. All initial conditions are zero except the initial pendulum deflection, $\theta(0) = \pi/2$ rad. The output $\theta(t)$ is shown in Figure 5.34.

It is important to verify the results obtained when “Memory” blocks are employed to break algebraic loops. The delay introduced by the “Memory” block adversely affects the numerical accuracy and stability of the simulation. A considerable reduction in the time required to execute a simulation is hardly a suitable trade-off for inaccurate results. In other words, if the integration step size has to be reduced significantly to combat the existence of the “Memory” block, then the overall savings in execution time may be insignificant, or worse yet, the net result might be an overall increase in time of execution. A “Memory” block is worth considering when Simulink reports difficulty in converging to a solution of the implicit equations arising from an algebraic loop.

5.4.2 ALGEBRAIC EQUATIONS

While Simulink is generally used for simulating dynamic systems described by ordinary differential equations, it can also be used to solve a system of algebraic equations. For example, the algebraic equations

$$\left. \begin{aligned} y &= f(x) \\ x &= g(y) \end{aligned} \right\} \quad (5.52)$$

comprise an algebraic loop. Consider the dynamic system modeled by

$$\left. \begin{aligned} \frac{dy}{dt} &= F(x, y) = f(x) - y \\ x &= g(y) \end{aligned} \right\} \quad (5.53)$$

The two parts of Equation 5.53 represent the model of a first-order autonomous system, that is,

$$\frac{dy}{dt} + y - f[g(y)] = 0 \quad (5.54)$$

Suppose we are able to find an equilibrium point y_0 of the system described by Equation 5.54. Then (x_0, y_0) , where $x_0 = g(y_0)$, constitutes a solution to the system of algebraic equations in Equation 5.52. To illustrate, let us attempt to find a point that lies on the circle $x^2 + y^2 = 100$ and the curve $x = y^2/5$. In this case,

$$\begin{aligned} y &= f(x) = (100 - x^2)^{1/2} \\ x &= g(y) = \frac{y^2}{5} \end{aligned} \quad (5.55)$$

The Simulink diagram in Figure 5.35 incorporates an integrator for solution to

$$\frac{dy}{dt} = f(x) - y = (100 - x^2)^{1/2} - y \quad (5.56)$$

along with the block to generate x from the second of the two equations in Equation 5.55.

The search for the solution to the algebraic equations in Equation 5.55 begins at $(x(0), y(0))$ where $y(0)$ is the initial condition of the integrator and $x(0) = g[y(0)]$. Starting from the point $(0, 0)$, the approach to the equilibrium point y_0 and corresponding value of x_0 is viewable by clicking on the “Scope” block. The edited output is shown in Figure 5.36.

The solution $x_0 = 7.804$, $y_0 = 6.247$ is visible in the respective “Display” blocks shown in the Simulink diagram. The “XY Graph” block allows a view of the trajectory $x = g(y) = y^2/5$ from $(0, 0)$ up to the solution (x_0, y_0) , as shown in Figure 5.37.

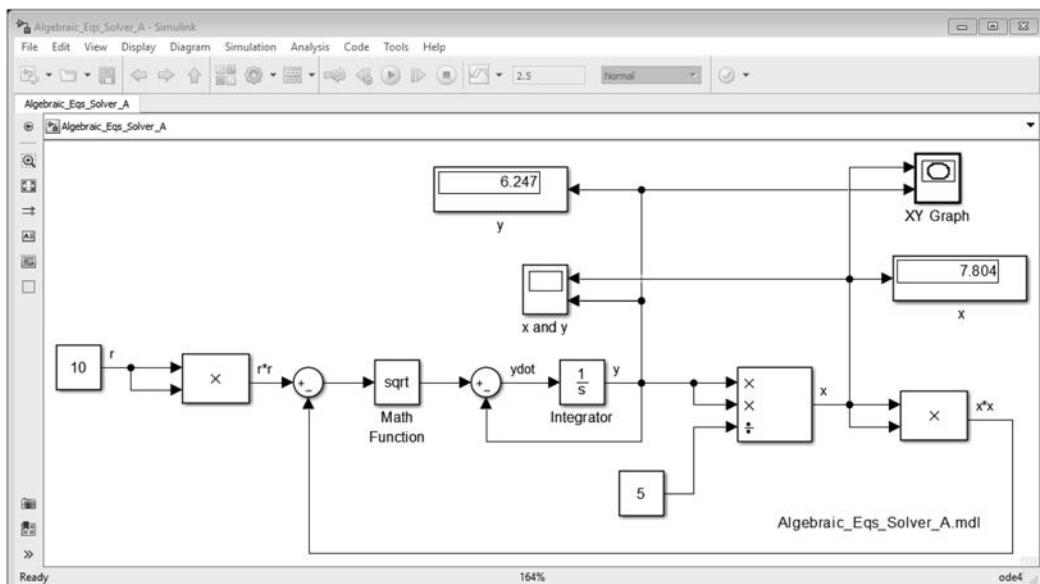


FIGURE 5.35 Simulink diagram for solving algebraic equations in Equation 5.55.

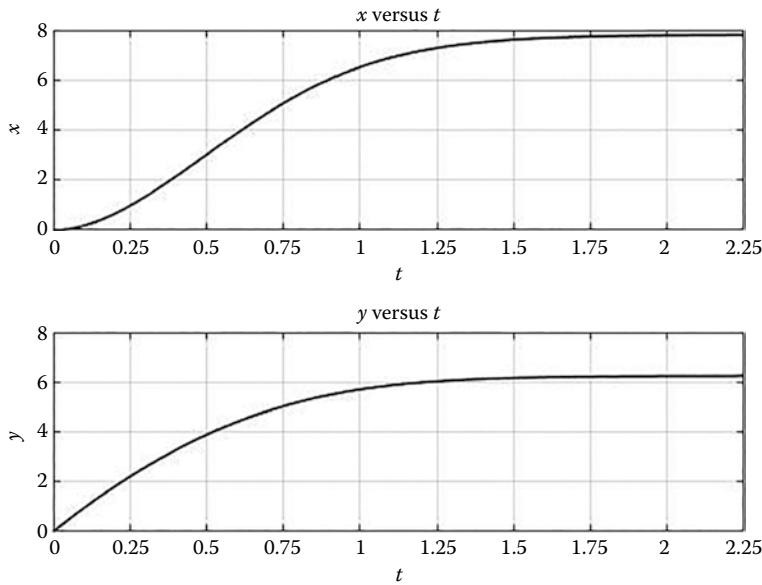


FIGURE 5.36 Graph of $x(t)$ and $y(t)$ from Simulink scope block.

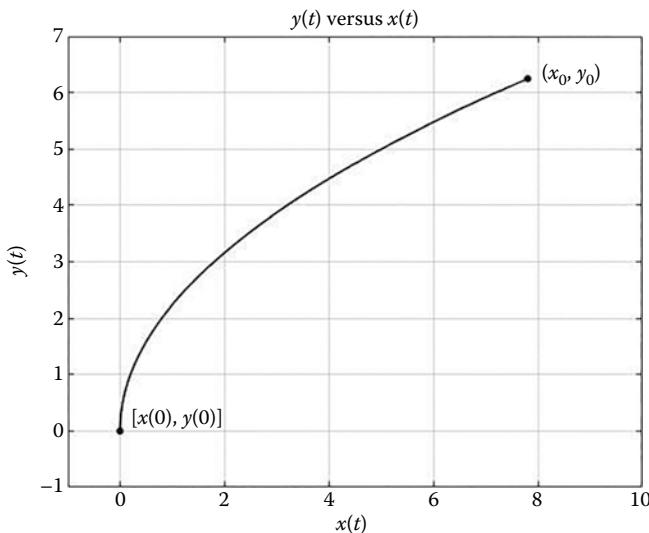


FIGURE 5.37 Trajectory from initial point $x(0) = 0, y(0) = 0$ to solution (x_0, y_0) .

If the simulation fails to converge to an equilibrium point, restarting from a new point $(x(0), y(0))$ may help. Only stable equilibrium points of Equation 5.53 can be discovered. Keep in mind that nonlinear algebraic equations may possess none, one, or several equilibrium points, and the number of such points may not be known beforehand.

A more direct approach to solving nonlinear algebraic equations with Simulink involves the use of an “Algebraic Constraint” block. This block changes its output in an iterative manner until its input approaches zero indicating that the algebraic constraint equation is satisfied, that is, the existence of a solution. Note that a feedback path must exist from the output to the input.

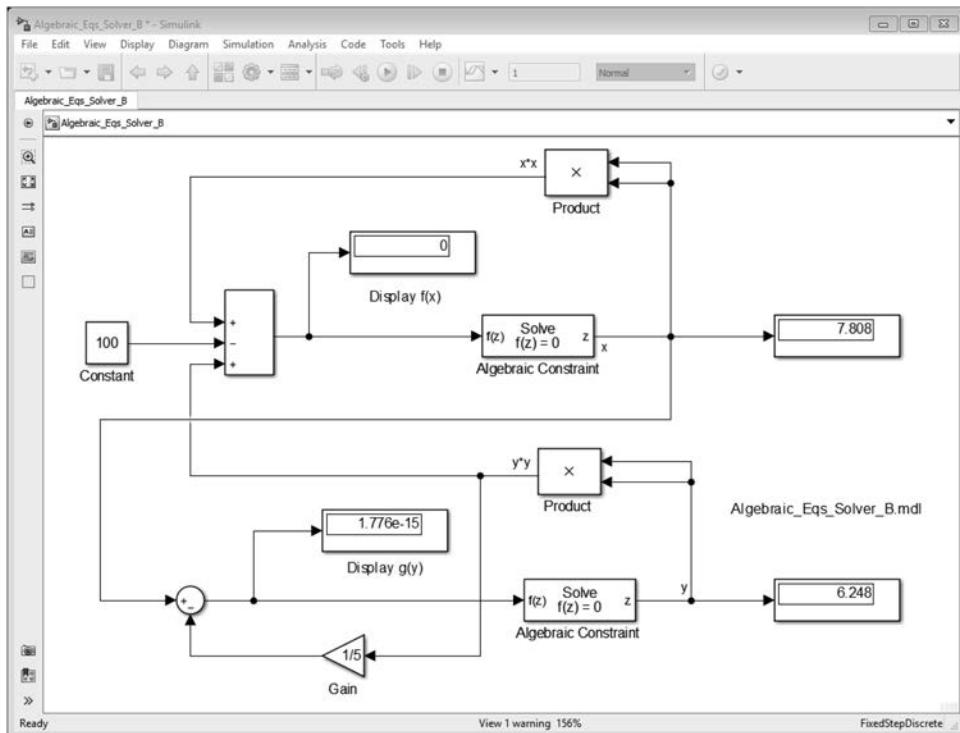


FIGURE 5.38 Using algebraic constraint blocks to solve algebraic equations in Equation 5.55.

The previous system of algebraic equations is solved using “Algebraic Constraint” blocks as shown in Figure 5.38. Initial guesses for the variables x and y are required. Note that the inputs to both “Algebraic Constraint” blocks have converged to zero and the algebraic states x and y are in agreement with the previous solution.

The “Algebraic Constraint” block is an effective tool for locating the equilibrium points of a nonlinear dynamic system.

EXERCISES

- 5.9 Run the Simulink model in Figure 5.33 using the “ode1” Euler integrator, and determine the largest step size possible for simulating the inverted pendulum dynamics with $u(t) = 0$, $t \geq 0$, and $\theta(0) = \pi/2$ for a period of 10 s. Repeat without the “Memory” block.
- 5.10 Starting with Equations 5.50 and 5.51 for the inverted pendulum,
- Find explicit functions $f(\theta, \dot{\theta}, u)$ and $g(\theta, \dot{\theta}, u)$ where

$$\ddot{x} = f(\theta, \dot{\theta}, u) \quad \text{and} \quad \ddot{\theta} = g(\theta, \dot{\theta}, u)$$

- Introduce state variables x_1, x_2, x_3 , and x_4 where $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$, and find the state derivative functions $f_1(x_1, x_2, x_3, x_4, u)$, $f_2(x_1, x_2, x_3, x_4, u)$, $f_3(x_1, x_2, x_3, x_4, u)$, and $f_4(x_1, x_2, x_3, x_4, u)$, where

$$\dot{x}_1 = f_1(x_1, x_2, x_3, x_4, u)$$

$$\dot{x}_2 = f_2(x_1, x_2, x_3, x_4, u)$$

$$\dot{x}_3 = f_3(x_1, x_2, x_3, x_4, u)$$

$$\dot{x}_4 = f_4(x_1, x_2, x_3, x_4, u)$$

- c. The outputs are $y_1 = x$ and $y_2 = \theta$. Find the output functions $g_1(x_1, x_2, x_3, x_4, u)$ and $g_2(x_1, x_2, x_3, x_4, u)$, that is,

$$y_1 = g_1(x_1, x_2, x_3, x_4, u)$$

$$y_2 = g_2(x_1, x_2, x_3, x_4, u)$$

- d. Prepare a Simulink diagram of the system based on the nonlinear state equations obtained in parts (b) and (c). Is an algebraic loop present?
- e. Compare outputs for $\theta(t)$, $t \geq 0$ using the Simulink diagram from [Figure 5.33](#) and a Simulink diagram based on the state equations $\dot{x} = f(x, u)$, $y = g(x, u)$ for the following cases:
- i. $u(t) = 0$, $t \geq 0$ and $x_1(0) = x_2(0) = 0$, $x_3(0) = 1^\circ$, $x_4(0) = 0$
 - ii. $u(t) = 0$, $t \geq 0$ and $x_1(0) = x_2(0) = x_3(0) = 0$, $x_4(0) = 10^\circ/\text{s}$

5.11 Rework the example designed to find the first quadrant solution to

$$y = f(x) = (100 - x^2)^{1/2} \quad \text{and} \quad x = g(y) = \frac{y^2}{5}$$

by looking for an equilibrium point of

$$\begin{aligned} \frac{dx}{dt} &= G(x, y) = g(y) - x \\ y &= f(x) \end{aligned}$$

5.12 Find both solutions to the algebraic equations

$$y = e^x - 1, y = 5 - (x - 1)^2$$

using “Algebraic Constraint” blocks.

- 5.13 Consider the system represented in block diagram form in [Figure 5.31](#) and the equivalent closed-loop transfer function in Equation 5.48.
Find the differential equation relating the output $y(t)$ and input $r(t)$ when

$$G(s) = \frac{K_1}{\tau s + 1} \tag{i}$$

$$G(s) = \frac{K_1(\tau_1 s + 1)}{\tau_2 s + 1} \tag{ii}$$

$$G(s) = \frac{K_1}{s^2 + 2\zeta\omega_n s + 1} \tag{iii}$$

Prepare Simulink diagrams to simulate the block diagram and transfer function representations of the system when $G(s) = 2/(0.5s + 1)$ and $K = 10$. Find and plot the responses to the following inputs:

- i. $r(t) = \hat{u}(t)$, the unit step input
- ii. $r(t) = e^{-t/2}$, $t \geq 0$
- iii. See graph of $r(t)$ in [Figure E5.13](#)

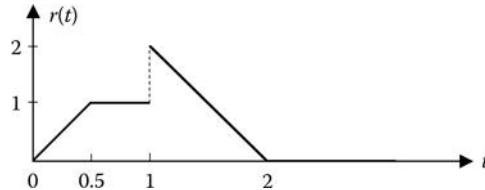


FIGURE E5.13

5.5 MORE SIMULINK BLOCKS

In this section, we introduce additional Simulink blocks to extend the simulation capabilities developed so far. The next example is a common one from the field of traffic engineering. The objective is to formulate a mathematical model suitable for describing the characteristics of a driver/vehicle attempting to follow a lead vehicle in a single lane of traffic. The result is referred to as a microscopic car-following model. Car-following models are an essential component of traffic simulation software used to predict traffic flows in tunnels and other roads where passing is restricted.

The basic situation is illustrated in [Figure 5.39](#), which shows a lead vehicle ($n - 1$) and a following vehicle (n), each of length L .

The system, comprised of the lead and following vehicle, is driven (no pun intended) by the speed of the lead vehicle \dot{x}_{n-1} , and the outputs include $\{x_{n-1}, x_n, \dot{x}_n, \ddot{x}_n\}$ in addition to the following quantities, which relate directly to the combination of lead and following vehicle movements.

$$\text{Vehicle spacing: } s_n = x_{n-1} - x_n \quad (5.57)$$

$$\text{Vehicle following distance: } d_n = (x_{n-1} - L) - x_n \quad (5.58)$$

$$\text{Speed difference: } \Delta \dot{x}_n = \dot{x}_{n-1} - \dot{x}_n \quad (5.59)$$

$$\text{Vehicle gap: } g_n = \frac{x_{n-1} - x_n}{\dot{x}_n} \quad (5.60)$$

The subscripts “ $n - 1$ ” and “ n ” are used, so that we can model a platoon consisting of a lead vehicle and several following vehicles. Except for the platoon leader and the last vehicle in the

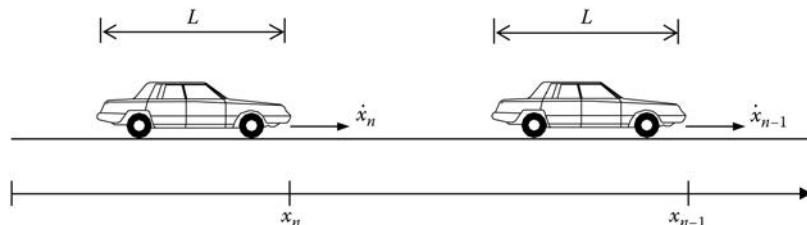


FIGURE 5.39 Diagram showing lead and following vehicles.

platoon, each vehicle operates in a following and lead vehicle mode as depicted in [Figure 5.39](#). Platoon dynamics is considered in the next section.

We have yet to formulate a mathematical model that governs the motion of the following vehicle in the case of small-to-moderate vehicle spacing. Note that car-following models are not applicable at low traffic densities since each vehicle is essentially a leader moving independently of the preceding vehicle.

Standard practice is to postulate an equation for the acceleration of the following vehicle in response to certain stimuli that are based on the relative movements of the two vehicles, that is,

$$\ddot{x}_n(t+T) = f(x_{n-1}(t), x_n(t), \dot{x}_{n-1}(t), \dot{x}_n(t)) \quad (5.61)$$

The acceleration response is delayed by an amount T , which represents the sum of the driver's cognition and reaction times in addition to the vehicle response time. The literature is replete with articles and chapters in books describing suitable candidates for the function "f()" in Equation 5.61 (Bender and Fenton 1966; Haberman 1977; Mesterton-Gibbons 1988; Aycin and Benekohal 2001).

The block diagram in [Figure 5.40](#) represents a specific function developed by the author used to simulate realistic traffic in a driving simulator.

The driver/vehicle combination behaves like a regulatory controller with output $\ddot{x}_n(t+T)$, a function of two error terms e_1, e_g and the following vehicle's speed \dot{x}_n . The first error term e_1 is the difference between 0 and $\dot{x}_n - \dot{x}_{n-1}$ weighted by the reciprocal of the spacing $(x_{n-1} - L) - x_n$. The second term e_g represents a gap error, that is, the difference between some desirable gap G and the actual gap g . The driver/vehicle controller attempts to drive both errors to zero by implementation of the control law

$$\ddot{x}_n(t+T) = K_1(e_1, \dot{x}_n) \cdot e_1 + K_g(e_g, \dot{x}_n) \cdot e_g \quad (5.62)$$

Note that when \dot{x}_{n-1} is constant and both errors are zero, the following vehicle is traveling at the same speed with a separation $x_{n-1} - x_n = G\dot{x}_{n-1}$.

The functions $K_1(e_1, \dot{x}_n)$ and $K_g(e_g, \dot{x}_n)$ are implemented as shown in [Tables 5.1](#) and [5.2](#). The constants $K_{1,a}, K_{1,d}, K_{g,d}$, and $K_{g,a}$ are gain parameters reflecting driver aggressiveness, SL is the speed limit, and Δ is a threshold above the speed limit.

A block diagram of the system is shown in [Figure 5.41](#). The blocks to limit the acceleration and speed are self-explanatory. The spacing limiter assures that the minimum vehicle separation $x_{n-1} - x_n$ is greater than one car length at all times.

A Simulink diagram of the system is shown in [Figure 5.42](#). The M-file "Ch5_cfparams1.m" assigns values to the system parameters referenced in a number of the Simulink blocks. Accordingly,

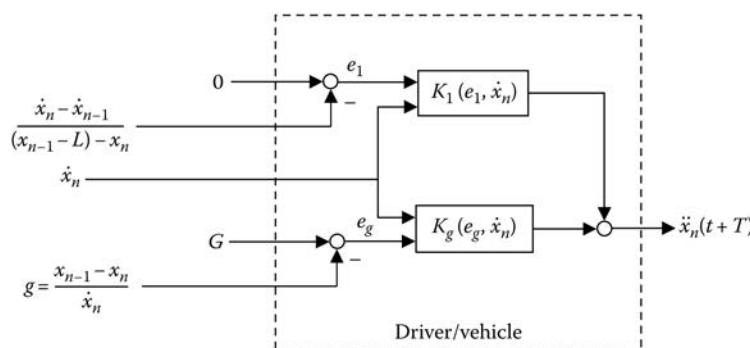


FIGURE 5.40 Block diagram of a car-following model.

TABLE 5.1
Function $K_1(e_1, \dot{x}_n)$

		$\leq SL + \Delta$	$> SL + \Delta$
e_1			
\dot{x}_n			
> 0	$K_{1,a}$	0	
≤ 0	$K_{1,d}$		$K_{1,d}$

TABLE 5.2
Function $K_g(e_g, \dot{x}_n)$

		$\leq SL + \Delta$	$> SL + \Delta$
e_1			
\dot{x}_n			
> 0	$K_{g,d}$	$K_{g,d}$	
≤ 0	$K_{g,d}$	0	

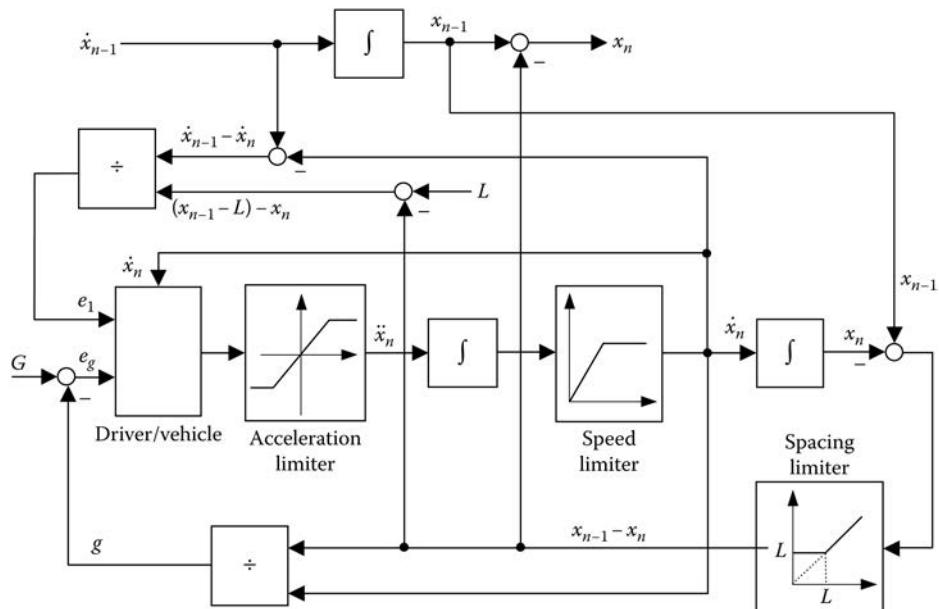


FIGURE 5.41 Block diagram of a car-following system.

it must be run prior to executing the simulation model file “*car_following.mdl*.” The new Simulink blocks in [Figure 5.42](#) and their function are described briefly as follows.

1. “Clock”: Outputs the simulation time variable “*t*” for the “Lookup Table” block.
2. “Lookup Table”: Linearly interpolates between specified data points to generate the lead car speed profile.
3. “MATLAB fcn”: Passes the inputs “*xld*,” “*e1*,” and “*eg*” to the MATLAB function “*acc.m*,” which computes the vehicle’s acceleration response.

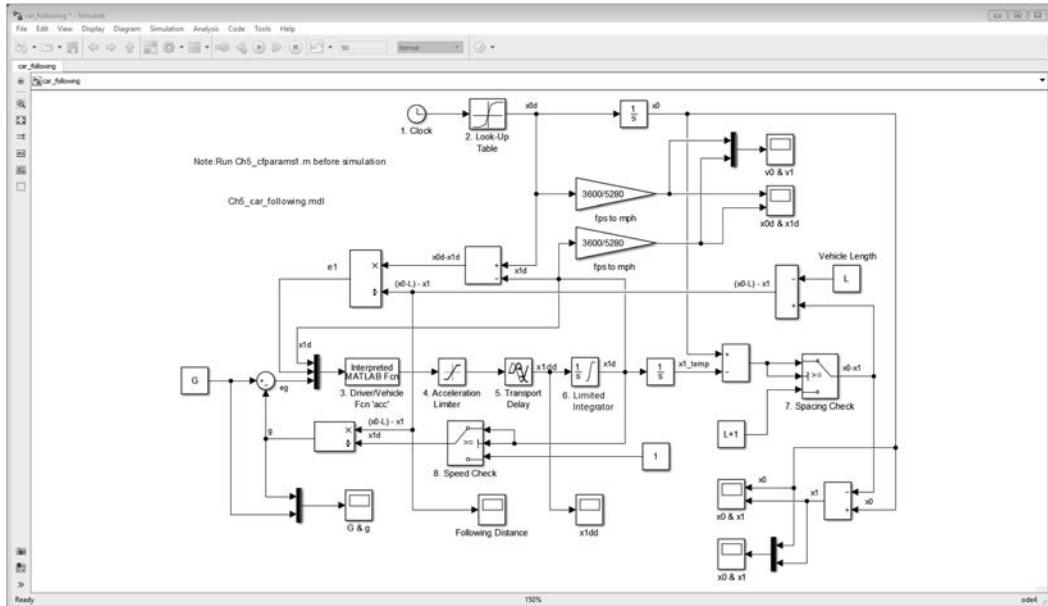


FIGURE 5.42 Simulink diagram for a car-following system.

4. “Saturation”: Sets limits for minimum and maximum vehicle acceleration.
5. “Transport Delay”: Delays vehicle acceleration by T , the driver/vehicle reaction time.
6. “Limited Integrator”: An integrator configured to limit vehicle speed between zero and a maximum value of “vmax.”
7. “Switch”: Logical blocks that limit the spacing “ $x0 - x1$ ” to at least $L + 1$ ft and the speed “ xld ” to at least 1 ft/s for calculation of the gap g .

Access to the MATLAB Workspace during execution allows Simulink block parameters to be variables specified in MATLAB script files. For example, The “Lookup Table” block parameters “T0,” “T1,” “T2,” “A1,” and “A2” shown in Figure 5.43 are set in the M-file “Ch5_cfparamsl.m.”

The “MATLAB Function” block is a powerful feature of Simulink, which exploits the tight integration between MATLAB and Simulink. The Simulink block outputs “ xld ,” “ $e1$,” and “ eg ”

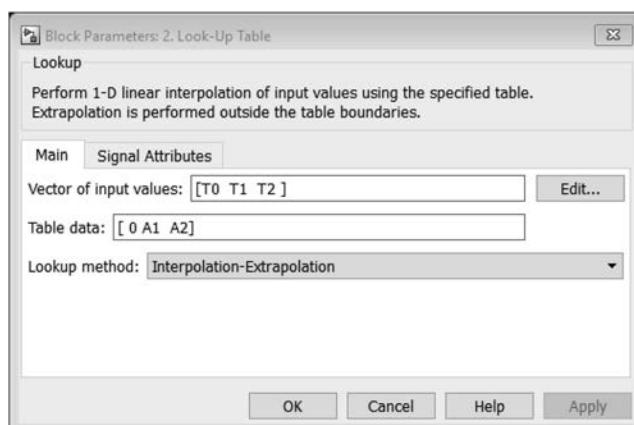


FIGURE 5.43 “Lookup Table” block parameters.

are accessible as inputs to the MATLAB function M-file “*acc.m*,” which implements the car-following algorithm in Equation 5.62. The computed output is sent to the “Acceleration Limiter” block in [Figure 5.42](#).

The M-file “*acc.m*” is listed as follows.

```
% function acc.m computes the temporary commanded acceleration
function y = acc(x1d,e1,eg,K1d,K1a,Kgd,Kga,SL,delta)
if e1< = 0
    y1 = K1d*e1;
elseif x1d< = SL+delta
    y1 = K1a*e1;
else
    y1 = 0
end
if eg>0
    yg = Kgd*eg;
elseif x1d< = SL+delta
    yg = Kga*eg;
else
    yg = 0;
end
y = y1+yg;
```

The results of simulating a pair of initially stopped vehicles, with one car length separation, followed by the lead vehicle accelerating (with constant acceleration) to 60 mph in 30 s are obtained by running the M-file “*Ch5_Fig5_44_thru_5_48.m*” and shown in [Figures 5.44](#) through [5.48](#). The commanded gap G is 2 s, the value recommended for highway driving by The American Automobile Association.

The initial blip in speed of the following vehicle (see [Figure 5.45](#)) is due to the excessive gap g that results whenever the following vehicle is moving at very low speeds. The car-following model, Equation 5.62, implemented in the M-file “*acc.m*” is not robust, that is, it is not valid at following vehicle speeds close to zero, which occurs when the simulation begins. Similar artifacts are present

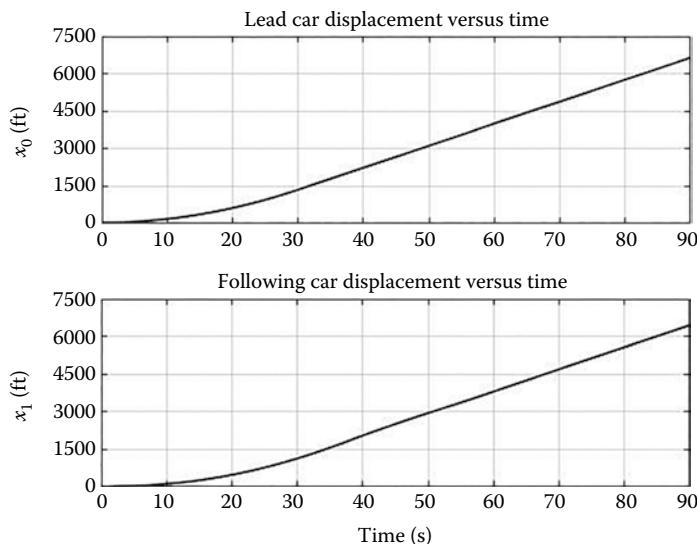


FIGURE 5.44 Lead and following vehicle positions.

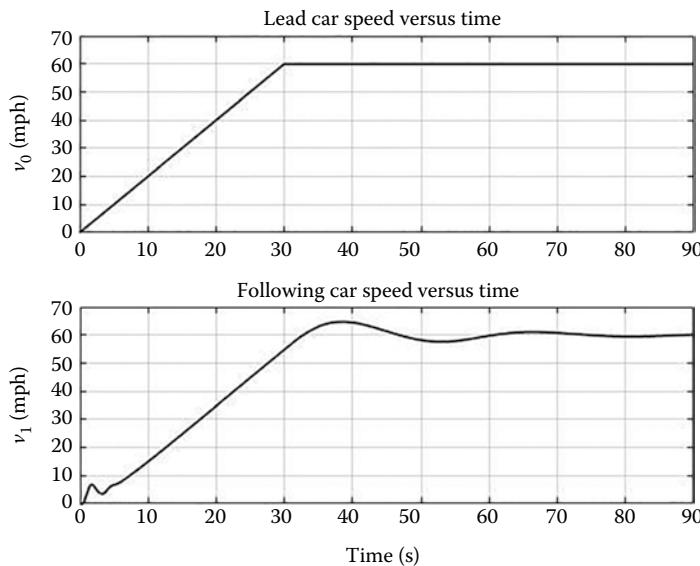


FIGURE 5.45 Lead and following vehicle speeds.

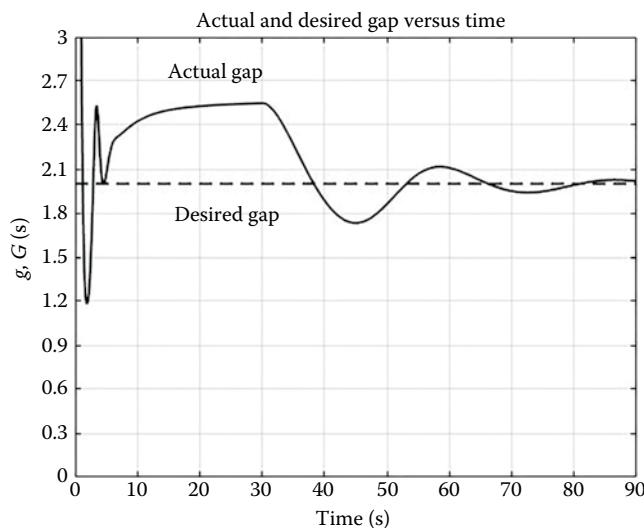


FIGURE 5.46 Desired and actual gaps.

in the gap (Figure 5.46) and acceleration (Figure 5.48) plots. One of the exercise problems addresses this point further.

5.5.1 DISCONTINUITIES

Each of the nonlinear elements presented in Section 2.7 are available as blocks in Simulink. From within the Simulink Library Browser, click on “Discontinuities” to display the element blocks as shown in Figure 5.49.

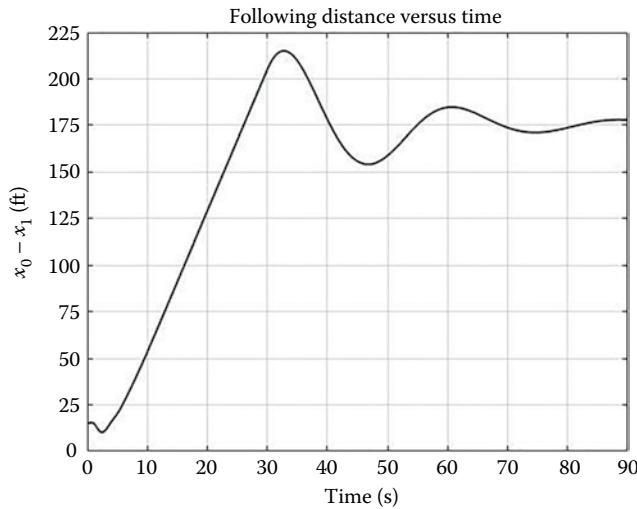


FIGURE 5.47 Following distance.

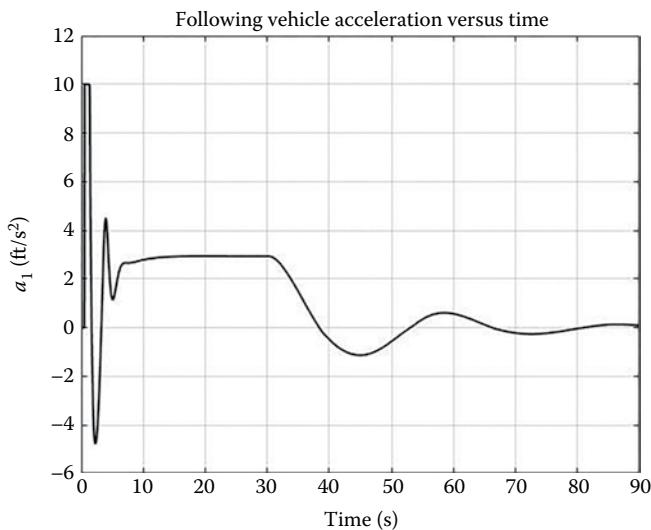


FIGURE 5.48 Simulink output of following vehicle acceleration.

In the right-hand column are nonlinear blocks for friction, dead zone, saturation, backlash, hysteresis (relay), and quantization.

5.5.2 FRICTION

Figure 5.50 shows the “Coulomb and Viscous Friction” parameter dialog box.

While the default conditions are shown in Figure 5.50, a more practical way to use the block is to assign a scalar value to the Coulomb friction value (Offset). This would represent the coefficient of *static* friction as in the case of initiating the motion of a sliding mass. Of course, the Coefficient of viscous friction (Gain) corresponds to the kinetic friction as the coefficient of the velocity term in the dynamic equations of motion.

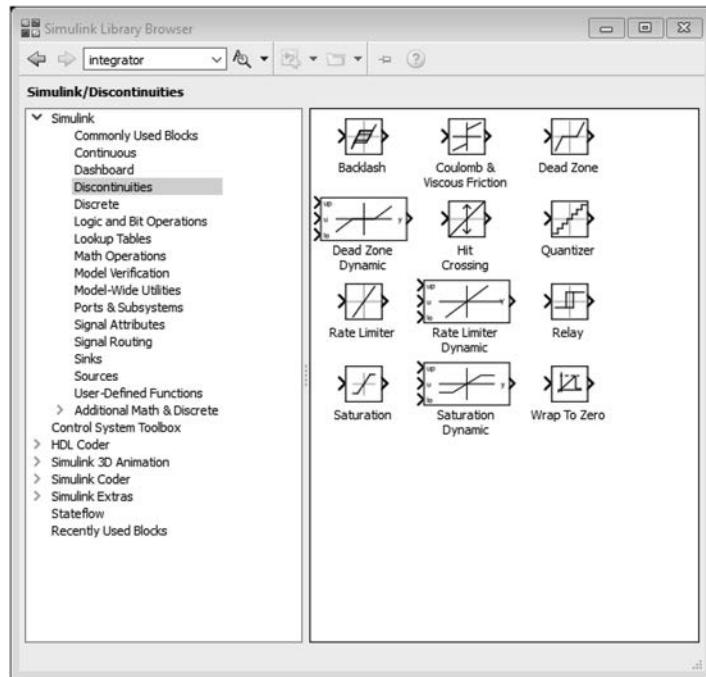


FIGURE 5.49 Simulink Library Browser—Discontinuities.

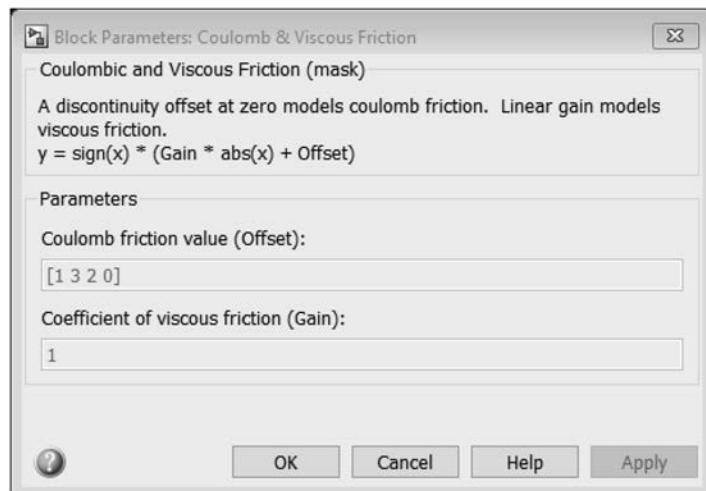


FIGURE 5.50 “Coulomb and Viscous Friction” parameter dialog box.

A detailed description of the “Coulomb and Viscous Friction” block can be found by clicking on Help from the dialog box.

5.5.3 DEAD ZONE AND SATURATION

Figure 5.51 shows the “Dead Zone” parameter dialog box.

The parameter dialog box for the dead zone block is rather intuitive. The user simply sets the beginning and the end of the dead zone according to the input being sent to the block. In the default

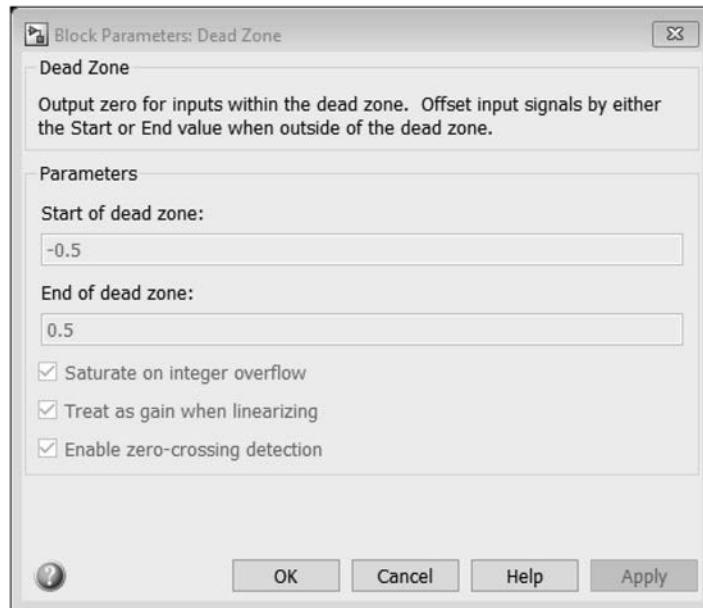


FIGURE 5.51 “Dead Zone” parameter dialog box.

example, the output is zero if the input signal is between -0.5 and 0.5 . Otherwise, the output tracks the input.

A detailed description of the “Dead Zone” block can be found by clicking on Help from the dialog box.

Figure 5.52 shows the “Saturation” parameter dialog box.

The parameter dialog box for the saturation block is also intuitive. The user simply sets the beginning and the end of the saturation limits according to the input being sent to the block. In the default example, the output is -0.5 for input values less than -0.5 , the output tracks the input between -0.5 and 0.5 , and the output is 0.5 for input values greater than 0.5 .

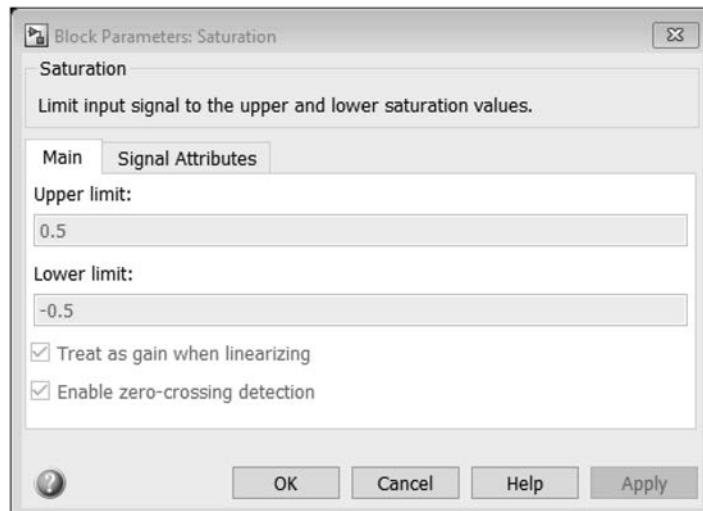


FIGURE 5.52 “Saturation” parameter dialog box.

A detailed description of the “Saturation” block can found by clicking on Help from the dialog box.

5.5.4 BACKLASH

[Figure 5.53](#) shows the “Backlash” parameter dialog box.

For the backlash block, the user sets the Deadband width and the Initial output. If the defaults are taken, the output of the backlash block is split evenly between upper and lower values of the input. For example, if the input is a square wave with an upper limit of +1 and a lower limit of -1, the deadband width is centered on zero (the Initial output default), and half of the Deadband width (0.5) is taken from the upper limit while the other half of the Deadband width is taken from the lower limit yielding an output of +0.5 (when the input is +1) and an output of -0.5 (when the input is -1).

Clarifying, if the Deadband width is 0.4, then 0.2 will be taken from each of the input values, that is, the output is a square wave between +0.8 and -0.8 (using the input square wave between +1 and -1).

If the Initial output is nonzero and exceeds the input value, then the backlash block can be used to simulate gears that have yet to be engaged. Continuing with the same square wave input between +1 and -1, if the Initial output is set to 2, then the output is +1 plus half of the Deadband width or +1.2. It is only when the gears engage that the output returns to the limits of +0.8 and -0.8.

A detailed description of the “Backlash” block can found by clicking on Help from the dialog box.

5.5.5 HYSTERESIS

One of the examples in Section 2.7 on nonlinear systems dealt with maintaining the temperature inside a building using a thermostat to control the heat from a furnace. The building temperature and thermostat control are governed by the equations repeated as follows.

$$\tau \frac{dT}{dt} + T = RQ + T_0 \quad (5.63)$$

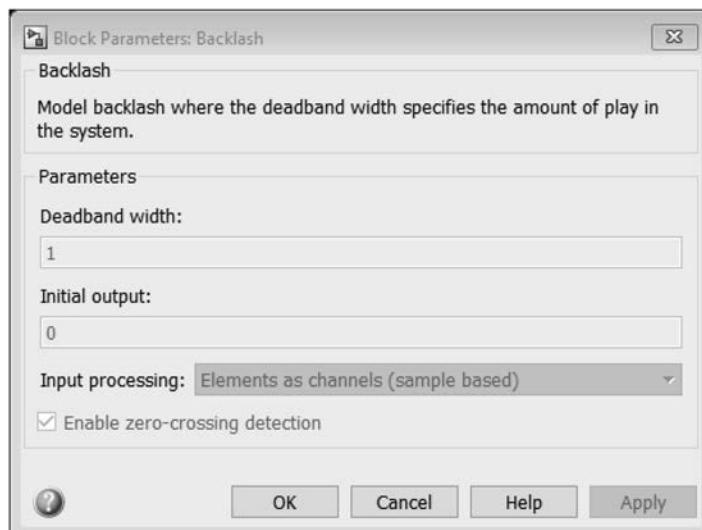


FIGURE 5.53 “Backlash” parameter dialog box.

$$Q = \begin{cases} \bar{Q}, T \leq T_d - \Delta & \text{or} \\ T_d - \Delta < T < T_d + \Delta & \text{and} \quad \frac{dT}{dt} > 0 \\ 0, T > T_d + \Delta & \text{or} \\ T_d - \Delta < T < T_d + \Delta & \text{and} \quad \frac{dT}{dt} < 0 \end{cases} \quad (5.64)$$

where

T is the building temperature (°F)

Q is the heat input from furnace (Btu/h)

T_0 is the outside temperature (°F)

R is the thermal resistance of building (°F/Btu/h)

τ is the time constant of building temperature response (h)

\bar{Q} is the rating of furnace (Btu/h)

T_d is the thermostat setting (°F)

Δ is the dead zone parameter for thermostat

The hysteresis effect associated with the thermostat, Equation 5.64, is illustrated graphically in [Figure 2.43](#). This type of nonlinear behavior is readily simulated using a “Relay” block from the Simulink “Discontinuities” sublibrary. A description of the “Relay” block can be found in the online Simulink Reference, which contains detailed documentation for each block (see [Figure 5.54](#)).

A Simulink diagram for simulating building temperature with conditions as described in Example 2.11 is shown in [Figure 5.55](#). The “Relay” block parameter box is also shown.

The furnace output and building temperature are shown in [Figure 5.56](#). The building temperature increases from 50°F, the constant outside temperature. The hysteresis kicks in once the temperature

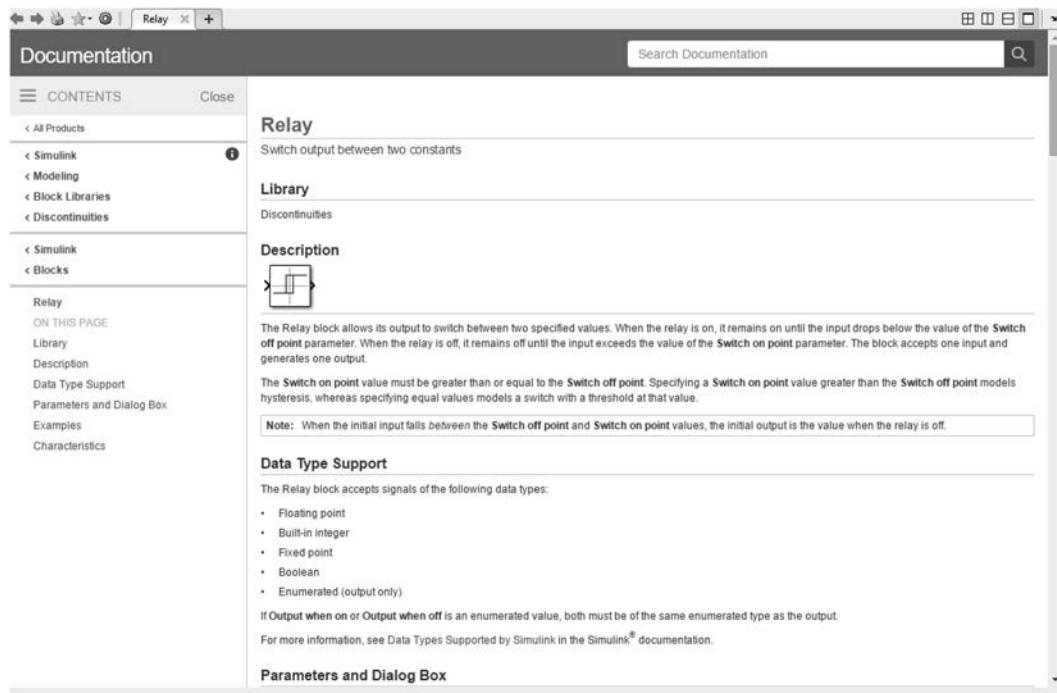


FIGURE 5.54 Description of the Simulink “Relay” block.

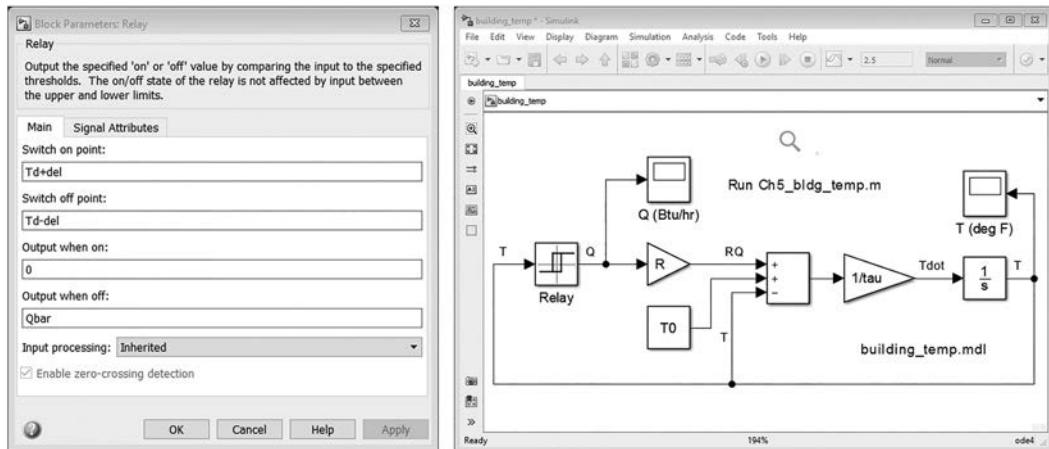


FIGURE 5.55 Simulink diagram for a simulation of building temperature using the “Relay” block for thermostat.

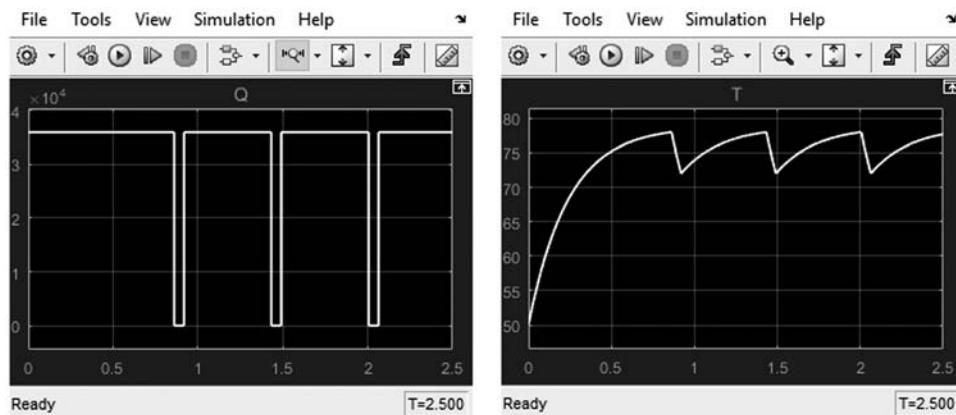


FIGURE 5.56 Furnace output and building temperature.

exceeds “ $Td-del$ ” = $75 - 3 = 72^\circ\text{F}$. The initial portion of the building temperature response in Figure 5.56 is identical to the temperature response graphed in Section 2.7.

5.5.6 QUANTIZATION

Figure 5.57 shows the “Quantization” parameter dialog box.

To demonstrate the use of the quantization block, use the default value given for the Quantization interval, that is, 0.5, where the input is a ramp with a slope of 0.5. When the input is between 0 and 0.5, the quantized output is 0; between 0.5 and 1, the quantized output is 0.5; et cetera.

Following the example from Section 2.7, the Quantization interval is set to 10/256. This corresponds to the analog input range of 0–10V for an 8-bit microprocessor with $2^8 = 256$ states, 0–255.

A detailed description of the “Quantization” block can be found by clicking on Help from the dialog box.

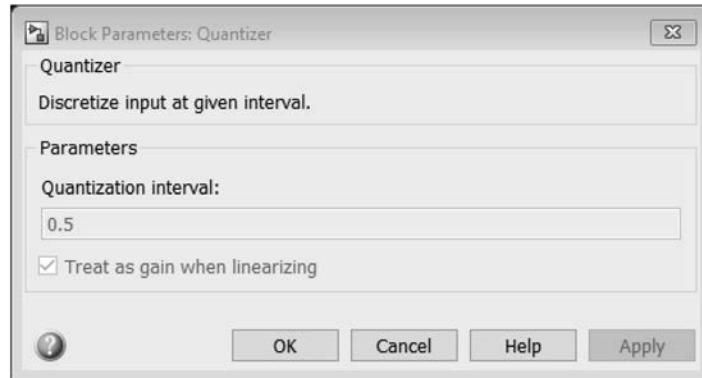


FIGURE 5.57 “Quantization” parameter dialog box.

EXERCISES

- 5.14 Simulate the motion of a lead vehicle and following vehicle for a period of time sufficient to reach steady state for the conditions given in [Table E5.14](#). The lead vehicle speed is

$$\dot{x}_0(t) = \begin{cases} v_0, & 0 \leq t \leq P/4 \\ v_0 + A \sin \frac{2\pi t}{P}, & P/4 < t \leq 5P/4 \\ v_0, & t > 5P/4 \end{cases}$$

Plot graphs similar to the ones shown in [Figures 5.44 through 5.48](#).

TABLE E5.14

Parameters	Case I	Case II	Case III
v_0	50 mph	60 mph	70 mph
A	5 mph	5 mph	10 mph
P	30 s	30s	60s
G	2 s	2.5s	3s
$x_0(0)$	Gv_0	Gv_0	Gv_0
$\dot{x}_1(0)$	v_0	v_0	v_0
$x_1(0)$	0 ft	0 ft	0 ft
SL	v_0	v_0	v_0
Δ	5 mph	10 mph	55 mph
a_{\min}	-10 ft/s ²	-12 ft/s ²	-15 ft/s ²
a_{\max}	10 ft/s ²	12 ft/s ²	15 ft/s ²
v_{\max}	90 mph	90 mph	90 mph
$K_{1,a}$	3 ft/s	4 ft/s	5 ft/s
$K_{1,d}$	3 ft/s	4 ft/s	5 ft/s
$K_{g,a}$	-5 ft/s ³	-5 ft/s ³	-5 ft/s ³
$K_{g,d}$	-4 ft/s ³	-4 ft/s ³	-4 ft/s ³
T (delay)	0.5s	0.75s	1 s
L	15 ft	15 ft	15 ft

- 5.15 Consider the second column of above table as baseline numerical values for simulation of a pair of vehicles. Perform a simulation study to analyze the effect of the desirable gap G on the following vehicle’s ability to follow at or near the desirable gap. Run the simulation for

$G = \{1, 1.5, 2, 2.5, 3, 3.5, 4\}$ for a duration of 3P s, and record the value of the average absolute gap error, that is,

$$|e_g|_{ave} = \frac{1}{2P} \int_P^{3P} |g(t) - G| dt$$

Plot $|e_g|_{ave}$ vs. G

- 5.16 Improve the robustness of the car-following simulation to make the output more realistic at very low vehicle speeds. Specifically, modify the code in “acc.m” and add additional blocks as necessary to the Simulink diagram in Figure 5.42. Use the baseline values from the above table to simulate the following vehicle’s response to a lead car with speed profile shown in Figure E5.16.

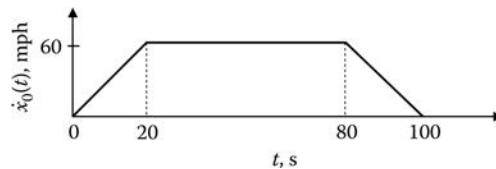


FIGURE E5.16

- 5.17 The Simulink diagram in Figure 5.42 contains a “Switch” block to maintain the vehicle separation $x_n - x_{n-1}$ greater than $L + 1$. This effectively eliminates the possibility of a rear end collision.
- Remove the “Switch” block and add the necessary Simulink blocks to detect the existence of a collision and halt the simulation.
 - Use the lead car profile in above figure and adjust the parameters Δ and $K_{1,a}, K_{1,d}, K_{g,a}, K_{g,d}$ to force a rear-end collision.
- 5.18 Flow into the tank shown in Figure E5.18a is either on or off. It turns on when the level falls below 20 ft and remains on when the tank is filling until there is 25 ft of liquid in the tank. It remains off as the tank empties until the level falls below 20 ft. In the on condition, the flow rate is 18 ft³/min. The tank dynamics are described by

$$A \frac{dH}{dt} + F_0 = F_1, \quad F_0 = \alpha H^{1/2}$$

where $A = 50$ ft², $\alpha = 3$ ft³/min/ft^{1/2}, and $H(0) = 0$ ft.

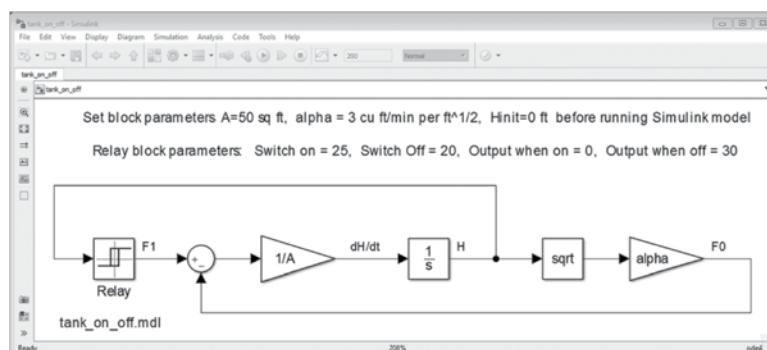


FIGURE E5.18A

- a. Develop your own Simulink diagram or use the one shown in [Figure E5.18b](#) to simulate the tank dynamics for a period of time sufficient to see several cycles of filling and emptying. Plot the tank level and the two flows vs. time.

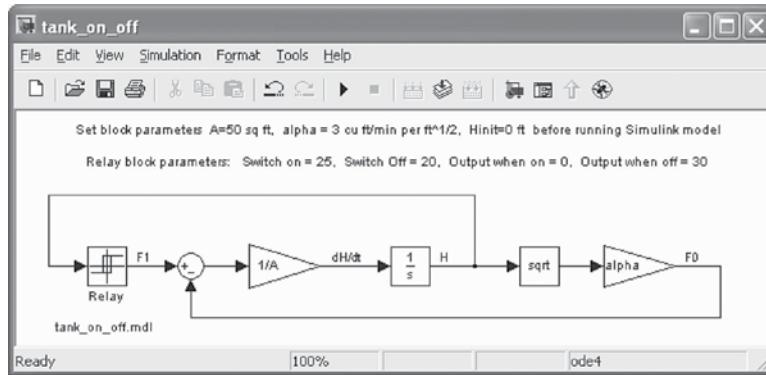


FIGURE E5.18B

- b. What is the minimum flow necessary to assure the tank is capable of filling to a level of 25 ft? Find the answer analytically and verify with Simulink.
 - c. Supplement your Simulink diagram with additional blocks to measure the percentage of time during a cycle in which the tank is filling up.
 - d. Instead of switching the flow off and on immediately when the level reaches 25 and 20 ft, respectively, suppose the flow switches off 30 s after the tank level reaches 25 ft and switches on 30 s after the tank level falls to 20 ft. The tank is 25 ft tall. Add Simulink blocks to account for spillover from the tank. Plot the flow in, flow out, spillover, and tank level vs. time.
- 5.19 Cascading the dead zone and saturation blocks, model the valve described in Section [2.7.2](#). Set the dead zone block's parameters to model opening currents of -0.5 and 0.5 amp (default settings) and set the saturation block's parameters to model saturation currents of -1.0 and 1.0 amp. Verify that the valve is modeled correctly by using a ramp input and observing the characteristics shown in [Figure 2.39](#).

5.6 SUBSYSTEMS

As the physical systems we model become progressively more complex, the Simulink representation increases in size, that is, the number of blocks required to model the systems' dynamics grows significantly. A Simulink diagram with hundreds of blocks makes it difficult, if not impossible, to understand the interactions among the systems' components. A more instructive approach consists of grouping specific blocks associated with various subsystems into single entities. At the highest level, the system is viewed in terms of the interactions between these entities.

This hierarchical approach is illustrated for the case of modeling the dynamics of an automobile. [Figure 5.58](#) shows a block diagram of the top level description for modeling the dynamics. At this level, the important interconnections between individual subsystems are identified.

The next step requires the development of concrete descriptions of the individual subsystems, either in mathematical or block diagram form. The mathematical models are transformed into Simulink models that are reusable, much in the same way a procedural function is used in high-level programming languages. Multiple levels are possible in this modeling hierarchy. Moving down one or more levels from the top subsystem level provides more of a microscopic, that is, detailed, description involving low-level components.

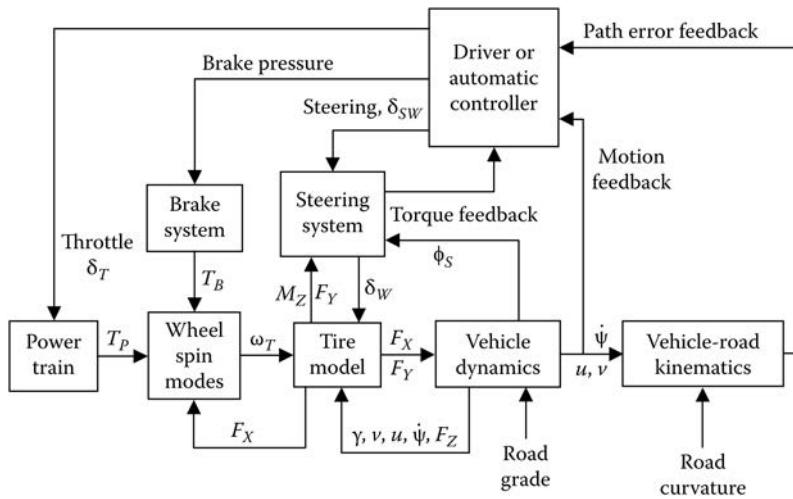


FIGURE 5.58 Top-level description of vehicle dynamics model. (From Allen, R.W. and Rosenthal, T. Systems technology/requirements for vehicle dynamics simulation models, Society of Automotive Engineers, SAE 941075, 1994.)

An advantage of this approach is the distribution of the modeling effort to individuals with expertise necessary for modeling the individual subsystems. For example, the “Tire Model” subsystem is a critical component in modeling vehicle response. A person knowledgeable in tire/road surface interaction phenomena and the properties of specific tires is needed to develop models that will produce correct tire forces required by the equations of motion.

Suppose a question arises concerning the handling characteristics of a vehicle with different classes of tires. The existing vehicle subsystem models are already in place and can be reused with a “Tire Model” developed specifically for the class of tire under consideration.

5.6.1 PHYSBE

PHYSBE is a benchmark simulation of the human circulatory system. It was first introduced by John Mcleod in 1966 in an article titled “PHYSBE … a Physiological Simulation Benchmark Experiment.” Over the years, it has appeared in numerous references involving modeling and simulation. The underlying dynamics have been simulated using the popular continuous simulation programs including Simulink.

The human circulatory system is represented by three main components: the lungs (pulmonary circulation), heart (coronary circulation), and the rest of the body (systemic circulation). Coronary and systemic circulations were further divided into subsystems as shown in the Simulink diagram in [Figure 5.59](#) (provided by The Mathworks, Inc.).

The simulation computes pressures, blood flows, volumes, temperatures, and heat flows after a number of parameters describing the physical nature of each subsystem have been specified. The dynamics of each subsystem are hidden in the macroscopic view of the human circulatory system in [Figure 5.59](#). A detailed description of the individual subsystem models is accessible by “looking inside” each of the blocks. For example, the LUNGS subsystem opens up to reveal the components used in modeling the blood flow, blood temperature, heat content, and heat dissipation within the lungs (see [Figure 5.60](#)).

The modular structure of the overall system makes it relatively simple to simulate, for example, the effects of partial blockages in the blood vessels of the systemic regions or the effect of changes in vascular compliance on blood pressure.

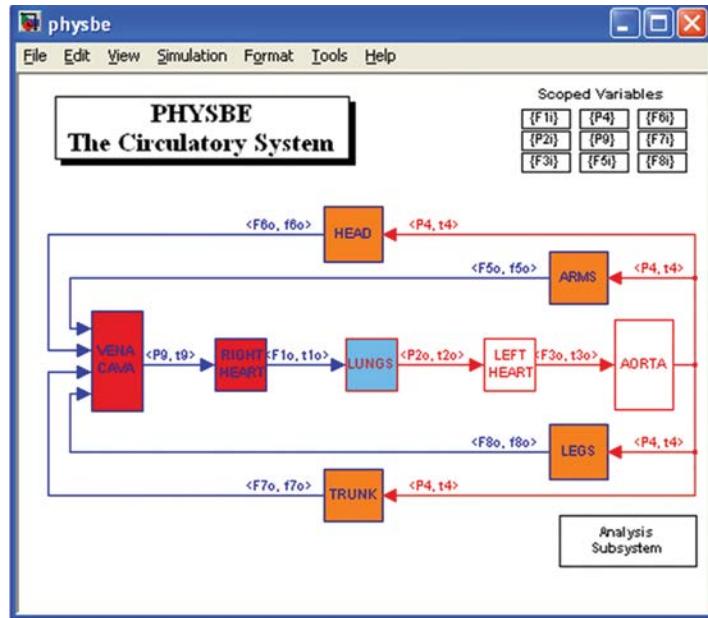


FIGURE 5.59 Simulink diagram of PHYSBE model.

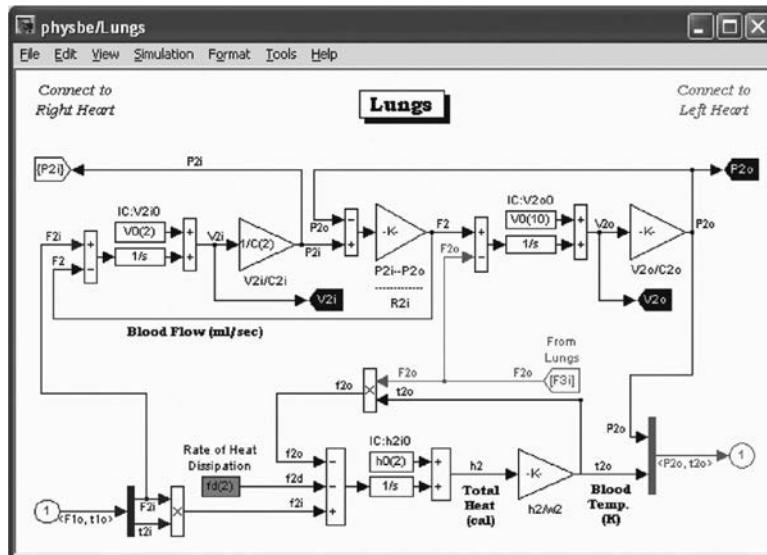


FIGURE 5.60 Subsystem model description of LUNGS.

5.6.2 CAR-FOLLOWING SUBSYSTEM

The next example of Simulink subsystems involves the dynamic behavior of a platoon of vehicles, that is, a lead car (platoon leader) followed by several vehicles whose motion is governed by the dynamics of the preceding vehicle. The Simulink diagram in Figure 5.42 was used to simulate car-following behavior. Deleting the “Scope” blocks, the “Clock” and “Lookup” blocks for generating the lead vehicle speed profile, and the integrator block for creating the lead vehicle position leaves the essential blocks for defining a Simulink car-following subsystem.

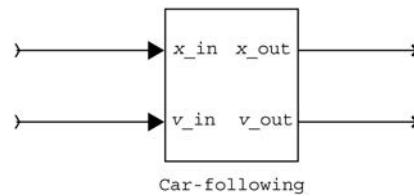


FIGURE 5.61 “Car-following” subsystem with lead vehicle inputs “ x_{in} ,” “ v_{in} ” and following vehicle outputs “ x_{out} ,” “ v_{out} .”

Input and output ports to the subsystem are created using Simulink “In” and “Out” blocks from the “Ports and Subsystems” sublibrary. A certain amount of discretion is possible when choosing subsystem inputs and outputs. For example, “In” blocks can be connected to the lead vehicle speed “ x_{0d} ” and position “ x_0 ,” while the following vehicle speed “ x_{1d} ” and position “ x_1 ” are selected as outputs by connecting them to “Out” blocks. Alternatively, the subsystem could be described in terms of a single input, namely, vehicle speed “ x_{0d} ,” and a single output such as vehicle acceleration “ x_{1dd} .”

A “car-following” subsystem is created by enclosing selected blocks in the Simulink diagram with a bounding box and choosing “Edit: Create Subsystem” from the menu. The selected blocks collapse into the “car-following” subsystem with renamed inputs and outputs as shown in Figure 5.61. Opening (double clicking) the subsystem reveals the underlying Simulink blocks that can be edited at any time.

The “car-following” subsystem constituent blocks are shown in Figure 5.62. Note that conversion factors from mph to fps (3600/5280) and vice versa were added (see “Gain” blocks in Figure 5.62) to maintain the vehicle speeds in and out of the subsystem in mph, while internal to the subsystem, vehicle speeds are in fps. Simulation of a platoon of vehicles is accomplished by repeated use of the “car-following” subsystem block.

Suppose we wish to simulate the dynamics of a five-vehicle platoon in response to a lead vehicle that decelerates and then accelerates back to a constant steady-state speed. In particular, our interest will focus on the induced perturbations in the stream of traffic. At the beginning of the simulation, each vehicle is traveling at the speed “ SL ,” separated in time by the desired gap G , as shown in Figure 5.63.

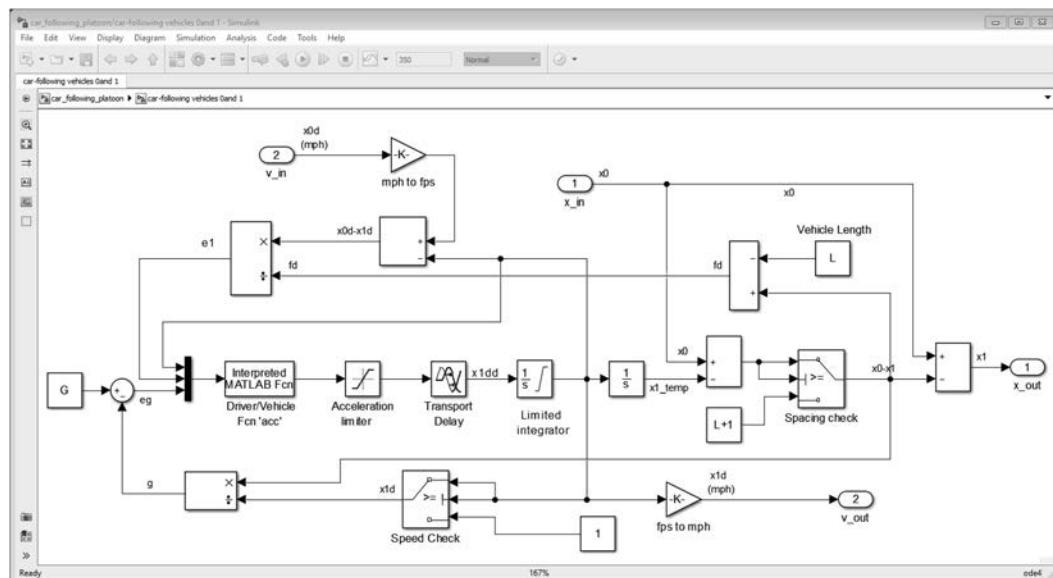


FIGURE 5.62 Simulink blocks comprising the car-following subsystem.

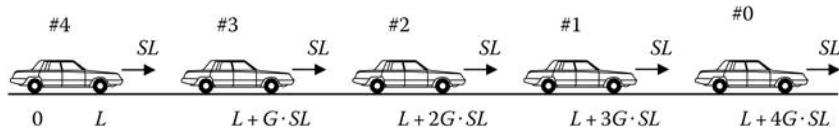


FIGURE 5.63 Initial conditions of the platoon vehicles.

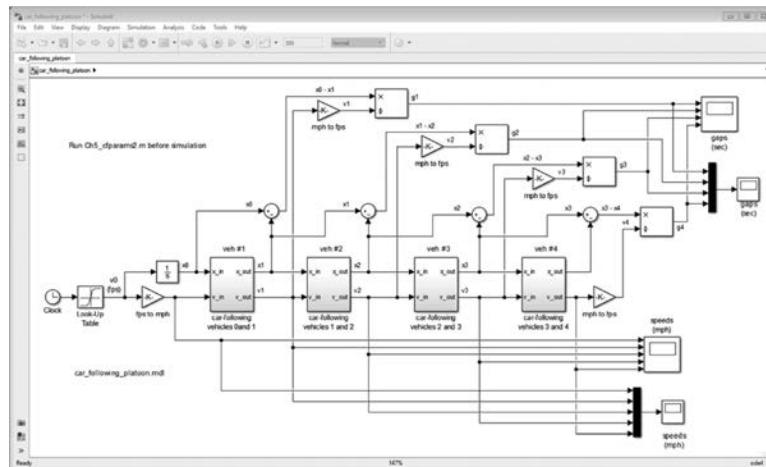


FIGURE 5.64 Simulink diagram with multiple instances of the “car-following” subsystem.

The file “*Ch5_cfparams2.m*” loads the parameters required by the Simulink subsystems and “*Lookup Table*” block for setting the lead vehicle speed. A top-level view of the model “*car _ following _ platoon.mdl*” is shown in Figure 5.64.

Initial conditions, namely, speeds and positions, of the trailing vehicles are set in the integrators of the appropriate subsystem blocks. The initial position of the lead vehicle is determined by the parameter of the integrator block feeding the first subsystem. The “*Mux*” block (lower right) multiplexes the five vehicle speeds on a single line for input to a “*Scope*” block that draws the five plots on a single set of axes. The heavy arrow emanating from the “*Mux*” indicates the presence of multiple signals.

Speeds of the lead vehicle and four following vehicles are shown in Figure 5.65. Figure 5.66 is a graph of the successive gaps between the vehicles of the platoon. The responses in Figures 5.65 and 5.66 indicate that the platoon achieves a new steady state identical to the initial one after the perturbations die out.

The “*Car-following*” subsystem can be daisy-chained as shown in Figure 5.64 to simulate the response of platoons of vehicles of any size. Furthermore, the following vehicles can be individualized by including a vector of randomly chosen driver/vehicle parameters as an additional input to each “*Car-following*” subsystem block.

5.6.3 SUBSYSTEM USING Fcn BLOCKS

The “*Fcn*” block is a convenient time saver when the mathematical model of the system consists primarily of algebraic and differential equations. Lengthy expressions are evaluated in equation form instead of being constructed from Simulink blocks. To illustrate, the frictionless inverted pendulum introduced in Section 5.4 can be treated as a subsystem with the governing equations for \ddot{x} and $\ddot{\theta}$ implemented by the use of “*Fcn*” blocks. Equations 5.50 and 5.51 are implicit in nature as a result of \ddot{x} and $\ddot{\theta}$ appearing in both equations. (Recall the presence of an algebraic loop in the

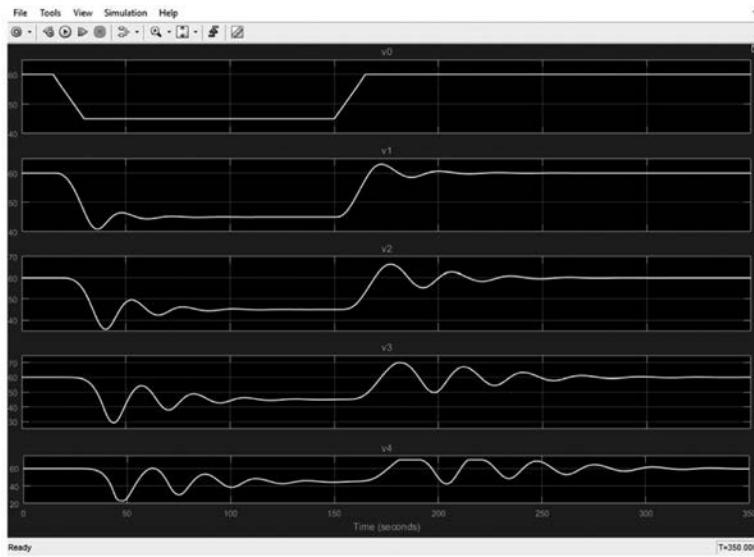


FIGURE 5.65 Speeds (mph) of platoon leader and following vehicles vs. time (s).

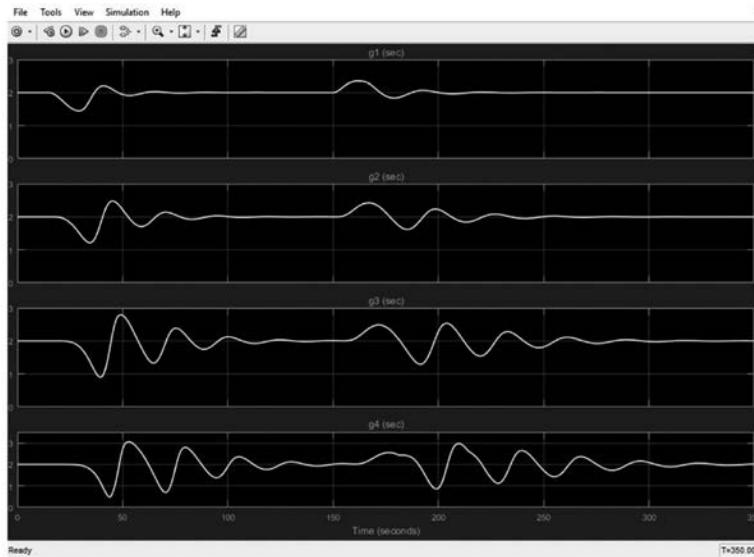


FIGURE 5.66 Gaps (s) of following vehicles vs. time (s).

Simulink diagram.) This can be overcome by solving for the second derivative terms explicitly leading to Equations 5.65 and 5.66.

$$\ddot{x} = \frac{ml\dot{\theta}^2 \sin \theta - mg \cos \theta \sin \theta + u}{M + m \sin^2 \theta} \quad (5.65)$$

$$\ddot{\theta} = \frac{-ml\dot{\theta}^2 \cos \theta \sin \theta + (m + M)g \sin \theta - u \cos \theta}{l(M + m \sin^2 \theta)} \quad (5.66)$$

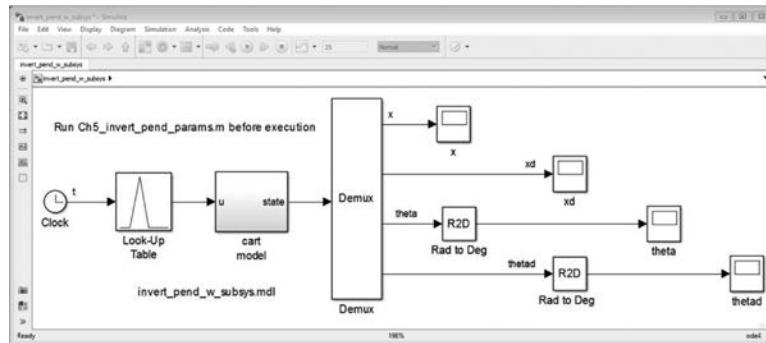


FIGURE 5.67 Top layer of a Simulink diagram for simulating an inverted pendulum.

Figure 5.67 shows the top layer with the “cart model” subsystem.

It includes Simulink blocks to generate the input u , decompose the state vector $[x; \dot{x}; \theta; \dot{\theta}]$ into its components, and feed the components to individual “scope” blocks. Note the use of the Simulink supplied “R2D” block for converting from radians to degrees. It is found in the “Simulink Extras” sublibrary under the “Transformations” heading. A number of useful coordinate transformation blocks are available there.

Opening the “cart model” subsystem reveals the blocks shown in Figure 5.68. Note that the “Display option” of the “Mux” parameter blocks is set to “signals” in order to identify its inputs. The parameters of the two “Fcn” blocks are expressions relating the accelerations “ xdd ” and “ $thetadd$ ” to the inputs “ x ,” “ \dot{x} ,” “ u ,” “ $thetad$,” and “ θ ” (from the “mux” block). The “Fcn” block input notation is $u[1], u[2], \dots, u[5]$ where $u[1]$ is the first input “ x ,” $u[2]$ is “ \dot{x} ,” and so forth.

From Equation 5.65, the “Fcn” block parameter expression for “ xdd ” is

$$\frac{(m_1 * u[4]^2 * \sin(u[5]) - m * g * \cos(u[5]) * \sin(u[5]) + u[3])}{(M + m * \sin(u[5]))^2}$$

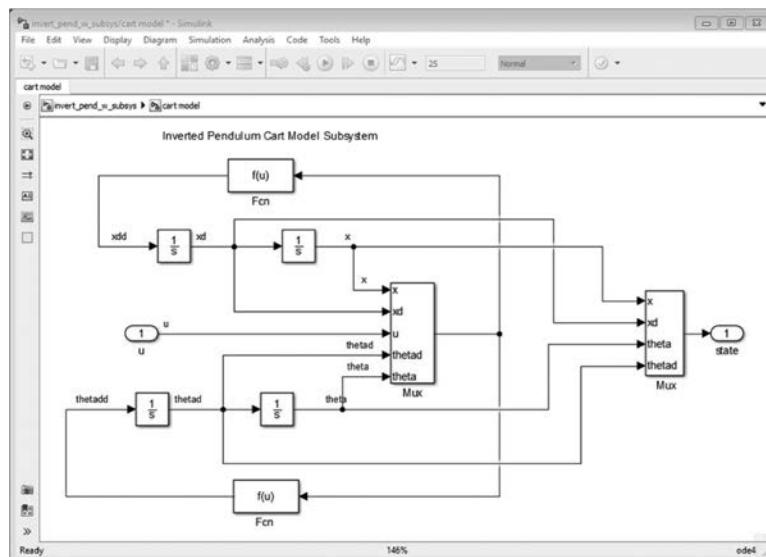


FIGURE 5.68 Cart subsystem using “Fcn” blocks.

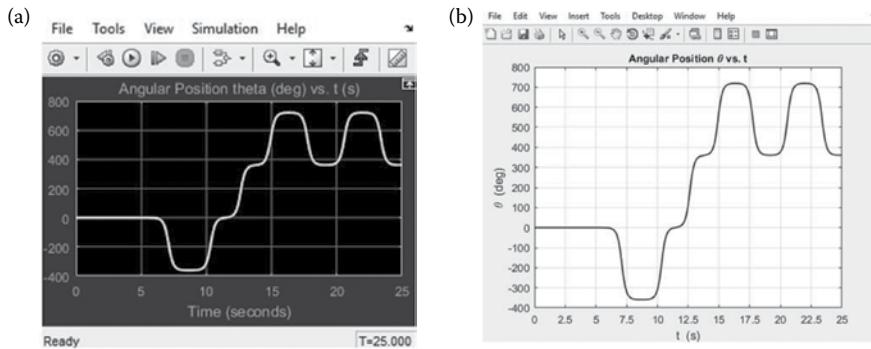


FIGURE 5.69 $\theta(t)$ vs. t for $u(t)$ shown in Figure 5.70 from (a) Simulink scope and (b) Matlab m-file “Ch5_invert_pend_params.m.”

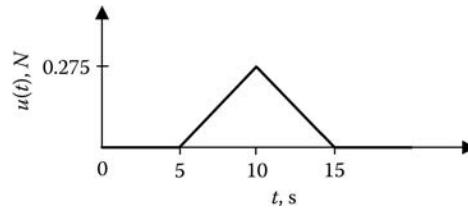


FIGURE 5.70 Force $u(t)$ applied to cart.

Referring to Equation 5.66, the “Fcn” block parameter expression for “thetadd” is

$$(-m*1*u[4]^2*cos(u[5])*sin(u[5]) + (m+M)*g*sin(u[5]) - u[3]*cos(u[5]) / (l*(M+m*sin(u[5])^2))$$

The angular position of the pendulum $\theta(t)$ is plotted in Figure 5.69 for the case when the cart and pendulum are initially at rest, that is, $x(0) = \dot{x}(0) = \theta(0) = \dot{\theta}(0) = 0$, and the input $u(t)$ is the triangular pulse shown in Figure 5.70. The numerical values of the system parameters are $M = 2$ kg, $m = 0.1$ kg, and $l = 0.5$ m.

EXERCISES

- 5.20 Randomize the car-following behavior by assuming that the desired gap G and driver/vehicle delay T are both normally distributed random variables, that is,

$$T \sim N(\mu_T, \sigma_T), \mu_T = 0.75 \text{ s}, \sigma_T = 0.15 \text{ s}$$

$$G \sim N(\mu_G, \sigma_G), \mu_G = 2.5 \text{ s}, \sigma_G = 0.3 \text{ s}$$

Use MATLAB to generate $\{G_i, T_i\}$, $i = 1, 2, 3, 4$ for four following vehicles, and repeat the simulation of the five vehicles shown in Figure 5.63.

- 5.21 Six vehicles are stopped at a traffic light with a distance of L feet from the rear bumper of the car in front to the front bumper of the following vehicle. The lead car accelerates uniformly from zero mph to the speed limit $SL = 45$ mph in 30 s and continues traveling at the speed limit. Use the robust car-following model developed in Exercise 5.16 to simulate the transient

- response of the platoon. Use the baseline conditions in the second column of [Table E5.14](#) for the parameters, or choose a new set of appropriate values. Obtain time history plots of
- Vehicle positions
 - Vehicle speeds
 - Vehicle gaps
 - Vehicle-following distances
- 5.22 A total of 11 cars are traveling at the speed limit SL with initial spacing similar to those in [Figure 5.63](#). At $t = 0$, the lead car speed begins to vary sinusoidally with amplitude of 3 mph and period of 20 s. Determine the peak amplitude in speed of the following vehicles for the nine combinations: $SL = 30, 45, 60$ mph and $G = 1.5, 2, 3$ s.
- 5.23 Starting with Equations 5.65 and 5.66 for the cart and inverted pendulum,
- Develop a state variable model of the system, that is, $\dot{\underline{x}} = f(\underline{x}, \underline{u})$ and $\underline{y} = g(\underline{x}, \underline{u})$ where the state $\underline{x} = [\underline{x}, \dot{\underline{x}}, \theta, \dot{\theta}]^T$ and output $\underline{y} = [\underline{x}, \theta]^T$.
 - Find the state equations for updating the discrete-time state $\underline{x}_A(n)$ and computing $\underline{y}_A(n)$ using forward Euler integration with step size T .
 - Solve the equations in part (b) recursively to find $\underline{x}_A(n)$ and $\underline{y}_A(n)$, $n = 1, 2, \dots, n_f$ where $T = 0.05$ s, $T_{\text{final}} - n_f T = 5$ s, $u(t) = 0$, $t \geq 0$, and $x(0) = [0, 0, \pi/6, 0]$.
 - Plot the discrete-time state vector $\underline{x}_A(n)$, $n = 0, 5, 10, \dots, n_f$
 - Simulate the response with Simulink for the same conditions in part (c) using the ode1 (Euler) integrator. Plot the state vector and compare the results to part (d).
- 5.24 Show that the frictionless cart and pendulum have two equilibrium points when the input $u(t) = 0$, $t \geq 0$, namely,

$$x_{1,e} = 0 \text{ m}, x_{2,e} = 0 \text{ m/s}, x_{3,e} = 0 \text{ rad}, x_{4,e} = 0 \text{ rad/s}$$

$$x_{1,e} = 0 \text{ m}, x_{2,e} = 0 \text{ m/s}, x_{3,e} = \pi \text{ rad}, x_{4,e} = 0 \text{ rad/s}$$

and verify by using Simulink that the first equilibrium point is unstable and the second one is stable. Is the second equilibrium point asymptotically stable?

- 5.25 Develop a subsystem model of the cart and pendulum where the pendulum rotation is opposed by a damping torque $T_D = c\dot{\theta}$ and the cart motion is subject to a constant friction force

$$f_\mu = \begin{cases} -\mu(m+M)g \cdot \text{sgn}(\dot{x}), & \dot{x} \neq 0 \\ 0, & \dot{x} = 0 \end{cases}$$

Simulate the response of the cart and pendulum, starting from the stable equilibrium point, to the input

$$u(t) = \begin{cases} 0, & 0 \leq t < 1 \\ U_0, & 1 \leq t < 3 \\ 0, & 3 \leq t \end{cases}$$

Numerical values of the system parameters are $m = 0.25$ kg, $M = 10$ kg, $l = 1$ m, $c = 0.5$ N · m/rad/s, $\mu = 0.015$, and $U_0 = 10$ N.

5.7 DISCRETE-TIME SYSTEMS

Up to this point, we have focused on using Simulink for simulation of systems with continuous-time mathematical models. Discrete-time systems evolve as approximate representations of continuous-time systems at specific points in time. The numerical integrators already considered as

well as those to come are predicated on some form of discrete-time approximation to the derivative function. Digital processors that manipulate streams of sampled numerical data are likewise discrete time in nature. Indeed, much of the first part of this book deals with methods for obtaining discrete-time model approximations of continuous-time systems. In the case of linear time-invariant (LTI) discrete-time systems, methods for finding solutions to specific inputs were presented as well.

Alternatively, some discrete-time systems process information, which by its very nature is allowed to change only at discrete instants of time. In that case, the systems are inherently discrete time. The difference equations are solved either recursively or by the use of a general solution (if one exists), resulting in an output sequence of numbers defined solely at discrete times 0, T , $2T$, $3T$,

Simulink is well suited to obtain solutions of discrete-time system models regardless of whether they are approximations of continuous-time systems or inherently discrete time to begin with. The procedure for obtaining a Simulink diagram of a discrete-time system is similar to the way Simulink diagrams of continuous-time systems were developed. With discrete-time systems, the goal is to express the highest order difference term as an explicit function of the lower order terms. For example, suppose an n th-order discrete-time system with output $y(k)$ and input $u(k)$ is modeled by the n th-order difference equation,

$$g[y(k+n), y(k+n-1), \dots, y(k+1), y(k), u(k+p), \dots, u(k+1), u(k)] = 0 \quad (5.67)$$

with initial conditions $y(0)$, $y(1)$, ..., $y(n - 1)$. Oftentimes it is possible to solve Equation 5.67 explicitly for $y(k + n)$, giving

$$y(k+n) = f[y(k+n-1), \dots, y(k+1), y(k), u(k+p), \dots, u(k+1), u(k)] \quad (5.68)$$

Starting with $y(k + n)$ and $u(k + p)$, delayed signals $y(k + n - 1)$, ..., $y(k + 1)$, $y(k)$ and $u(k + p - 1)$, ..., $u(k + 1)$, $u(k)$ are generated using the “Unit Delay” block and combined according to Equation 5.68 to complete the simulation diagram of the discrete-time system.

5.7.1 SIMULATION OF AN INHERENTLY DISCRETE-TIME SYSTEM

We begin with an inherently discrete-time system most of us are familiar with, namely, a fixed interest loan with constant periodic payments. An amount of money is borrowed for a specified period of time, and equally spaced installments are paid to the lender until the loan is completely repaid. The interest rate on the loan is established at the time of the loan. Furthermore, each payment consists of a portion that reduces the loan principal and the remaining portion that is interest on the outstanding balance. The situation is illustrated in [Figure 5.71](#).

The system parameters consist of

P_0 : Loan amount

i : Interest rate per period (fixed for the duration of the loan)

n : Number of interest periods for duration of loan

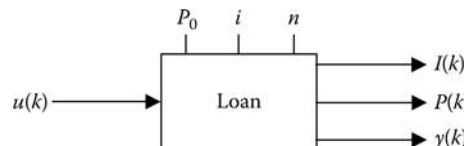


FIGURE 5.71 Repayment and amortization of a loan.

The discrete-time input

$$u(k) = \begin{cases} 0, & k = 0 \\ A, & k = 1, 2, 3, \dots, n \end{cases} \quad (5.69)$$

is the constant payment A made at the end of the k th interest period. The discrete-time outputs are

$y(k)$: Outstanding balance of loan immediately following the k th payment

$P(k)$: Portion of k th payment used to reduce the outstanding balance

$I(k)$: Interest portion of k th payment

The unpaid balance after the $(k + 1)$ st payment is simply the unpaid balance following the k th payment plus the interest accrued for one period on the unpaid balance minus the amount of the $(k + 1)$ st payment. Thus,

$$y(k+1) = y(k) + iy(k) - u(k+1), \quad k = 0, 1, 2, \dots, n-1 \quad (5.70)$$

$$= (1+i)y(k) - u(k+1), \quad k = 0, 1, 2, \dots, n-1 \quad (5.71)$$

$P(k + 1)$, the portion of $u(k + 1)$ used for loan principal reduction, is equal to the reduction in outstanding balance from the k th to the $(k + 1)$ st payment, that is,

$$P(k+1) = y(k) - y(k+1), \quad k = 0, 1, 2, \dots, n-1 \quad (5.72)$$

$I(k)$, the interest portion of $u(k)$, is obtained from

$$P(k) + I(k) = u(k) = A, \quad k = 1, 2, \dots, n \quad (5.73)$$

$$\Rightarrow I(k) = A - P(k), \quad k = 1, 2, \dots, n \quad (5.74)$$

It can be shown (Thuesen 1971) that the constant payment A necessary to fully repay the loan in n periods, that is, make $y(n) = 0$, is given by

$$A = P_0 \left[\frac{i(1+i)^n}{(1+i)^n - 1} \right] \quad (5.75)$$

Equations 5.71, 5.72, and 5.74 are the difference equations for the first-order discrete-time system in [Figure 5.71](#). A Simulink diagram of the system is shown in [Figure 5.72](#). Note the use of a single “Unit Delay” block to generate the signal $y(k)$ and the sum block in the upper right corner producing $y(k + 1)$ as the difference of $(1 + i)y(k)$ and the payment amount $u(k + 1)$ according to Equation 5.71.

The “Simulation Parameters” dialog box is shown in [Figure 5.73](#). A “Fixed-step” integrator with “Fixed-step size” of 1 is selected to force the simulation to step through integer values of discrete time. Since there is no continuous-time integration present in an inherently discrete-time system, the “discrete (no continuous states)” option is chosen from the drop-down menu of integrators.

The numerical values shown in [Figure 5.72](#) correspond to a \$125,000 loan at 8% interest per annum repaid over 30 years. The monthly payment A is calculated inside the “Fcns” block and appears in the “Display” as \$917.20. The unpaid balance $y(k)$ is shown in [Figure 5.74](#). As expected, the loan balance is zero following the 360th monthly payment.

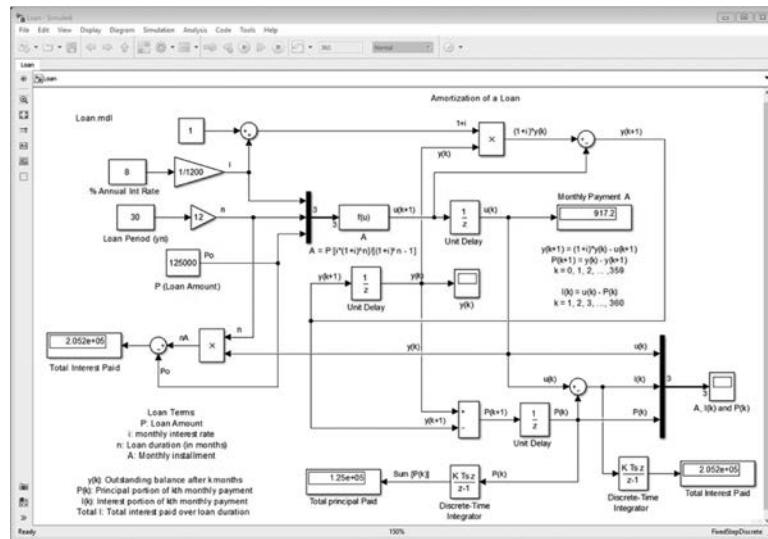


FIGURE 5.72 Simulink diagram for loan repayment.



FIGURE 5.73 Simulation parameters dialog box for loan simulation.

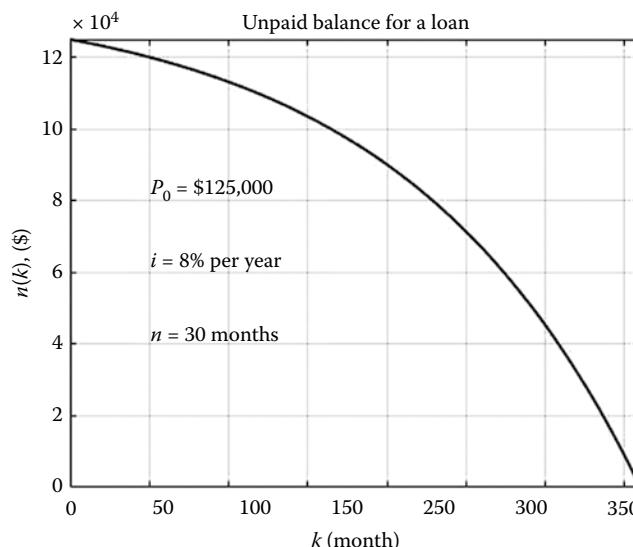


FIGURE 5.74 Unpaid balance $y(k)$ vs. interest period k .

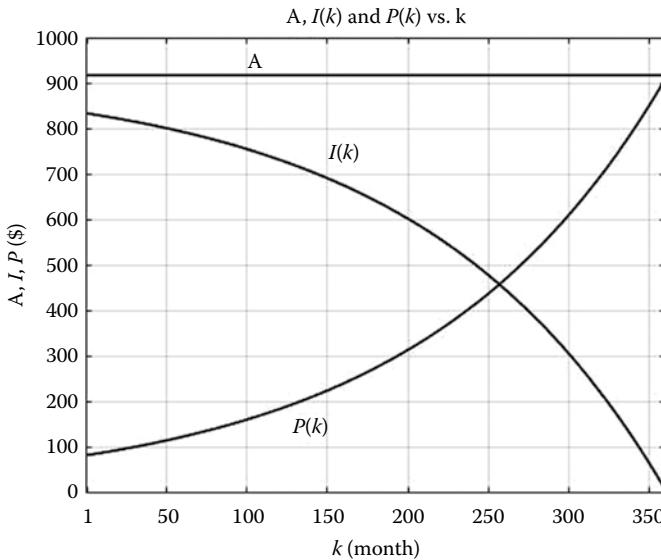


FIGURE 5.75 Monthly installment A , interest portion $I(k)$, and principal portion $P(k)$ vs. k .

The total monthly payment A , interest portion $I(k)$, and principal portion $P(k)$ are shown in Figure 5.75. Note that the early payments consist almost entirely of interest with only a small amount going towards principal reduction. As the loan progresses, the portion of each monthly installment used to reduce the outstanding balance increases. Conversely, the interest portion of each subsequent payment is less than the previous one.

The total interest paid over the life of the loan is computed in two different ways. The simplest approach is to compute $n_A - P_0$, the result shown in the “Display” block on the left side of the Simulink diagram. The second method employs a “Discrete-Time Integrator” situated in the lower right corner of Figure 5.72.

5.7.2 DISCRETE-TIME INTEGRATOR

A “Discrete-Time Integrator” is a numerical integrator from the “Discrete” sublibrary reserved for discrete-time systems. The “Discrete-Time Integrator” can be configured as a forward (explicit) Euler, backward (implicit) Euler, or trapezoidal integrator. The z-domain transfer functions derived in Section 4.7 for each of the discrete-time integrators are placed inside the appropriate block shown in Figure 5.76.

The two discrete-time integrators in Figure 5.72 function as summing devices, one for the total interest and the other for the computation of the total of all the principal payments. To understand

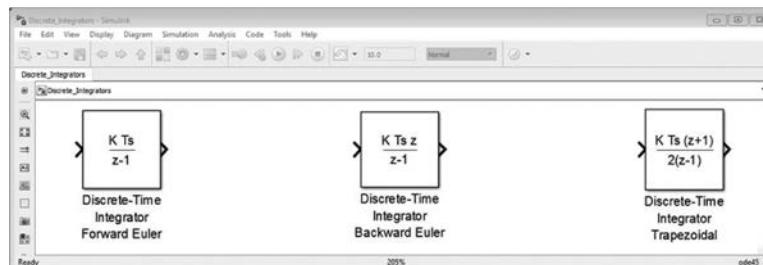


FIGURE 5.76 Simulink discrete-time integrators.

why, consider a discrete-time backward (implicit) Euler integrator with input signal $I(k)$ and output $I_T(k)$. The difference equation is

$$I_T(k+1) = I_T(k) + T \cdot I(k+1) \quad (5.76)$$

With $T = 1$ and $I_T(0) = 0$, it follows that $I_T(1) = I(1)$, $I_T(2) = I(1) + I(2)$, ... and

$$I_T(n) = \sum_{k=1}^n I(k) \quad (5.77)$$

Simulink's fixed-step integrators "ode1" (Euler), "ode2" (Heun) through "ode5" are all explicit. Thus, if we elect to use a fixed-step implicit integrator, our choice is limited to either "Backward Euler" or "Trapezoidal" from the "Discrete" sublibrary. In this case, the Simulink diagram is similar to the simulation diagram representation of the continuous-time system with the exception that the continuous-time integrators (1/s blocks) are replaced by the preferred implicit discrete-time integrator.

To illustrate, consider the vehicle of weight W lb in [Figure 5.77](#) rolling backwards down an incline (θ) subject to an aerodynamic drag force F_D , a rolling friction force F_μ ($F_\mu/4$ on each tire), and a gravitational component of weight F_W . The vehicle travels a length L along the inclined section and then continues on a level section of road.

Summing the forces on the vehicle in the direction of travel gives

$$m\ddot{x} = -F_D - F_\mu + F_W \quad (5.78)$$

$$= \begin{cases} -0.5C_D\rho A\dot{x}^2 - \mu W \cos\theta + W \sin\theta, & x \leq L \\ -0.5C_D\rho A\dot{x}^2 - \mu W, & x > L \end{cases} \quad (5.79)$$

where

m is the mass of the vehicle

C_D is the aerodynamic drag coefficient

ρ is the density of air

A is the exposed vehicle front area

μ is the coefficient of rolling friction between the tires and the road

A Simulink diagram using trapezoidal integration for both integrators is shown in [Figure 5.78](#). The first is a limited integrator with the lower limit set to zero. The "Switch" block guarantees that the friction force is zero when the vehicle is stopped.

Numerical values of the system parameters are shown in the "Con" and "Gain" blocks except for $L = 200$ ft and $\theta = 10^\circ$, which appear in the "Lookup table" parameters. Results are shown

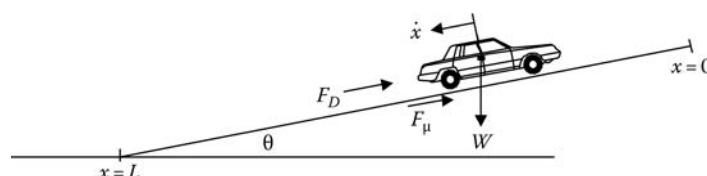


FIGURE 5.77 Vehicle rolling down an incline.

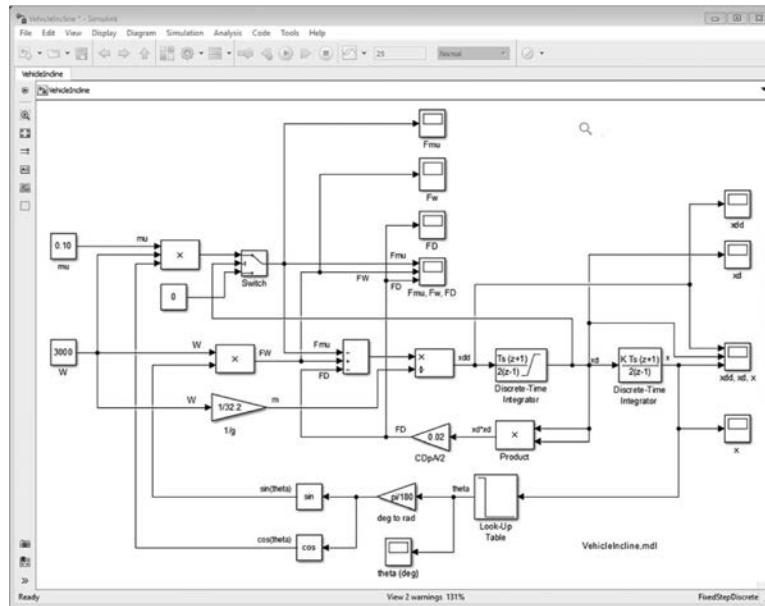


FIGURE 5.78 Simulink diagram for vehicle rolling down incline.

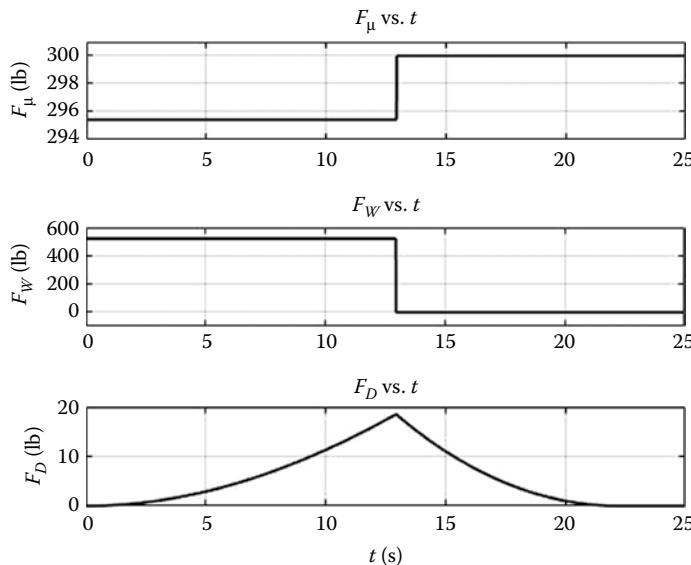


FIGURE 5.79 F_μ , F_W , F_D vs. t .

in Figures 5.79 through 5.81 for the case when the vehicle is released from a stopped position and starts rolling down the incline.

Implicit integrators like the “Backward Euler” and “Trapezoidal” may lead to algebraic loops, which require additional computational effort to resolve at each discrete-time step. In fact, an “Algebraic Loop Warning” appears in executing the simulation corresponding to the Simulink diagram in Figure 5.78. Algebraic loops never include a continuous-time integrator because all of Simulink’s continuous-time integrators are implemented by explicit numerical integration algorithms.

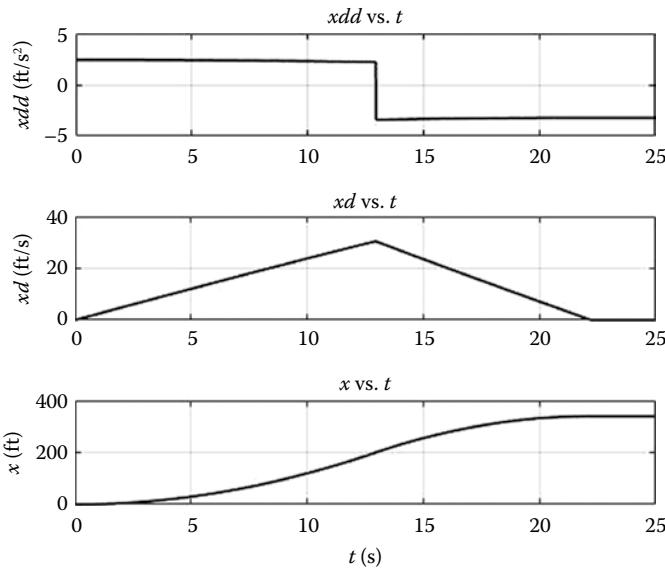


FIGURE 5.80 \ddot{x}_n , \dot{x}_n , x vs. t .

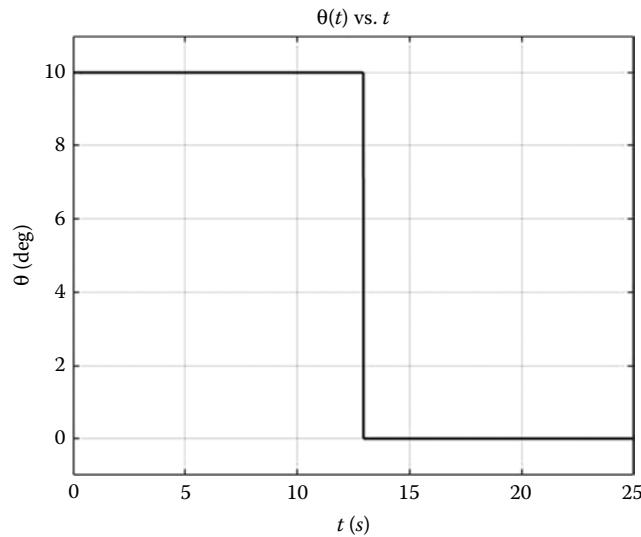


FIGURE 5.81 θ vs. t .

Typical of all Simulink blocks in the “Discrete” sublibrary, the discrete-time integrator outputs are clamped or held constant for the duration of the sample time (integration step), 0.01 s, in this example (see [Figure 5.82](#)).

5.7.3 CENTRALIZED INTEGRATION

The simulation diagram of an n th-order continuous-time system model will contain n distinct integrators. The Simulink diagram will contain one integrator for each continuous-time state or equivalently a “State-Space,” “Transfer Fcn,” or “Zero-Pole” block from the “Continuous” sublibrary to model components with one or more continuous-time states. In either

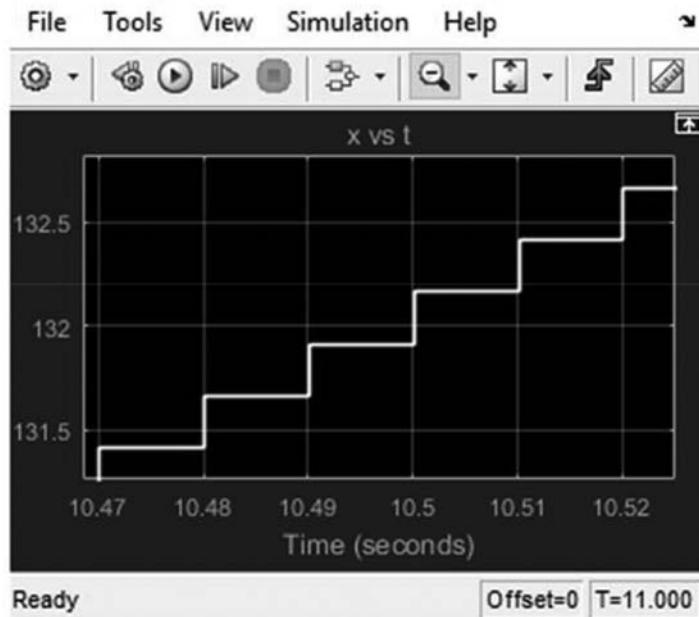


FIGURE 5.82 Close-up of discrete-time integrator output illustrating discrete-time nature (sample time equal 0.01 s).

case, the numerical integrator selected from the choice of fixed-step and variable-step integrators in the “Simulation Parameters” dialog box will be applied to all the continuous-time state derivatives.

For “One-Step” numerical integration algorithms (discussed in Chapter 6), which includes the explicit Euler integrator, all state derivatives are calculated at $t_n = nT$ prior to updating a single state at $t_{n+1} = (n + 1)T$. The entire collection of continuous-time states are updated at t_{n+1} based on the state derivatives at t_n , which in turn depend on the values of the states and inputs (if present) at $t_n = nT$. This is referred to as centralized integration.

There are situations when centralized integration is not the most advantageous approach when it comes to updating the states. For example, when the state derivatives are themselves states, some of the states can be updated at time $t_{n+1} = (n + 1)T$ based on calculated values of other states at t_{n+1} .

Figure 5.83 is a simulation diagram of a mechanical system with inertia. It contains an acceleration term that is twice integrated to produce velocity and position.

With centralized explicit Euler integration, the discrete-time states $x_{1,A}(n)$ and $x_{2,A}(n)$ are updated at time $t_{n+1} = (n + 1)T$ in the sequence of steps as follows:

$$\dot{x}_2(n) = \ddot{x}(n) = \frac{1}{M} [F(n) - Kx_{1,A}(n) - Bx_{2,A}(n)] \quad (5.80)$$

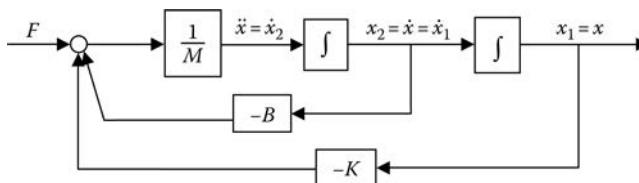


FIGURE 5.83 Simulation diagram of second-order system with sequential integrators.

$$\dot{x}_1(n) = x_{2,A}(n) \quad (5.81)$$

$$x_{2,A}(n+1) = x_{2,A}(n)T\dot{x}_2(n) \quad (5.82)$$

$$\Rightarrow x_{2,A}(n+1) = x_{2,A}(n) + \frac{T}{M} [F(n) - Kx_{1,A}(n) - Bx_{2,A}(n)] \quad (5.83)$$

$$x_{1,A}(n+1) = x_{1,A}(n) + T\dot{x}_1(n) \quad (5.84)$$

$$\Rightarrow x_{1,A}(n+1) = x_{1,A}(n) + Tx_{2,A}(n) \quad (5.85)$$

Instead of starting with Equation 5.84 to update $x_{1,A}(n)$, the implicit Euler form can be used, that is, the updated state $x_{1,A}(n+1)$ is obtained from

$$x_{1,A}(n+1) = x_{1,A}(n) + T\dot{x}_1(n+1) \quad (5.86)$$

$$\Rightarrow x_{1,A}(n+1) = x_{1,A}(n) + Tx_{2,A}(n+1) \quad (5.87)$$

Equations 5.83 and 5.87 form the basis of a noncentralized integration scheme, which uses explicit Euler integration to update $x_{2,A}(n)$ and implicit Euler integration to update $x_{1,A}(n)$ at $t_{n+1} = (n+1)T$. The explicit Euler/implicit Euler combination is superior to explicit Euler/explicit Euler integration by virtue of its using the updated velocity $x_{2,A}(n+1)$ in the computation for the new state $x_{1,A}(n+1)$ in Equation 5.87.

With Simulink, explicit Euler/implicit Euler (or explicit Euler/trapezoidal) integration of a second-order component is straightforward. [Figure 5.84](#) is a Simulink diagram for simulating the unit step response of the second-order system

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = \omega_n^2 u \quad (5.88)$$

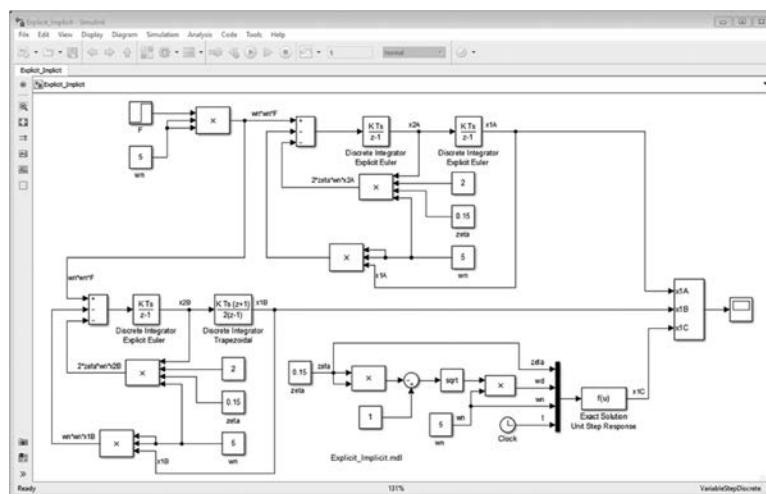


FIGURE 5.84 Simulink diagram with explicit Euler/explicit Euler integration, explicit Euler/trapezoidal integration and exact solution of second-order system unit step response ($\zeta = 0.15$, $\omega_n = 5$ rad/s), sample time $T = 0.02$ s.

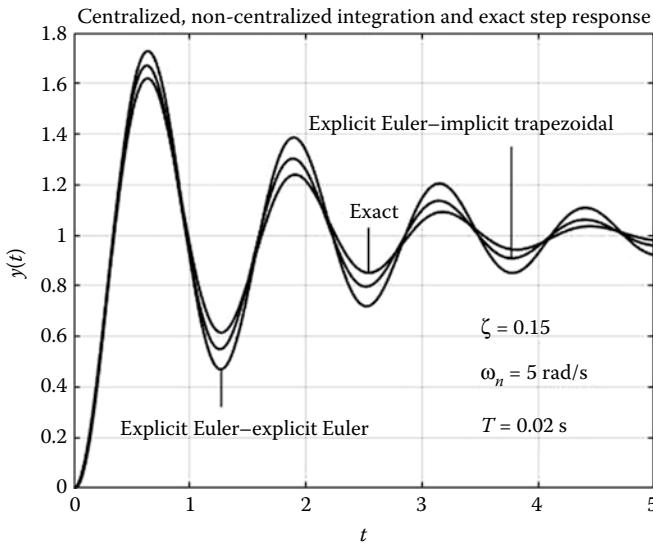


FIGURE 5.85 Results of centralized integration and noncentralized integration.

The explicit Euler/explicit Euler and explicit Euler/trapezoidal integration routines are implemented, and the exact solution is generated for comparison. The lightly damped system step responses are shown in Figure 5.85. The response obtained using noncentralized integration (explicit Euler/trapezoidal) is closer to the exact solution.

5.7.4 DIGITAL FILTERS

Digital filters were introduced in this chapter. A digital filter is a discrete-time system designed to process discrete-time data for the purpose of extracting useful information. In many cases, the data is comprised of a useful signal and an unwanted component such as noise. When the frequency components in the signal and noise are confined to distinct regions in the frequency spectrum, a properly designed digital filter can remove a significant portion of the noise without appreciable degradation of the signal component. The following example (Cadzow 1973) illustrates a notch filter designed to remove 60 Hz ($\omega_0 = 2\pi \times 60 \text{ rad/s}$) noise from a signal.

The sixth-order digital filter with input $u(k)$ and output $y(k)$ is represented as a cascaded system of second-order filters governed by the following difference equations:

$$y_1(k) = u(k) + b_1 u(k-1) + u(k-2) - a_1 y_1(k-1) - a_2 y_1(k-2) \quad (5.89)$$

$$y_2(k) = y_1(k) + b_1 y_1(k-1) + y_1(k-2) - a_3 y_2(k-1) - a_4 y_2(k-2) \quad (5.90)$$

$$y_3(k) = y_2(k) + b_1 y_2(k-1) + y_2(k-2) - a_5 y_3(k-1) - a_6 y_3(k-2) \quad (5.91)$$

$$y(k) = b y_3(k) \quad (5.92)$$

Parameters a_1, a_2, \dots, a_6 and b_1 influence the location of the six poles and two zeros of the filter's z-domain transfer function $H(z)$. The first requirement is for the magnitude function at the noise frequency, $|H(e^{j\omega_0 T})| = 0$.

The constant b is selected to make the magnitude function approximately one at other frequencies. Numerical values of the filter's constants are listed in the Simulink diagram shown in Figure 5.86. A sampling period of $T = 0.001 \text{ s}$ is used.

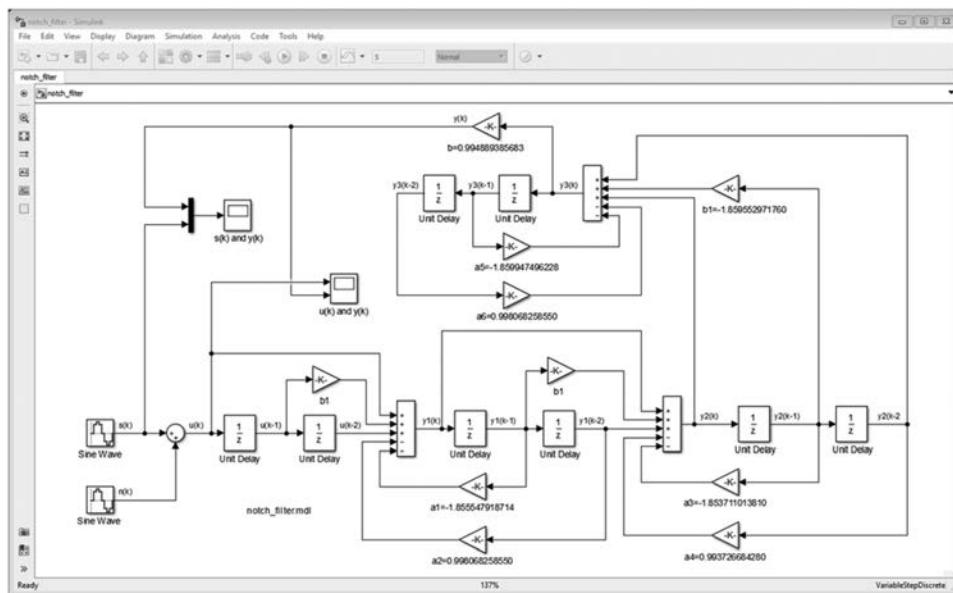


FIGURE 5.86 Simulink diagram for a notch filter (notch frequency = 60 Hz).

The continuous-time signal $u(t)$ consists of a sinusoidal component $s(t) = 2.5 \sin \omega t$ and 60 Hz noise component $n(t) = \sin 120\pi t$. Figure 5.87 shows the sampled input signal $u(k)$ with a low frequency 5 Hz ($\omega = 10\pi$ rad/s) signal component. Also shown is the filter output $y(k)$ at steady state following the transient period. Figure 5.88 shows that the signal component $s(k)$ and the discrete-time filter output $y(k)$ are nearly identical.

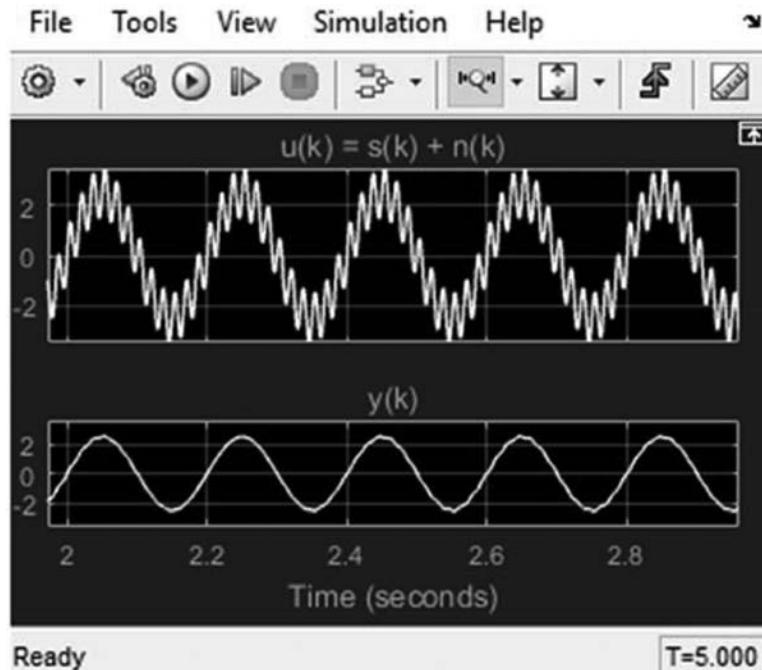


FIGURE 5.87 $u(k), y(k), \omega = 10\pi$ rad/s.

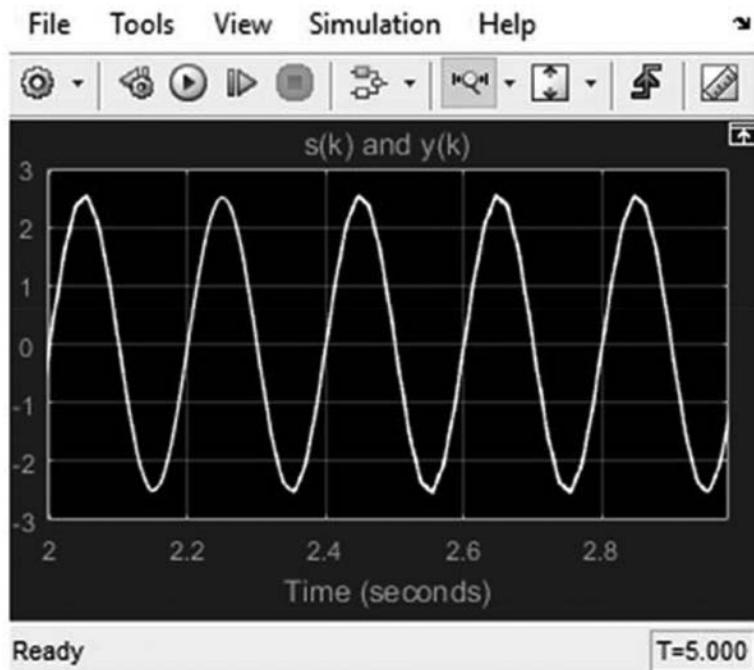


FIGURE 5.88 $s(k), y(k)$, $\omega = 10\pi$ rad/s.

Figures 5.89 and 5.90 show the steady-state response of the filter when the frequency of the signal component is closer to the noise frequency, that is, 57.5 Hz ($\omega = 115\pi$ rad/s).

The reader should try running the simulation for the case when the signal frequency is higher than the notch frequency (but less than the Nyquist frequency $\pi/T = 1000\pi$ rad/s) to verify similar results.

5.7.5 DISCRETE-TIME TRANSFER FUNCTION

Simulink is capable of simulating the response of linear discrete-time systems based on the knowledge of the system's discrete-time transfer function. Similar to continuous-time systems with transfer function $H(s)$ in pole-zero form or a ratio of polynomials, the response of a discrete-time system with transfer function $H(z)$ to a discrete-time input is obtained by using one of the blocks shown in Figure 5.91.

The “Discrete Filter” block is used primarily for digital filters where the numerator and denominator are polynomials in z^{-1} . It is easily obtained from either of the other two forms by multiplying numerator and denominator by z^{-n} where n is the order of the denominator polynomial. The following example illustrates the use of the “Discrete Zero-Pole” block for a low-pass digital filter obtained by approximating the dynamics of a continuous-time filter.

The *RC* circuit in Figure 5.92 was shown to exhibit the characteristics of a low-pass filter in Chapter 4.

The continuous-time transfer function $H(s)$ is given by

$$H(s) = \frac{K\omega_c}{s + \omega_c} \quad (5.93)$$

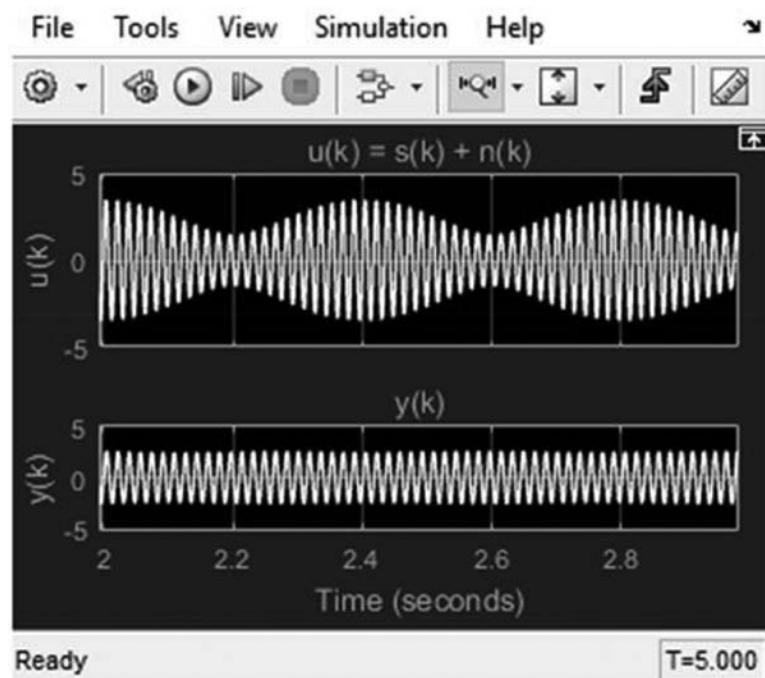


FIGURE 5.89 $u(k), y(k), \omega = 115\pi$ rad/s.

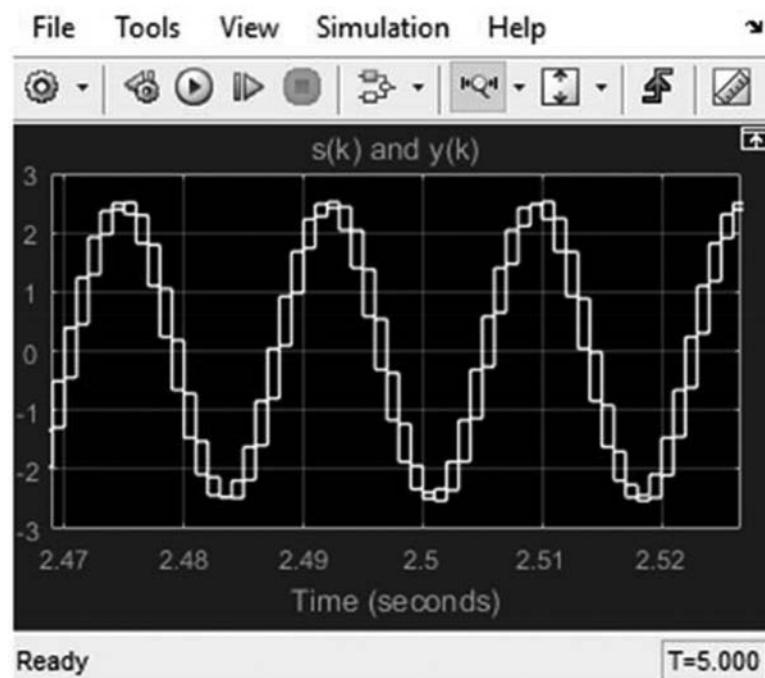


FIGURE 5.90 $s(k), y(k), \omega = 115\pi$ rad/s.

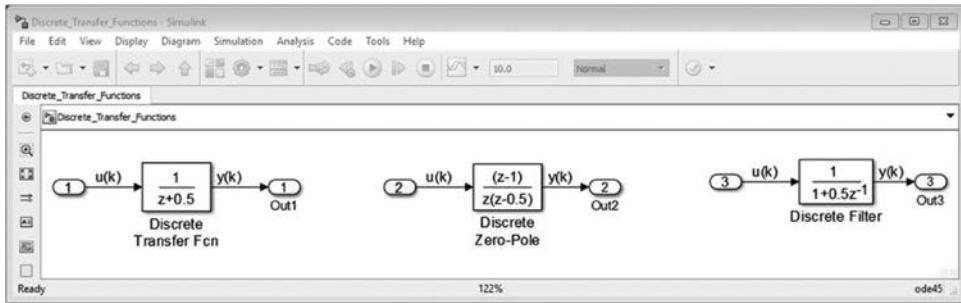


FIGURE 5.91 Simulink discrete-time transfer function blocks.

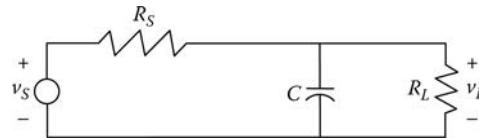


FIGURE 5.92 Circuit for a low-pass filter.

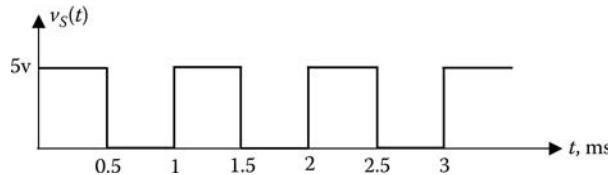


FIGURE 5.93 Input signal $v_s(t)$, $t \geq 0$.

where ω_c , the cutoff frequency, and DC gain K are related to circuit parameters R_S , R_L , and C according to

$$\omega_c = \frac{(1/R_S) + (1/R_L)}{C} \quad (5.94)$$

$$K = \frac{R_L}{R_S + R_L} \quad (5.95)$$

Suppose the signal $v_s(t)$ is the square wave shown in Figure 5.93.

The capacitor in the circuit of Figure 5.92 removes high-frequency components from $v_s(t)$, resulting in smoother pulse transitions in $v_L(t)$.

For values of $R_S = 50 \Omega$ and $R_L = 200 \Omega$, the capacitance C is selected to make the cutoff frequency 5 kHz ($\omega_c = 5 \times 10^3 \times 2\pi \text{ rad/s}$). The initial design calls for the Nyquist frequency π/T to be twice the cutoff frequency making the sample time $T = 50 \mu\text{s}$ (20 kHz sampling rate).

The discrete-time filter transfer function $H(z)$ is to be synthesized from the continuous-time filter transfer function $H(s)$ using the bilinear transform (see Chapter 4).

$$H(z) = H(s) |_{s=(2/T)((z-1)/(z+1))} \quad (5.96)$$

$$= \frac{K\omega_c}{(2/T)((z-1)/(z+1)) + \omega_c} \quad (5.97)$$

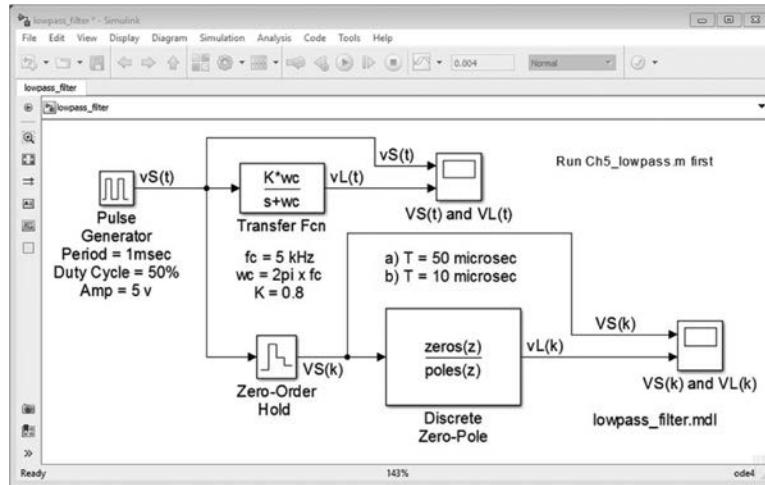


FIGURE 5.94 Simulink diagram for continuous- and discrete-time low-pass filters.

$$\Rightarrow H(z) = \frac{K\omega_c T}{2 + \omega_c T} \left[\frac{z + 1}{z - ((2 - \omega_c T)/(2 + \omega_c T))} \right] \quad (5.98)$$

Figure 5.94 shows the Simulink diagram for simulating the response of the continuous-time filter when the input is $v_s(t)$, $t \geq 0$ and the digital filter when the discrete-time input is $v_s(kT)$, $k = 0, 1, 2, \dots$. The “Zero-Order Hold” block is required when the discrete-time system sample time T exceeds the integration step size (10 μ s) for the continuous-time “Transfer Fcn” block. The continuous-time signals are shown in Figure 5.95.

The discrete-time signals are shown in Figure 5.96 ($T = 50 \mu$ s) and Figure 5.97 ($T = 10 \mu$ s). As expected, the digital filter response $v_L(K)$ is closer to the output $v_L(t)$ of the analog circuit at the higher sampling rate.

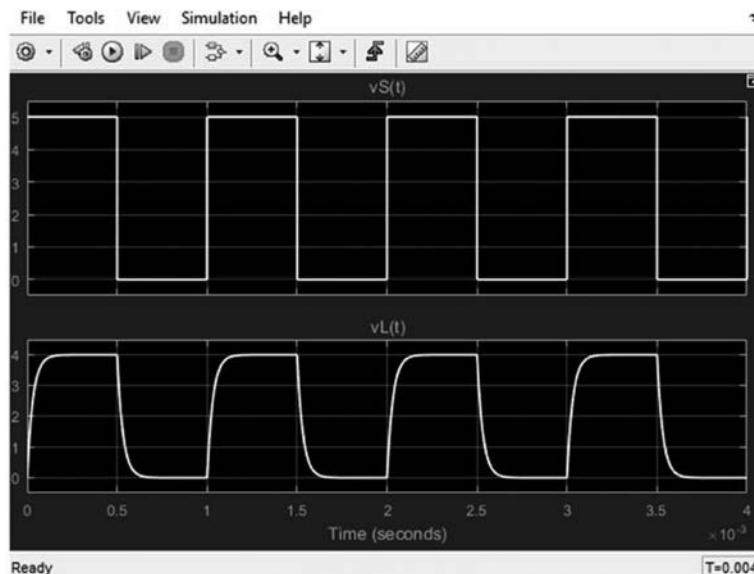


FIGURE 5.95 Low-pass continuous-time filter input and output.

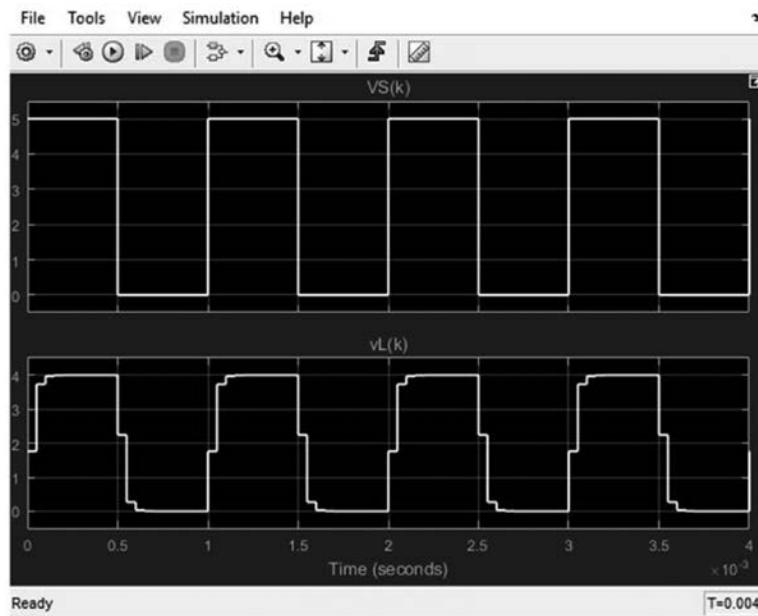


FIGURE 5.96 Digital filter input and output ($T = 50 \mu\text{s}$).

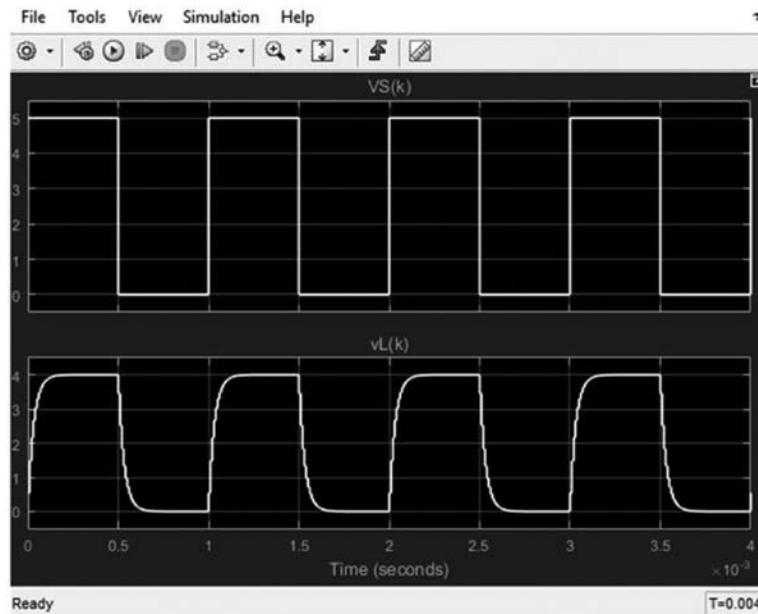


FIGURE 5.97 Digital filter input and output ($T = 10 \mu\text{s}$).

EXERCISES

- 5.26 A car loan in the amount of \$25,000 is to be paid off in 5 years with an annual interest rate of 8%. Use the Simulink loan simulation to find
- The monthly installment
 - The unpaid balance after the 30th payment
 - The principal portion of the 12th payment

- d. The total interest paid over the life of the loan
e. The time required for the unpaid balance to equal \$12,500
- 5.27 A prospective home buyer is considering purchasing a \$200,000 house with a 10% down payment and financing the balance over 30 years. He is able to afford monthly payments of \$1,450.
- What is the maximum interest rate per annum on the mortgage for which the house is affordable?
 - Repeat part (a) for a 15 year mortgage.
- 5.28 A college savings account is created on January 1, 2000 with a deposit of \$1000. The account earns 4% per year. End-of-month deposits in the amount of \$150 are made for a period of 18 years with the last deposit scheduled for December 31, 2017.
- Write a difference equation for $y(k)$, $k = 1, 2, 3, \dots$ the account balance after the k th deposit. Note that $y(0) = 1000$ and $u(k) = 150$, $k = 1, 2, 3, \dots, 216$.
 - Find the account balance after the last deposit.
- 5.29 Numerical differentiation is a procedure for approximating the derivatives of a mathematical function based on sampled values from it. The following backward difference formulas estimate the first three derivatives of a signal $f(t)$ at time t_i :

$$f'(t_i) = \frac{1}{2T} [f(t_i - 2T) - 4f(t_i - T) + 3f(t_i)]$$

$$f''(t_i) = \frac{1}{T^2} [-f(t_i - 3T) + 4f(t_i - 2T) - 5f(t_i - T) + 2f(t_i)]$$

$$f'''(t_i) = \frac{1}{2T^2} [3f(t_i - 4T) - 14f(t_i - 3T) + 24f(t_i - 2T) - 18f(t_i - T) + 5f(t_i)]$$

Develop a Simulink program to approximate

- $f'(t_i)$, $t_i = 2T, 3T, 4T, \dots$ given $f(0), f(T)$
- $f''(t_i)$, $t_i = 3T, 4T, 5T, \dots$ given $f(0), f(T), f(2T)$
- $f''(t_i)$, $t_i = 4T, 5T, 6T, \dots$ given $f(0), f(T), f(2T), f(3T)$ where the function $f(t)$ is obtained from an “Fcni” block as shown in [Figure E5.29](#):

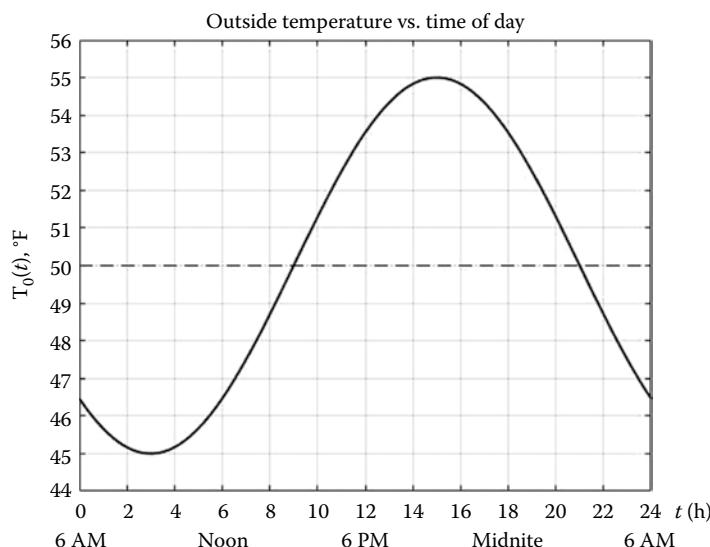


FIGURE E5.29

- d. Run the simulations for approximating the derivatives for the following cases:

$$f(t) = A \sin \omega t, \quad A = 1, \quad \omega = 2\pi, \quad T = 0.01 \frac{2\pi}{\omega} \quad (\text{i})$$

$$f(t) = K e^{-t/\tau}, \quad K = 10, \quad \tau = 1, \quad T = 0.01\tau \quad (\text{ii})$$

$$f(t) = K \left[1 - \frac{\omega_n}{\omega_d} e^{-\zeta \omega_n t} \sin(\omega_d t + \varphi) \right], \quad K = 1, \quad \zeta = 0.25, \quad \omega_n = 4, \quad T = 0.01 \left(\frac{1}{\zeta \omega_n} \right) \quad (\text{iii})$$

where

$$\begin{aligned} \omega_d &= \sqrt{1 - \zeta^2 \omega_n} \\ \varphi &= \tan^{-1}(\omega_d / \zeta \omega_n) \end{aligned}$$

Run the simulations for a period of time sufficient to include two cycles of the sine function and the transient periods of the second and third functions.

- e. Compare the approximate and exact values of $f'(t_i)$, $f''(t_i)$, and $f'''(t_i)$ for each function at 10 equally spaced points.

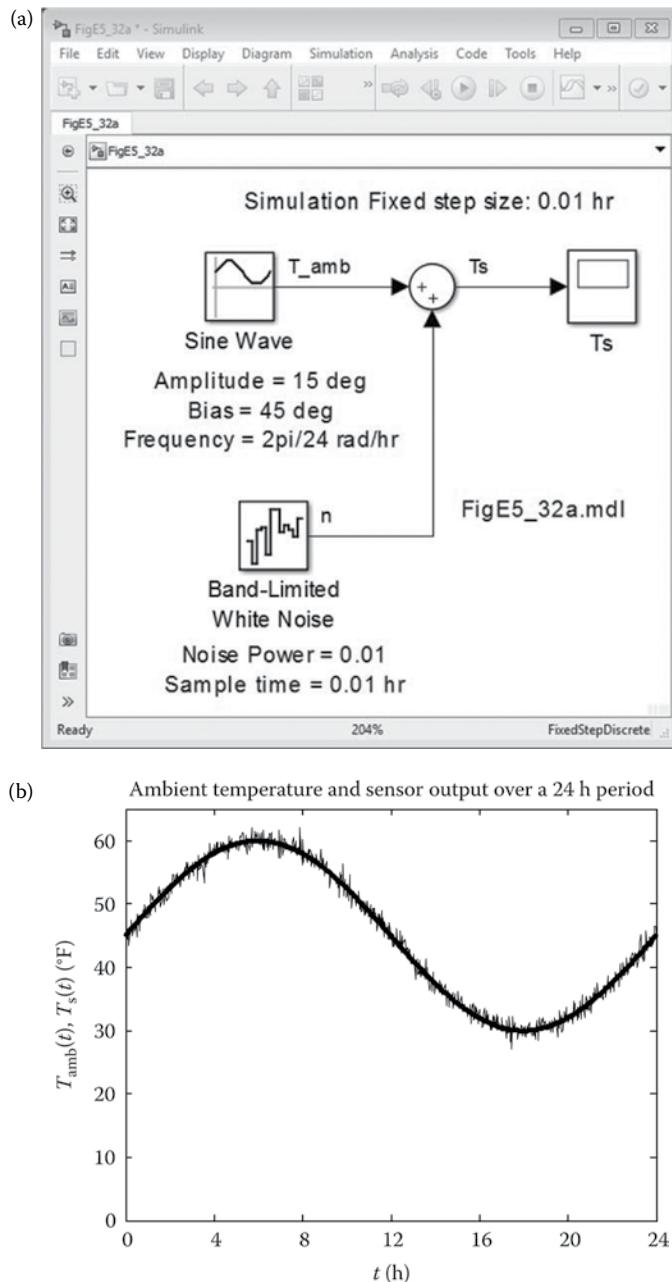
5.30 A second-order system is governed by the differential equation

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = K\omega_n^2 u(t) + b_1\dot{u}(t) + b_2\ddot{u}(t)$$

- a. Draw a simulation diagram of the system and label the states x_1 and x_2 .
- b. Draw a Simulink diagram using “Forward Euler” and “Trapezoidal” discrete-time integrators to calculate $x_{2,A}(n)$ and $x_{1,A}(n)$, respectively.
- c. Use the values $\zeta = 0.5$, $\omega_n = 4$, $K = 2$, $b_1 = 0$, and $b_2 = 1$ to simulate $y(t)$ in response to the input

$$u(t) = \begin{cases} 9 - t^2, & 0 \leq t < 3 \\ 0, & t \geq 3 \end{cases}$$

- with $y(0) = \dot{y}(0) = 0$.
- d. Find the analytical solution for $y(t)$. Use an “Fcrl” block in the Simulink diagram with input t and output the analytical solution for $y(t)$. Compare the simulated and analytical solutions.
- 5.31 For the notch filter given by Equations 5.89 through 5.92, find an expression for the constant b in Equation 5.92 if the DC gain of the filter is one.
- 5.32 An electronic sensor is used to measure ambient temperature $T_{\text{amb}}(t)$, which varies over a 24 h period in sinusoidal fashion about 45°F with amplitude of 15°F (Figure E5.32a). The sensor output $T_s(t)$ is corrupted with an additive white noise component (power = 0.01). A Simulink diagram for generating $T_{\text{amb}}(t)$ and $T_s(t)$ is shown in Figure E5.32b, along with time histories of each signal.

**FIGURE E5.32**

A simple low-pass digital filter is required to smooth the output signal without seriously degrading the signal component. The first-order filter difference equation is

$$T_f(k+1) = (1 - \alpha)T_s(k+1) + \alpha T_f(k), \quad k = 0, 1, 2, \dots$$

where

- $T_s(k)$ is the sampled output of the sensor
- $T_f(k)$ is the filter output

Note that the initial filter output $T_f(0) = T_{\text{amb}}(0) = 45^\circ\text{F}$.

The filter sample time is $T = 0.01 \text{ h}$. The parameter α is related to the bandwidth of the filter ω_0 by

$$\alpha = 2 - \cos \omega_0 T - \sqrt{(3 - \cos \omega_0 T)(1 - \cos \omega_0 T)}$$

Choose ω_0 as the frequency of the ambient temperature (in rad/h), and simulate the filter's response over a 24 h period. Plot $T_{\text{amb}}(t)$, $T_s(t)$, and $T_f(t)$ on the same graph.

- 5.33 Differentiation of noisy analog signals in continuous-time systems is often accomplished by first removing the high-frequency components. To illustrate, suppose the signal $x(t) = s(t) + n(t)$ where $s(t) = 2t$, $t \geq 0$, and $n(t) = 0.25 \sin 120\pi t$ is fed to a differentiator as shown in [Figure E5.33](#). A series of low-pass filters with $H(z) = (1 - a)z/(z - a)$ like the ones shown in [Figure E5.33](#) are inserted between the signal $x(t)$ and another differentiator. Vary the number of low-pass filters and the constant "a" and compare
- The noisy analog signal "x" and the filtered signal "xf"
 - The outputs "xd" and "xfd" of the two differentiators

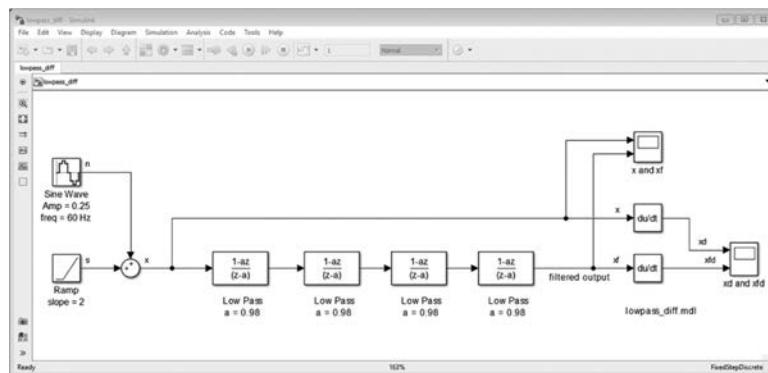


FIGURE E5.33

5.8 MATLAB AND SIMULINK INTERFACE

While it is possible to work with signals and systems exclusively within the MATLAB environment, it is far more efficient to utilize Simulink and the MATLAB toolboxes to solve problems in specific disciplines. Sharing of data between MATLAB and Simulink is a seamless process, enabling MATLAB's extensive capabilities in data analysis and visualization to be utilized.

The following example illustrates how to effectively exploit the MATLAB and Simulink interface. It deals with the Fourier Series and its application to frequency response of linear systems.

A periodic signal $u(t)$ is shown in [Figure 5.98](#). Equation 5.99 describes the signal over one period from $-T/2$ to $T/2$. It is periodic as a result of $u(t + T) = u(T)$, $-\infty < t < \infty$.

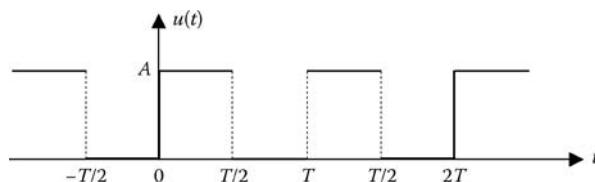


FIGURE 5.98 Periodic signal $u(t)$.

$$u(t) = \begin{cases} 0, & -\frac{T}{2} \leq t < 0 \\ A, & 0 \leq t < \frac{T}{2} \end{cases} \quad (5.99)$$

Its Fourier Series expansion (O'Neil 1983) is

$$u(t) = a_0 + \sum_{n=1,3,5,\dots}^{\infty} a_n \sin n\omega_0 t \quad (5.100)$$

where

$\omega_0 = 2\pi/T$ is called the fundamental frequency
 $u_n(t) = a_n \sin n\omega_0 t, n = 1, 3, 5, \dots$ is the n th harmonic

the Fourier coefficients are

$$a_0 = \frac{A}{2}, \quad a_n = \frac{2}{n\pi}, \quad n = 1, 3, 5, \dots \quad (5.101)$$

Suppose $u(t)$ is the input to a second-order system with transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5.102)$$

By the principle of superposition, $y(t)$ is equal to the sum of the second-order system response to the constant a_0 and the responses to the harmonic components $u_n(t), n = 1, 3, 5, \dots$ Fourier coefficients of the truncated series $a_0 + \sum_{n=1,3,5,\dots}^{19} a_n \sin n\omega_0 t$ are evaluated in the M-file “Ch5_Fourier_Series.m,” a portion of which is listed as follows:

```
% MATLAB Script File Ch5_Fourier_Series.m
% Fourier Series of periodic function u(t)
% f(t) = A, 0 <= t < T/2
% = 0, T/2 <= t < T
n = 19; % order of truncated Fourier Series of u(t)
k = 1:2:n; % harmonics of u(t)
A = 10; % amplitude of u(t)
a = 2*A./(k.*pi); % Fourier Series coefficients a(k), k = 1, 3, 5, ..., n
a0 = 0.5*A; % ave value of u(t)
T = 0.1; % period of u(t)
w0 = 2*pi/T; % input frequency
wh = k*w0; % harmonic frequencies
wr = wh(5); % Set resonant frequency equal to freq of 9th harmonic
wn = 1.01*wr; % calculate natural frequency
zeta = sqrt((1- (wr/wn)^2)/2); % calculate damping ratio
w = linspace (0,1500,500); % range of freq for jG(jw)j plot
s = j*w; % complex freqs
magG_w = (wn.^2)./abs(s.^2*p2*zeta*wn*s*pwn.^2); % jG(jw)j plot(w,magG_w)
hold on,s=j*wh;
magG_wh = (wn.^2)./abs(s.^2*p2*zeta*wn*s*pwn.^2); % jG(jwh)j
plot(wh,magG_wh,".", "MarkerSize",12)
num = [wn^2]; % numerator of G(s)
denom = [1 2*zeta*wn wn^2]; % denominator of G(s)
```

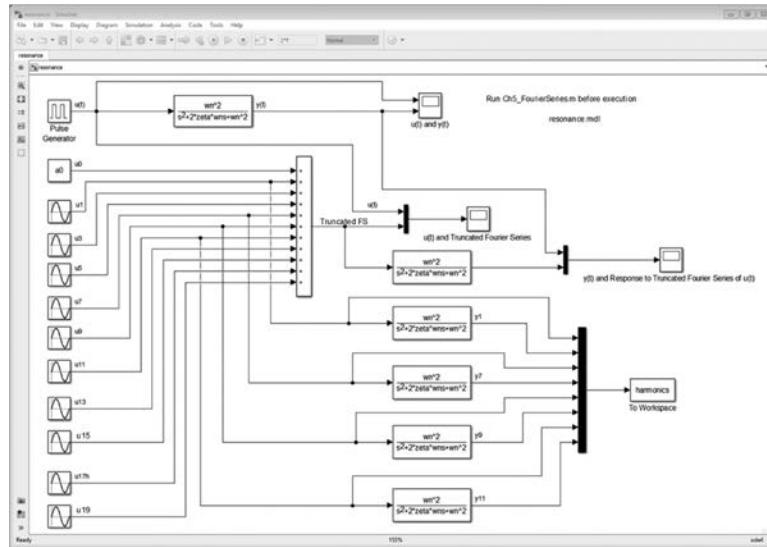


FIGURE 5.99 Simulink diagram for finding the response to $u(t)$ and truncated Fourier Series of $u(t)$.

```

SYS = TF (num, denom); % transfer function of G(s)
Figure, bode (SYS,{50, 1500})
[MAG, PHASE, wh] = BODE (SYS, wh); % Evaluate jG(jwh)j and Angle (G(jwh))
sim("resonance") % call Simulink model "resonance.mdl"
subplot(4, 2, 1); plot(t, harmonics (:,1))
subplot (4, 2, 3); plot(t, harmonics (:,2))

```

The truncated series expansion of $u(t)$ is evaluated and compared to the input $u(t)$ as part of a Simulink simulation shown in Figure 5.99. The Fourier coefficients and harmonic frequencies are used to set the parameters in the “Sine Wave” blocks. A comparison of the signal $u(t)$ and the truncated Fourier Series is shown in Figure 5.100.

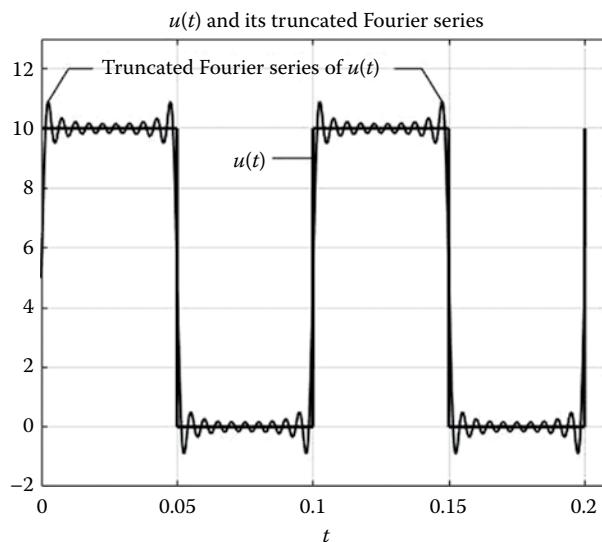


FIGURE 5.100 Periodic signal $u(t)$ ($t = 0.1$ s) and 19th-order truncated series.

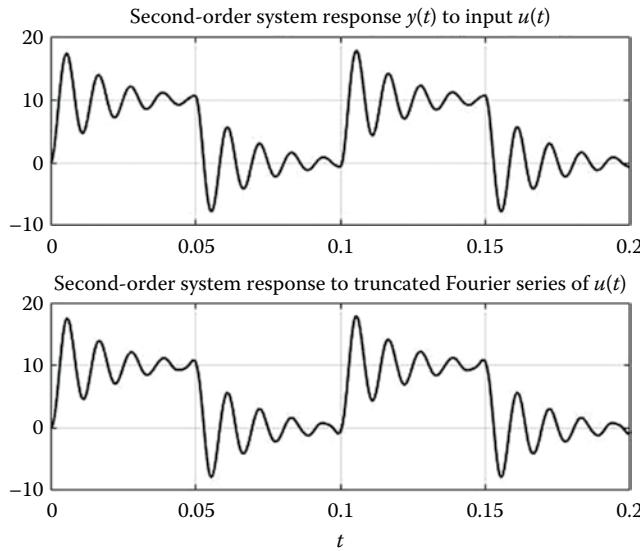


FIGURE 5.101 Response of a second-order system to $u(t)$ and its truncated Fourier Series.

The resonant frequency ω_r of the second-order system with transfer function $G(s)$ is set equal to the frequency of the ninth harmonic $9\omega_0 = 9(2\pi/T) = 565.49$ rad/s. The natural frequency ω_n is chosen slightly higher, that is, $\omega_n = 1.01\omega_r$, producing a lightly damped system with damping ratio of approximately 0.1 calculated from (Ogata 1998)

$$\zeta = \sqrt{\frac{1 - (\omega_r/\omega_n)^2}{2}} \quad (5.103)$$

The second-order system response to $u(t)$ and its response to the truncated Fourier Series representation of $u(t)$ are shown in Figure 5.101.

Clearly, enough harmonics of $u(t)$ have been retained in the truncated Fourier Series to accurately predict the response of the second-order system under consideration.

Next, we discuss how MATLAB and Simulink can be used effectively to demonstrate the phenomenon of resonance. The M-file “Ch5_Fourier_Series.m” evaluates the magnitude function $|G(j\omega)|$ over the frequency range $0 \leq \omega \leq 1500$ rad/s and plots the results with the harmonic frequencies shown in Figure 5.102. Note that the resonant frequency ω_r , where the peak amplitude of $|G(j\omega)|$ occurs is in fact equal to the frequency of the ninth harmonic. A similar finding is possible using the control system toolbox to specify the transfer function and draw a Bode plot or merely compute the magnitude function with “MAG” at selected frequencies and plot the results.

The Fourier coefficients of the truncated series expansion of $u(t)$ are shown in Table 5.3. Also listed are the frequency response characteristics of the second-order system at the harmonic frequencies.

The peak magnitude at the resonant frequency is

$$\underset{\omega \geq 0}{\text{Max}} |G(j\omega)| = |G(j\omega_r)| = 5.0624 \quad (5.104)$$

The harmonic components $u_n(t)$, $n = 1, 7, 9, 11$ in the truncated series expansion of $u(t)$ in Figure 5.98 are given by

$$u_1(t) = a_1 \sin \omega_0 t = 6.3662 \sin 62.8t \quad (5.105)$$

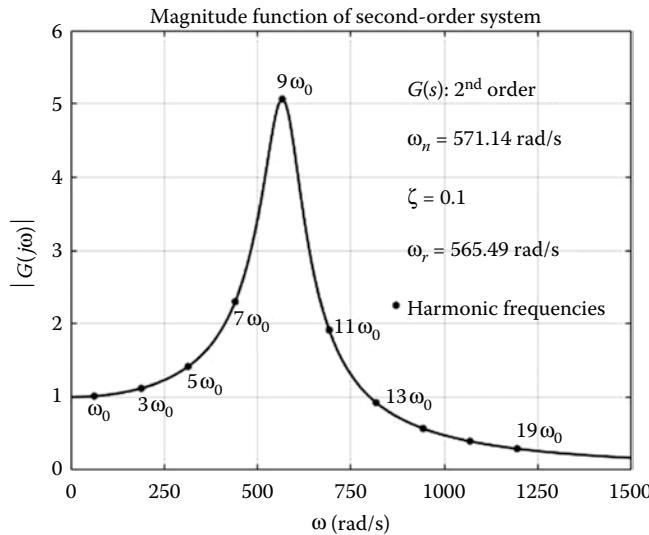


FIGURE 5.102 Magnitude function of a second-order system.

$$u_7(t) = a_7 \sin 7\omega_0 t = 0.9095 \sin 439.8t \quad (5.106)$$

$$u_9(t) = a_9 \sin 9\omega_0 t = 0.7074 \sin 565.5t \quad (5.107)$$

$$u_{11}(t) = a_{11} \sin 11\omega_0 t = 0.5787 \sin 691.2t \quad (5.108)$$

The second-order system response to the above components is

$$y_1(t) = |G(j\omega_0)| a_1 \sin [\omega_0 t + \angle G(j\omega_0)] \quad (5.109)$$

$$= 1.0120(6.3662) \sin (62.8t - 0.0221) \quad (5.110)$$

$$= 6.4426 \sin (62.8t - 0.0221) \quad (5.111)$$

TABLE 5.3
Fourier Coefficients and Magnitude Function at Selected Frequencies

N	$n\omega_0$ (rad/s)	a_n	$ G(jn\omega_0) $	$\angle G(jn\omega_0)$ (rad)
0	0	5	1	0
1	$\omega_0 = 62.8$	6.3662	1.0120	-0.0221
3	$3\omega_0 = 188.5$	2.1221	1.1192	-0.0734
5	$5\omega_0 = 314.2$	1.2732	1.4166	-0.1553
7	$7\omega_0 = 439.8$	0.9095	2.3002	-0.3593
9	$9\omega_0 = 565.5$	0.7074	5.0624	-1.4709
11	$11\omega_0 = 691.2$	0.5787	1.9126	-2.6642
13	$13\omega_0 = 816.8$	0.4897	0.9232	-2.8764
15	$15\omega_0 = 942.5$	0.4244	0.5702	-2.9537
17	$17\omega_0 = 1068.1$	0.3745	0.3960	-2.9940
19	$19\omega_0 = 1193.8$	0.3351	0.2946	-3.0190

$$y_7(t) = |G(j7\omega_0)| a_7 \sin [7\omega_0 t + \angle G(j7\omega_0)] \quad (5.112)$$

$$= 2.3002(0.9095) \sin (439.8t - 0.3593) \quad (5.113)$$

$$= 2.0920 \sin (439.8t - 0.3593) \quad (5.114)$$

$$y_9(t) = |G(j9\omega_0)| a_9 \sin [9\omega_0 t + \angle G(j9\omega_0)] \quad (5.115)$$

$$= 5.0624(0.7074) \sin (565.5t - 1.4709) \quad (5.116)$$

$$= 3.5811 \sin (565.5t - 1.4709) \quad (5.117)$$

$$y_{11}(t) = |G(j11\omega_0)| a_{11} \sin [11\omega_0 t + \angle G(j11\omega_0)] \quad (5.118)$$

$$= 1.9126(0.5787) \sin (691.2t - 2.6642) \quad (5.119)$$

$$= 1.1068 \sin (691.2t - 2.6642) \quad (5.120)$$

The Simulink data for the signals in Equations 5.105 through 5.108, 5.111, 5.114, 5.117, and 5.120 are returned to the MATLAB Workspace (see [Figure 5.99](#)) for use by M-file “*Ch5_Fourier_Series.m*” in preparing the graph shown in [Figure 5.103](#). The input and output components can be identified by referring to the amplitudes in Equations 5.105 through 5.120.

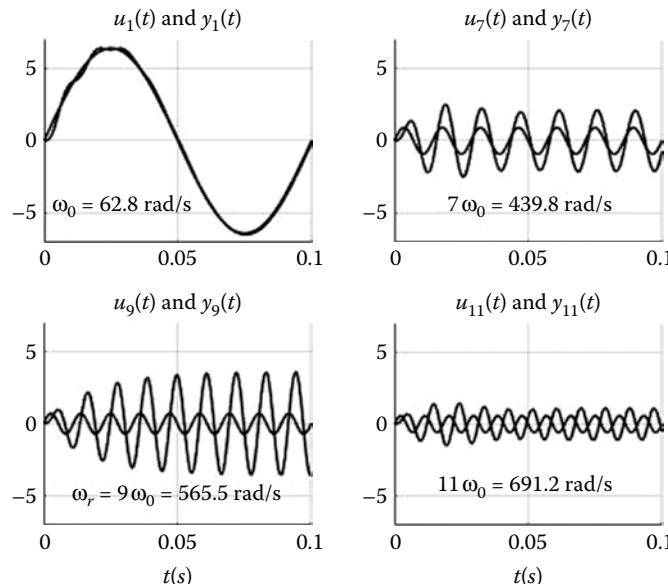


FIGURE 5.103 Several harmonic components of $u(t)$ and response of second-order system.

Note that $y_1(t)$ has a larger amplitude than $y_9(t)$, the system response to the harmonic component at the resonant frequency. This results from $|G(j\omega_0)|a_1 = 6.4426$ being larger than $|G(j9\omega_0)|a_9 = 3.5811$. As a final comment, the line in “*Ch5_Fourier_Series.m*”

```
sim('resonance') % call Simulink model 'resonance.mdl'
```

enables the MATLAB script file “*Ch5_Fourier_Series.m*” to initiate execution of the Simulink model file “*resonance.mdl*.” With additional parameters in the “*sim*” command, the user has control of many of the settings entered in the Simulink “*Simulation Parameters*” dialog box.

EXERCISES

- 5.34 A spring mass system described by $m\ddot{y} + ky = F$ is subject to an external periodic force $F(t)$ shown in [Figure E5.34](#):

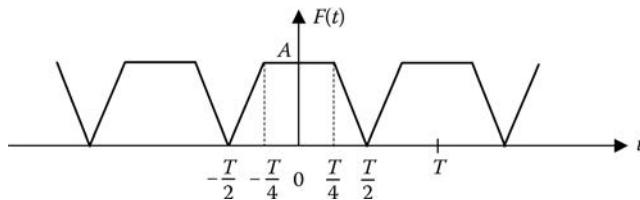


FIGURE E5.34

- a. The Fourier series expansion of $F(t)$ is

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos n\omega_0 t, \quad \omega_0 = \frac{2\pi}{T}$$

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} F(t) dt, \quad a_n = \frac{2}{T} \int_{-T/2}^{T/2} F(t) \cos\left(\frac{2n\pi t}{T}\right) dt, \quad n = 1, 2, 3, \dots$$

Find expressions for the Fourier coefficients a_n , $n = 0, 1, 2, 3, \dots$.

- b. The mass $m = 1$ slug and the natural frequency of the system is $\omega_n = 25$ rad/s. The period of the forcing function T is related to the natural frequency according to $T = 2N\pi/c\omega_n$ where N is a positive integer and c is a constant. Write a MATLAB script file that reads values of c and N and computes the period T and Fourier coefficients a_n , $n = 0, 1, 2, 3, \dots, 3N$. For $A = 1$, $c = 1$, and $N = 5$, use the MATLAB script file to
- c. Plot on the same graph $F(t)$ and the truncated Fourier Series

$$f_{FS}(t) = \frac{a_0}{2} + \sum_{n=1}^{3N} a_n \cos n\omega_0 t$$

for $0 \leq t \leq 3$ T. Comment on the results.

- d. Prepare a Simulink diagram for simulating the response of the system with zero initial conditions. Call the simulation from the script file using the same values for c and N . Return the values of $\{t, y(t)\}$ to the MATLAB Workspace and plot the response. (The simulation should run long enough to recognize the steady-state response.) Comment on the results.

- 5.35 The dynamic interaction of rabbit and fox populations in a forest is under investigation. The predator-prey ecosystem is illustrated in block diagram form in [Figure E5.35a](#):

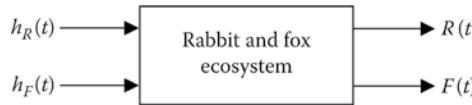


FIGURE E5.35A

$R(t)$ = Population of rabbits after “ t ” weeks

$F(t)$ = Population of foxes after “ t ” weeks

$h_R(t)$ = Rate of rabbit hunting (rabbits/week)

$h_F(t)$ = Rate of fox hunting (fox/week)

A Simulink diagram of the system is shown in [Figure E5.35b](#):

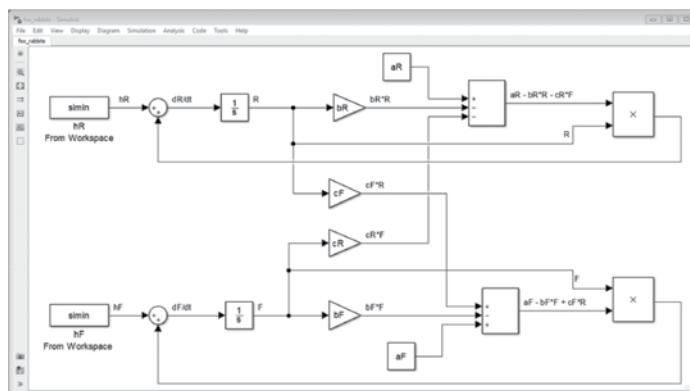


FIGURE E5.35B

a_R, b_R, c_R = constant parameters defining the growth rate of rabbits

a_F, b_F, c_F = constant parameters defining the growth rate of foxes

- Find the mathematical model governing the system dynamics.
- Find the nontrivial equilibrium points (R_e, F_e) when $h_R(t) = 0, t \geq 0$ and $h_F(t) = 0, t \geq 0$.
- Write a MATLAB script file to set the following baseline parameter values:

$$a_R = 0.05 \frac{\text{rabbits/week}}{\text{rabbit}^2}, \quad b_R = 5 \times 10^{-7} \frac{\text{rabbits/week}}{\text{rabbit}^3}, \quad c_R = 1.25 \times 10^{-5} \frac{\text{rabbits/week}}{\text{rabbit}^2 - \text{fox}}$$

$$a_F = 0.04 \frac{\text{foxes/week}}{\text{fox}^2}, \quad b_F = 2 \times 10^{-5} \frac{\text{foxes/week}}{\text{fox}^3}, \quad c_F = 8 \times 10^{-7} \frac{\text{foxes/week}}{\text{foxes}^2 - \text{rabbit}}$$

$$h_R(t) = 0, \quad t \geq 0 \quad h_F(t) = 0, \quad t \geq 0$$

$$R(0) = 50,000 \quad F(0) = 1000$$

- Run the simulation and plot $R(t)$ and $F(t)$ vs. t until the system reaches equilibrium.
- Obtain a solution trajectory R vs. F . Place a vertical line at $F = F_e$ and a horizontal line at $R = R_e$. This will allow you to verify by inspection if the solution trajectory approaches the theoretical equilibrium.

- f. Investigate the effect of changes in c_R , a parameter that measures the interaction between foxes and rabbits. Plot families of appropriate responses corresponding to 0%–50% change in c_R .
 - g. Establish a policy for hunting rabbits that makes the number of foxes equal to approximately 2500 at equilibrium.
 - h. Establish a policy for hunting foxes that makes the number of rabbits equal to approximately 35,000 at equilibrium.
- 5.36 The tank shown in Figure E5.36 has a brine solution flowing into it. The solution is stirred well enough, so that the concentration of salt in the tank is uniform.

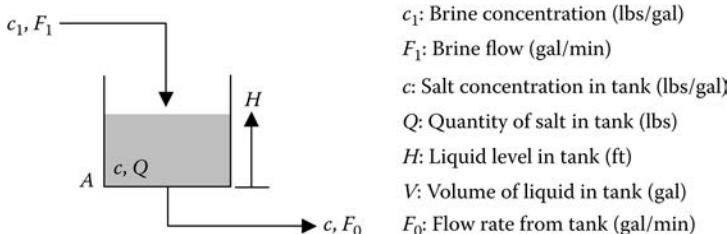


FIGURE E5.36

The mathematical model consists of the following equations:

$$\frac{dQ}{dt} = c_1 F_1 - c F_0$$

$$c = \frac{Q}{A}, \quad V = AH$$

$$A \frac{dH}{dt} + F_0 = F_1, \quad F_0 = \alpha H^{1/2}$$

The system baseline parameter values are $A = 20 \text{ ft}^2$ and $\alpha = 6 \text{ gal/min per ft}^{1/2}$.

Note: 1 ft^3 of water is roughly 8.3 gal.

- a. Draw a simulation diagram of the system.
- b. Choose the state variables as $x_1 = Q$ and $x_2 = H$ and the outputs $y_1 = c$, $y_2 = Q$, and $y_3 = V$. Write the state equations in the form

$$\dot{x}_1 = f_1(x_1, x_2, c_1, F_1), y_1 = g_1(x_1, x_2, c_1, F_1)$$

$$\dot{x}_2 = f_2(x_1, x_2, c_1, F_1), y_2 = g_2(x_1, x_2, c_1, F_1)$$

$$y_3 = g_3(x_1, x_2, c_1, F_1)$$

- c. Find expressions for the steady-state values of the states $x_1(\infty)$ and $x_2(\infty)$ and the outputs $y_1(\infty)$, $y_2(\infty)$, and $y_3(\infty)$ assuming c_1 and F_1 are constant.
- d. The tank is initially filled with 100 gal of water (no salt). Brine starts flowing into the tank at the rate of 12 gal/min. The salt concentration of the brine is 0.25 lb/gal. Both the flow rate and salt concentration of the brine flow remain constant. Using explicit Euler integration, find the discrete-time state equations

$$\underline{x}_A(n+1) = \underline{f}[\underline{x}_A(n), \underline{u}(n)]$$

$$\underline{y}_A(n) = g[(\underline{x}_A(n), \underline{u}(n)]$$

used to obtain an approximate solution for the continuous-time states and outputs.

- e. Solve the discrete-time state equations recursively for the discrete-time states $x_{1,A}(n)$ and $x_{2,A}(n)$ and the outputs $y_{1,A}(n)$, $y_{2,A}(n)$, and $y_{3,A}(n)$. Graph the transient responses. Comment on the value of T used for the numerical integrator.
- f. Compare the steady-state results obtained in part (e) with the predicted values from part (c). Comment on the results.
- g. Use Simulink to verify the responses obtained in part (e).

5.9 HYBRID SYSTEMS: CONTINUOUS- AND DISCRETE-TIME COMPONENTS

Hybrid systems consist of continuous-and discrete-time components and the interfaces bridging the gap between them. A good example is a digital controller (microprocessor or general-purpose digital computer) determining discrete-time input (s) to a continuous-time process.

[Figure 5.104](#) shows a digital controller used to regulate the temperature inside a chamber. The DC voltage input to the heater $v(t)$ is determined by a digital control algorithm represented by discrete-time transfer function $D(z)$. The heat input to the chamber is assumed proportional to the square of the heater voltage. A temperature sensor with gain K_S produces a voltage signal $v_S(t)$ for comparison with a reference voltage $v_R(t)$. The reference voltage is based on the commanded temperature $T_R(t)$ (not shown in [Figure 5.104](#)).

The error signal $e(t)$ is sampled every T s in an analog-to-digital (A/D) converter. The A/D converter functions as an interface between the continuous-time inputs (sensor and reference voltage) and the discrete-time digital controller. The error signal $e(k)$ is processed by the digital controller, resulting in an output $v(k)$, the intended voltage to the heater. A digital-to-analog (D/A) converter, operating synchronously with the A/D, produces the voltage. Internal circuitry in the D/A latches the discrete-time input for the duration of the sampling period, resulting in a stepwise constant voltage $v(t)$ applied to the heater. The D/A converter serves as an interface between the discrete-time and continuous-time components. It is modeled by a zero-order hold (ZOH) in [Figure 5.104](#).

The digital controller implements a linear difference equation for $v(k)$ in terms of past values $v(k-1)$, $v(k-2)$, ..., $v(k-n)$ as well as present and past values $e(k)$, $e(k-1)$, ..., $e(k-p)$. Digital controllers are often synthesized by approximating continuous-time controllers. For example, the transfer function of a continuous-time proportional-integral-derivative (PID) controller is

$$\frac{V(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s \quad (5.121)$$

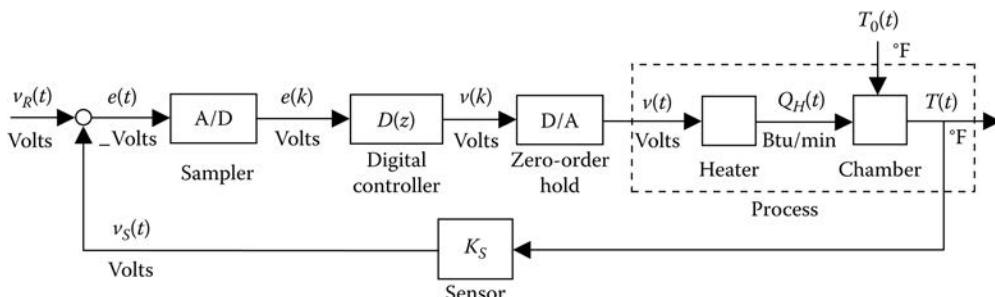


FIGURE 5.104 Digital control of chamber temperature.

Approximating the integral by trapezoidal integration and the derivative by a backward difference equation leads to the equivalent digital transfer function (Jacquot 1981)

$$D(z) = \frac{c_1 z^2 + c_2 z + c_3}{z^2 - z} \quad (5.122)$$

where

$$c_1 = K_P + \frac{K_I T}{2} + \frac{K_D}{T}, \quad c_2 = \frac{K_I T}{2} - K_P - \frac{2 K_D}{T}, \quad c_3 = \frac{K_D}{T} \quad (5.123)$$

The continuous-time process model is (see Section 2.7)

$$C \frac{dT}{dt} + \frac{1}{R} T = \frac{1}{R} T_0 + Q_H \quad (5.124)$$

where

C is the thermal capacitance of the interior space (objects and volume of air assumed to be at the same temperature)

R is the effective thermal resistance of the material separating the inside and outside of the chamber

The heater output is

$$Q_H(t) = \frac{v^2(t)}{R_e} \quad (5.125)$$

where R_e is the electrical resistance of the heater coil. A temperature sensor produces a voltage proportional to the interior temperature

$$v_s(t) = K_s T(t) \quad (5.126)$$

A Simulink diagram of the system is shown in [Figure 5.105](#). The commanded reference temperature T_Y is converted to a reference voltage input with a Units Converter, that is, a “Gain” block with parameter equal to K_S . A “Saturation” block limits the actual voltage $v(t)$ to the heater. The “Zero-Order Hold” and “Rate Transition” blocks are included to resolve timing issues related to the faster simulation execution rate (based on the integration step) and the slower sampling rate of the digital controller.

System parameters were set in the MATLAB script file “*Ch5_dig_cont.m*.”

Chamber: $R = 0.175^\circ\text{F}/\text{Btu}/\text{min}$, $C = 50 \text{ Btu}/^\circ\text{F}$

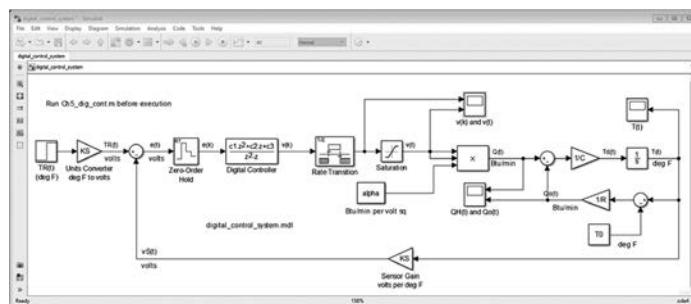


FIGURE 5.105 Simulink diagram of a digital control system for chamber temperature.

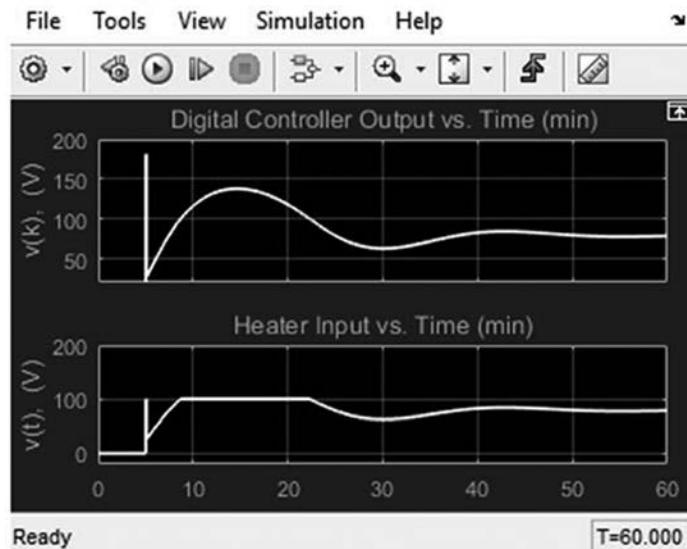


FIGURE 5.106 Digital controller output $v(k)$ and heater input $v(t)$.

Sensor: $K_S = 0.25\text{V}/^\circ\text{F}$

Controller: $K_P = 2, K_I = 2, K_D = 0.25$

Heater: $R_e = 1.25 \Omega, v_{\max} = 100 \text{ V}$

Inputs: $T_R(t) = 125^\circ\text{F}, t \geq 5, T_0(t) = 75^\circ\text{F}, t \geq 0$

Timing: $T = 0.02 \text{ min}$ (sample time), $\Delta t = 0.002 \text{ min}$ (integration step size)

Figure 5.106 shows the voltage $v(k)$ computed from the digital control algorithm and the actual voltage $v(t)$ to the heater. Note the initial spike due to the presence of the proportional control and derivative action in the controller. The initial continuous-time voltage to the heater is “maxed out” at a 100V, the upper limit of the saturation block.

Figure 5.107 shows the heat flows to and from the chamber. Note the constant heat flow to the chamber when the heater is at saturation. At the end of the transient response period, the heat flows have equalized, and the chamber interior is in thermal equilibrium with its surroundings.

Figure 5.108 is a graph of the chamber temperature increasing from its initial value of 75°F to the commanded value of 125°F . The step response is typical of a slightly underdamped second-order system with a settling time between 50 and 60 min.

The thermal time constant of the chamber is

$$\tau = RC = 0.175 \frac{^\circ\text{F}}{\text{Btu/min}} \times 50 \frac{\text{Btu}}{^\circ\text{F}} = 8.75 \text{ min}$$

The sampling time $T = 0.02 \text{ min}$ of the A/D converter is chosen several orders of magnitude less than the process time constant in order to capture the transient behavior of the chamber temperature. A more precise way of determining the sampling rate will be discussed in a subsequent chapter.

The control system is nonlinear as a consequence of Equation 5.125. Laplace transforms cannot be used to find an analytical solution for the system variables. Simulation is the only viable approach to examining the system dynamics.

EXERCISES

In Exercises 5.37 through 5.40, use baseline values for the system parameters found in “Ch5_dig_cont.m” unless otherwise stated.

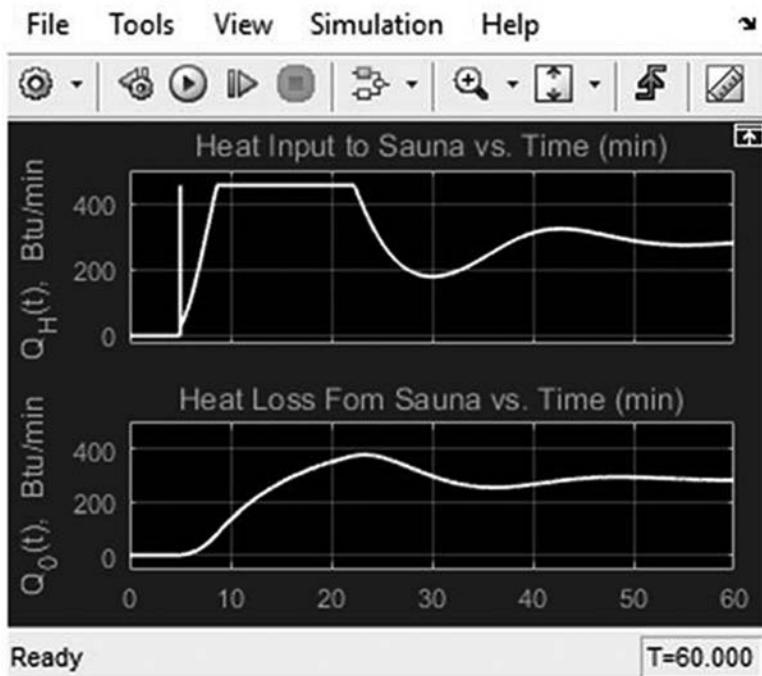


FIGURE 5.107 Heater input $Q_H(t)$ and heat loss $Q_0(t)$ from chamber to surroundings.

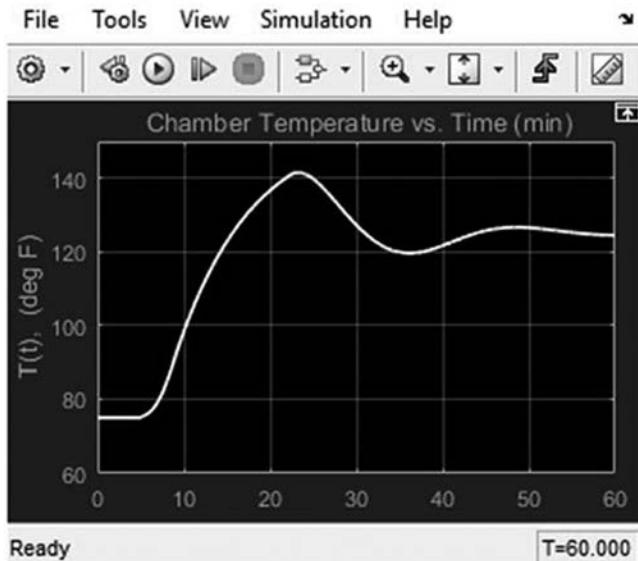


FIGURE 5.108 Chamber temperature response $T(t)$ to reference input $T_R(t) = 125^\circ\text{F}$, $t \geq 5$.

- 5.37 Plot the simulated chamber temperature responses (on the same graph) corresponding to a range of sampling intervals from 0.01 to 0.25 min. Comment on the results.
- 5.38 The maximum output from the chamber heater in watts is $(QH)_{\max} = v_{\max}^2/R_e$.
- Find T_{\max} , the maximum temperature achievable in the chamber.
 - Note: 1 kW = 56.896 Btu/min.

- c. Simulate the chamber temperature when the commanded temperature is set to
 - i. T_{\max}
 - ii. 10% higher than T_{\max}
 - iii. 25% higher than T_{\max}
- 5.39 Simulate the temperature response of the control system with proportional control only, that is, $K_I = 0$ and $K_D = 0$. The set point temperature is 200°F. Vary K_p from 1 to 10 and plot the responses on the same graph.
- 5.40 Suppose the chamber temperature has been constant at $T_R = 125^\circ\text{F}$ for some time. Simulate the chamber temperature $T(t)$ when
 - a. the heater is turned off
 - b. the reference temperature is set to 150°F
- 5.41 Simulate the chamber temperature using a digital controller obtained by approximating the continuous controller in Equation 5.121 using Tustin's method (trapezoidal integration). Compare the results with those shown in [Figures 5.106](#) through [5.108](#).

5.10 MONTE CARLO SIMULATION

The dynamic systems, which have been simulated to this point, were all deterministic, that is, there have been no random components associated with either the system's parameters or inputs. In reality, knowledge of the values of a system's parameters is inexact for a number of reasons. Precise measurement or observation of the parameters may be difficult, or it is possible that the numerical values drift over time as the components age. Quantitative descriptions of the input signals a priori may be probabilistic in nature. The existence of random inputs and uncertain system parameter values leads to stochastic differential equation models with solutions in the form of stochastic processes.

An alternate approach is based on the technique of Monte Carlo simulation. An empirical rather than analytical method, its name stems from the random nature of gambling and associated probabilities. The underlying premise in Monte Carlo simulation is that by repeatedly sampling from known probability distributions, the probabilities of events or probability distributions of functions of a random variable (s) can be approximated. Sampling from the probability distribution of a random variable (or random variables) to generate random deviates is substituted for the process of making observations of the random variable (s) from the real world or physical process itself. In other words, random samples obtained by actual measurements or observations of a random variable are replaced by simulated random samples based on random number generators and known probability distributions.

Consider a simple mechanical system with mass M , spring constant K , and damping coefficient B described by

$$M\ddot{y} + B\dot{y} + Ky = f(t) \quad (5.127)$$

where

y is the displacement of the mass from equilibrium

$f(t)$ is a force acting on the mass

Suppose M , B , and K are continuous random variables with known probability density functions (pdf's) $f_M(u)$, $f_B(u)$, and $f_K(u)$, respectively. The damping ratio ζ

$$\zeta = \zeta(M, B, K) = \frac{B}{2\sqrt{MK}} \quad (5.128)$$

is a new random variable, which, along with the natural frequency, characterizes the system's natural dynamics. Finding the theoretical probability distribution of ζ , that is, its pdf $f_\zeta(u)$, is a

formidable task despite the relative simplicity of Equation 5.128. The following example demonstrates a Monte Carlo simulation to obtain what we shall refer to as an empirical probability density function denoted $\hat{f}_\zeta(u)$ to distinguish it from the true pdf $f_\zeta(u)$. The empirical pdf can be used to approximate probability distributions of other random variables functionally related to the damping ratio such as the overshoot in the step response of underdamped second-order systems.

The parameters M , B , and K are each assumed to vary uniformly between specified limits. The pdf for random variable M is the uniform pdf, denoted $U(M_l, M_u)$ where M_l and M_u are the lower and upper limits of M , respectively. In mathematical terms, the pdf is given by

$$f_M(u) = \begin{cases} \frac{1}{M_u - M_l}, & M_l \leq u \leq M_u \\ 0, & \text{elsewhere} \end{cases} \quad (5.129)$$

Similar expressions apply for the pdfs of random variables B and K , that is,

$$f_B(u) = \begin{cases} \frac{1}{B_u - B_l}, & B_l \leq u \leq B_u \\ 0, & \text{elsewhere} \end{cases} \quad (5.130)$$

$$f_K(u) = \begin{cases} \frac{1}{K_u - K_l}, & K_l \leq u \leq K_u \\ 0, & \text{elsewhere} \end{cases} \quad (5.131)$$

A random variable, uniformly distributed between 0 and 1, also referred to as a random number, is generated by the MATLAB function “rand.” To be more precise, the generated numbers are actually pseudo random numbers, which depend on the specific algorithm implemented for generation. A random number R_i uniformly distributed $U(0, 1)$ is transformed to a new random variable X_i with pdf $U(A, B)$ by

$$X_i = A + (B - A)R_i \quad (5.132)$$

The MATLAB M-file “Ch5_MonteCarlo_damping_ratio.m” generates 100,000 random vectors (M_i, B_i, K_i) , $i = 1, 2, \dots, 100,000$ using lower and upper limits $M_l = 0.9$, $M_u = 1.1$, $B_l = 1.75$, $B_u = 2.25$, $K_l = 3.8$, and $K_u = 4.2$. The corresponding 100,000 damping ratios $\zeta_i = 1, 2, \dots, 100,000$ computed from Equation 5.128 are segregated into equal intervals of width 0.005, several of which are shown in [Table 5.4](#).

A histogram based on the first and third columns of the complete table is shown in the left graph of [Figure 5.109](#). The empirical probability density function $\hat{f}_\zeta(u)$ is obtained by connecting the points $(\bar{\zeta}_i, n_i)$ and rescaling the ordinate values to f_i using Equation 5.133 to make the area under the resulting curve equal to 1.

$$f_i = \frac{n_i}{\text{Number of trials} \times \text{width of interval}} = \frac{n_i}{100,000 \times 0.005} = \frac{n_i}{500} \quad (5.133)$$

Finally, a data point is added $\bar{\zeta}_i = 0.6050$, $f_i = 0$ to assure the pdf $\hat{f}_\zeta(u)$ returns to zero at the upper tail. The result is shown in the right graph of [Figure 5.109](#).

The theoretical probability of ζ falling in a certain interval is the area under $f_\zeta(u)$ for that interval. It is approximated by the area under the empirical pdf $\hat{f}_\zeta(u)$ for the same interval. For example, the

TABLE 5.4
Monte Carlo Simulation Results for Damping Ratio

Interval ($\zeta_{i-1} \leq \zeta \leq \zeta_i$)	Center of Interval $\bar{\zeta}_i$	Frequency of Occurrence n_i	Normalized Frequency of Occurrence f_i
(0.3975, 0.4025)	0.4000	0	0
(0.4025, 0.4075)	0.4050	0	0
(0.4075, 0.4125)	0.4100	33	0.0660
(0.4875, 0.4925)	0.4900	4053	8.1060
(0.4925, 0.4975)	0.4950	4098	8.1960
(0.4975, 0.5025)	0.5000	4062	8.1240
(0.5025, 0.5075)	0.5050	4033	8.0660
(0.5075, 0.5125)	0.5100	3986	7.9720
(0.5875, 0.5925)	0.5900	341	0.6820
(0.5925, 0.5975)	0.5950	164	0.3280
(0.5975, 0.6025)	0.6000	59	0.1180

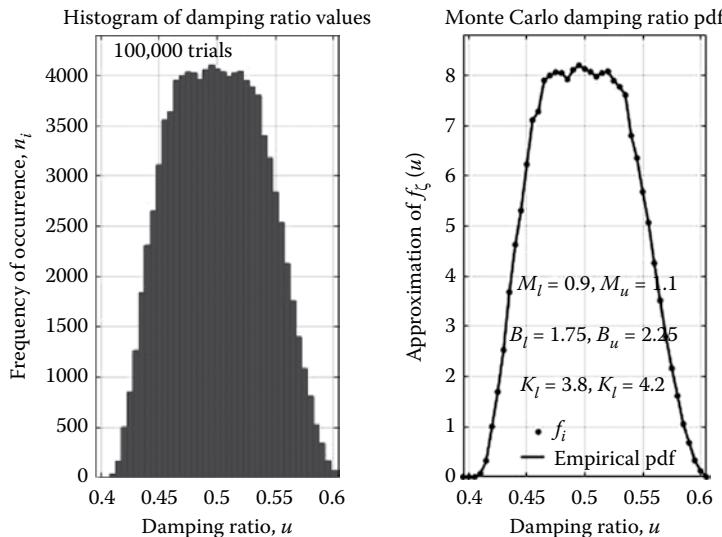


FIGURE 5.109 Histogram of ζ values and empirical pdf $\hat{f}_\zeta(u)$.

estimate of $\Pr(0.45 \leq \zeta \leq 0.5)$ is computed in the M-file “Chain_MonteCarlo_damping_ratio.m” to be 0.4105.

The empirical pdf $\hat{f}_\zeta(u)$ can be used to approximate probabilities involving various performance measures related to the damping ratio. For example, the percent overshoot in the unit step response and the peak amplitude of the frequency response

$$P.O. = f_1(\zeta) = 100e^{-\zeta\pi/\sqrt{1-\zeta^2}} \quad (5.134)$$

$$M_{p\omega} = f_2(\zeta) = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (5.135)$$

How shall we go about determining the empirical pdf $\hat{f}_{M_p\omega}(u)$? A table similar to [Table 5.4](#) with equally spaced intervals of $M_{p\omega}$ and frequencies of occurrence is needed. The first step is to generate a random sample from a population with pdf $\hat{f}_\zeta(u)$. The random sample $(\zeta_1, \zeta_2, \dots, \zeta_n)$ and Equation 5.135 are used to generate the sample $[(M_{p\omega})_1, (M_{p\omega})_2, \dots, (M_{p\omega})_n]$ needed for the new table.

The random sample $(\zeta_1, \zeta_2, \dots, \zeta_n)$ can be generated in several ways. One method relies on the use of random numbers (R_1, R_2, \dots, R_n) and the cumulative probability distribution function (cdf), $\hat{F}_\zeta(u)$ given by

$$\hat{F}_\zeta(u) = \int_{-\infty}^u \hat{f}_\zeta(x)dx, \quad -\infty < u < \infty \quad (5.136)$$

The empirical pdf $\hat{f}_\zeta(u)$ is numerically integrated in “*Ch5_MonteCarlo_damping_ratio.m*,” resulting in $\hat{F}_\zeta(u)$ shown in [Figure 5.110](#). A random damping ratio ζ_i is obtained as the solution to the equation

$$R_i = \hat{F}_\zeta(\zeta_i) \quad (5.137)$$

where (R_i) is a random number uniformly distributed between 0 and 1. That is, ζ_i is obtained from

$$\zeta_i = \hat{F}_\zeta^{-1}(R_i) \quad (5.138)$$

where $\hat{F}_\zeta^{-1}(R_i)$ is the inverse function. The Inverse Transformation Method (Gordon 1978) based on Equation 5.138 is illustrated in [Figure 5.110](#).

After the random sample $[(M_{p\omega})_1, (M_{p\omega})_2, \dots, (M_{p\omega})_n]$ is generated from $(\zeta_1, \zeta_2, \dots, \zeta_n)$ and Equation 5.135, the empirical pdf $\hat{f}_{M_p\omega}(u)$ is obtained in the same way $\hat{f}_\zeta(u)$ was determined.

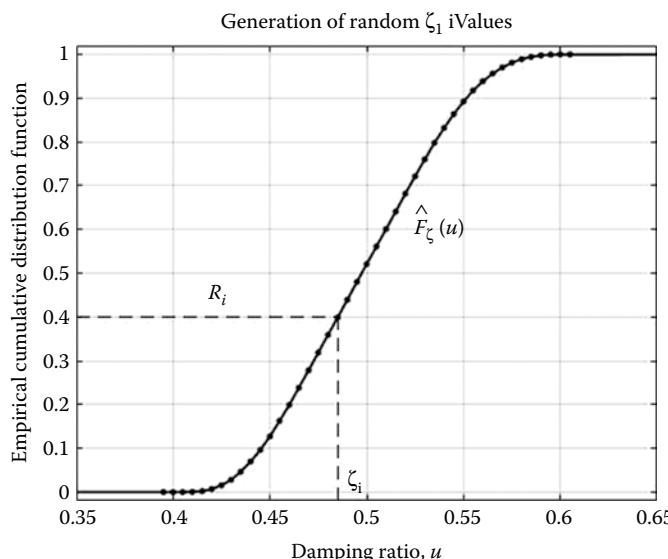


FIGURE 5.110 Illustration of method for generating ζ_i using $\hat{F}_\zeta(u)$.

Suppose we have reason to estimate $\Pr[1.1 \leq M_{p\omega} \leq 1.3]$. The area under $\hat{f}_{M_{p\omega}}(u)$ between 1.1 and 1.3 is easily computed. Alternatively, we could numerically integrate $\hat{f}_{M_{p\omega}}(u)$ to obtain $\hat{F}_{M_{p\omega}}(u)$ and estimate the required probability from

$$\Pr [1.1 \leq M_{p\omega} \leq 1.3] = \hat{F}_{M_{p\omega}}(1.3) - \hat{F}_{M_{p\omega}}(1.1) \quad (5.139)$$

The details are left for an exercise at the end of the section.

5.10.1 MONTE CARLO SIMULATION REQUIRING SOLUTION OF A MATHEMATICAL MODEL

In the previous example, the parameters M , B , and K of a second-order system were random variables with known probability density functions $f_M(u)$, $f_B(u)$, and $f_K(u)$. Additional random variables were introduced, namely, ζ and $M_{p\omega}$ in Equations 5.128 and 5.135. Using Monte Carlo simulation, the theoretical probability density functions $f_\zeta(u)$ and $f_{M_{p\omega}}(u)$ were approximated by empirical pdfs $\hat{f}_\zeta(u)$ and $\hat{f}_{M_{p\omega}}(u)$ without ever solving the differential equation model, Equation 5.127. In the next example, simulation of the mathematical model is an integral component of the overall Monte Carlo simulation study.

Suppose an archer is attempting to hit a falling target as shown in Figure 5.111.

Considering the aerodynamic drag forces on the arrow and target, the differential equations governing the motions of each are

$$m_A \ddot{x}_A = -\alpha_A \dot{x}_A^{n_A} \quad (5.140)$$

$$m_A \ddot{y}_A = -m_A g - \text{sgn}(\dot{y}_A) \cdot \alpha_A |\dot{y}_A|^{n_A} \quad (5.141)$$

$$m_T \ddot{y}_T = -m_T g + \alpha_T |\dot{y}_T|^{n_T} \quad (5.142)$$

where

m_A and m_T are masses of the arrow and target

α_A , n_A , α_T and n_T are parameters for modeling arrow and target drag forces

$x_A(t)$, $y_A(t)$, x_T and $y_T(t)$ are the x - y coordinates of the center of the arrow and center of the circular target

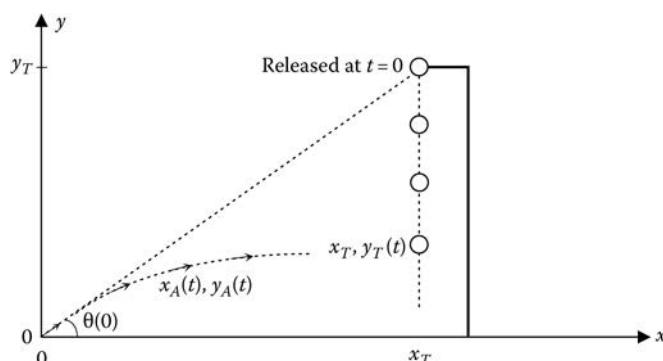


FIGURE 5.111 Arrow fired at a falling target.

The $\text{sgn}(\dot{y}_A)$ function

$$\text{sgn}(\dot{y}_A) = \frac{\dot{y}_A}{|\dot{y}_A|} = \begin{cases} 1, & \dot{y}_A > 0 \\ -1, & \dot{y}_A < 0 \end{cases} \quad (5.143)$$

is required to produce the proper sign on the arrow drag term for both upward and downward flights. Absolute values appear in Equations 5.141 and 5.142 to avoid raising negative speeds to noninteger powers.

The velocity of the arrow is uniquely determined by its speed $v_A(t)$ and direction $\theta(t)$, that is, the angle between the arrow and the horizontal axis. The speed is calculated from

$$v_A(t) = [\dot{x}_A^2(t) + \dot{y}_A^2(t)]^{1/2} \quad (5.144)$$

and the angle $\theta(t)$ is determined from

$$\tan \theta(t) = \frac{\dot{y}_A(t)}{\dot{x}_A(t)} \quad (5.145)$$

$$\Rightarrow \theta(t) = \tan^{-1} \left[\frac{\dot{y}_A(t)}{\dot{x}_A(t)} \right] \quad (5.146)$$

Baseline parameter values for the system are

$$m_A = 0.125/g \text{ slugs}, \alpha_A = 4.75 \times 10^{-6}, n_A = 1.85$$

$$m_T = 1/g \text{ slugs}, \alpha_T = 3 \times 10^{-6}, n_T = 2.3$$

$$x_A(0) = 0 \text{ ft}, y_A(0) = 0 \text{ ft}, v_A(0) = 80 \text{ ft/s}$$

$$x'_T = 100 \text{ ft}, y_T(0) = 150 \text{ ft}$$

The arrow is 2.4 ft in length and the target is 3 ft in diameter.

A Simulink diagram for simulating the trajectories of the arrow and target is shown in [Figure 5.112](#). The Simulink model “arrow.mdl” is called from the M-file “Ch5_MonteCarlo_arrow.m.”

Initially, the aerodynamic drag forces were zeroed out ($\alpha_A = \alpha_T = 0$) and the arrow’s angle of departure $\theta(0)$ was set to the angle of the line of sight to the target because the archer knows, from Physics, that the target will be struck (in the absence of aerodynamic drag forces) under those conditions. The flight path of the arrow and its position at 0.25 s increments is shown in [Figure 5.113](#). The target is captured at 0.25 s increments starting at 2 s and shown as well. [Figure 5.113](#) confirms that the arrow and target appear to be at the same point after approximately 2.25 s.

[Figure 5.114](#) is a close-up snapshot of the arrow and target when the arrow has traveled the horizontal distance to the target. The arrow strikes the target after 2.25 s have elapsed. The arrow and target coordinates at impact are $(x_A = 100 \text{ ft}, y_A = 68.32 \text{ ft})$ and $(x_T = 100 \text{ ft}, y_T = 68.39 \text{ ft})$, respectively. How do you explain the slight difference between y_A and y_T ? Note that the simulation is halted when the arrow strikes the ground after 4.15 s (see “display” in [Figure 5.112](#)).

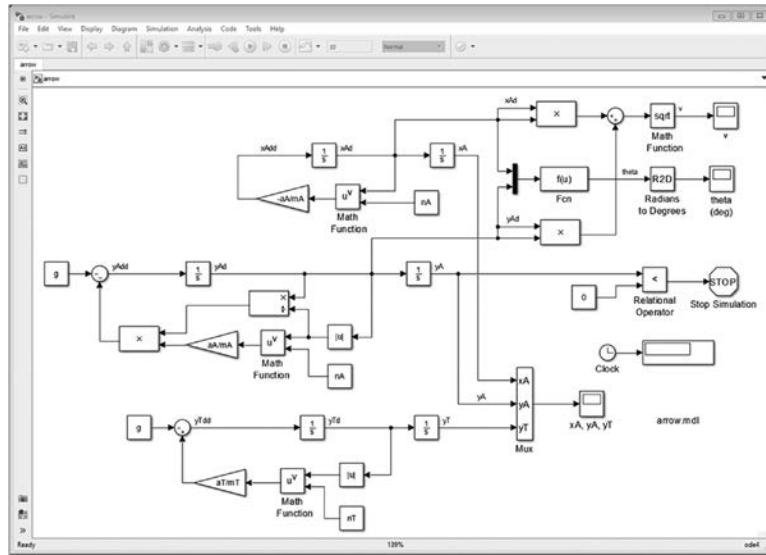


FIGURE 5.112 Simulink diagram of arrow and target simulation.

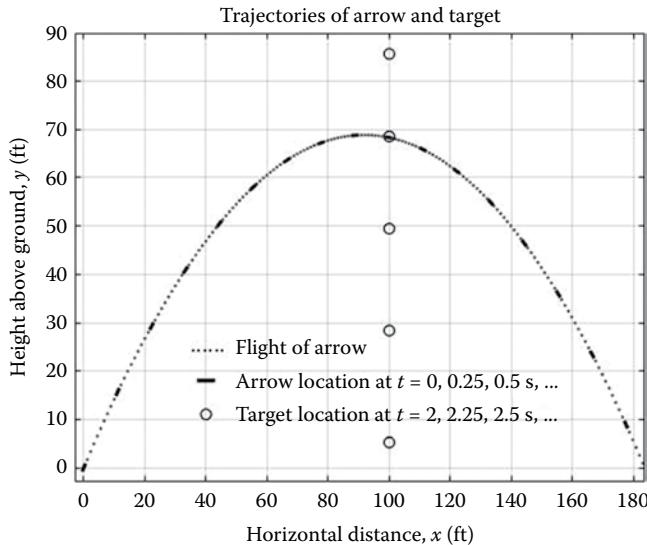


FIGURE 5.113 Path of the arrow and target at selected times.

The reader should experiment with different target heights and arrow initial velocities to confirm the result whereby the arrow always intercepts the target. Naturally, the time and location where the arrow hits the target will depend on the new initial conditions.

A natural question to ask is “What happens when the drag forces are accounted for and an element of randomness associated with the archer’s aim and initial speed of the arrow is introduced?” To answer the question, let us assume that the arrow’s angle of departure $\theta(0) = \theta_0$ and its initial velocity $v(0) = v_0$ off the bow are random variables with probability density functions

$$f_{\theta_0}(u) \sim N(\mu_{\theta_0}, \sigma_{\theta_0}), \quad -\infty < u < \infty \quad (5.147)$$

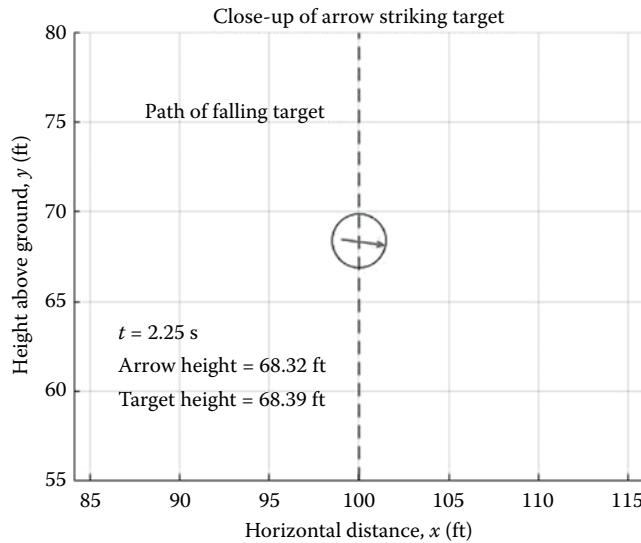


FIGURE 5.114 Close-up of the arrow and target.

$$f_{v_0}(v) \sim U[(v_0)_L, (v_0)_U], \quad (v_0)_L \leq v \leq (v_0)_U \quad (5.148)$$

where

μ_{θ_0} and σ_{θ_0} are the mean and standard deviation of the Normal population
 $(v_0)_L$ and $(v_0)_U$ are the lower and upper limits of the Uniform population

A Monte Carlo experiment can be designed to estimate the probability of hitting the target. “Ch5_MonteCarlo_arrow.m” uses the MATLAB functions “rand” and “randn” to generate random deviates $R_i \sim U(0, 1)$ and $z_i \sim N(0, 1)$. R_i and z_i are transformed to random deviates from the desired populations, Equations 5.147 and 5.148 by

$$(\theta_0)_i = \mu_{\theta_0} + z_i \sigma_{\theta_0} \quad (5.149)$$

$$(v_0)_i = (v_0)_L + [(v_0)_U - (v_0)_L]R_i \quad (5.150)$$

For now, let us assume that the mean angle of departure of the arrow is equal to the sight angle to the target and the standard deviation is 1° . Further, assume that the initial velocities are uniformly distributed between 75 and 85 ft/s. Two Monte Carlo experiments were performed, each with a total of 50,000 random vectors $[(\theta_0)_i, (v_0)_i]$ generated and 50,000 Simulink simulation runs executed. During each run, the occurrence of a “hit” or “miss” is determined and recorded. A “hit” occurs at time $t = \hat{t}$ when the arrow has traveled a horizontal distance x_T , that is, $x_A(\hat{t}) = x_T$ provided

$$y_T(\hat{t})r_T \leq y_A(\hat{t}) \leq y_T(\hat{t}) + r_T \quad (5.151)$$

where $r_T = 1.5$ ft is the radius of the target. The vertical separation between the arrow and target at $t = \hat{t}$ is the distance Δ ,

$$\Delta = y_A(\hat{t}) - y_T(\hat{t}) \quad (5.152)$$

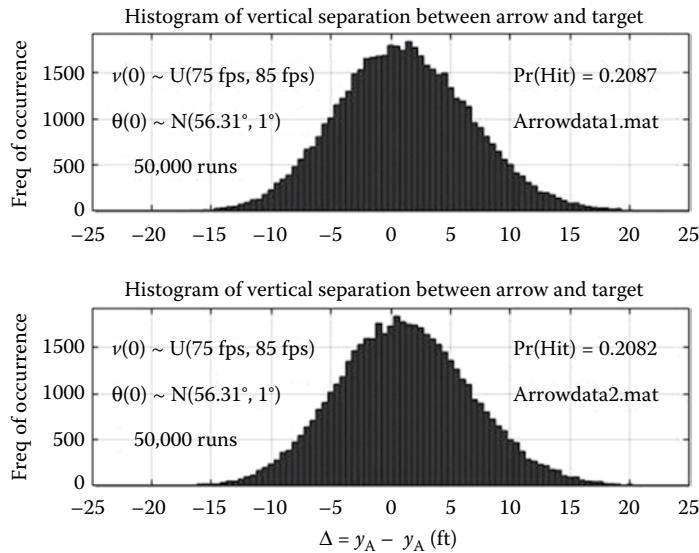


FIGURE 5.115 Estimated $\Pr(\text{hit})$ and histogram of separations.

Results of both experiments are saved in MATLAB data files “arrowdata1.mat” and “arrowdata2.mat.” Histograms of the separations $\Delta_i, i = 1, 2, \dots, 50,000$ for both Monte Carlo runs are plotted in M-file “*Ch5_plot_arrow_histogram.m*” and shown in [Figure 5.115](#) along with the estimated probability of hitting the target.

The histograms suggest that the separation Δ is approximately normally distributed with mean zero. Since more than 99% of the total area under a Normal pdf lies within the mean plus and minus three standard deviations, the standard deviation of Δ is approximately 5 ft.

A question that naturally arises with Monte Carlo simulation is “How many random trials are needed to accurately estimate an unknown theoretical probability?” [Figure 5.116](#) is a plot of the

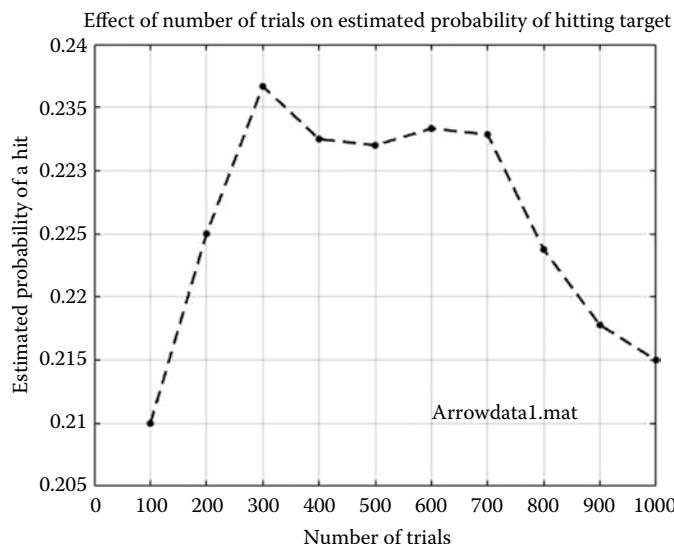


FIGURE 5.116 Estimated probability of a hit after 100, 200, ..., 1000 trials.

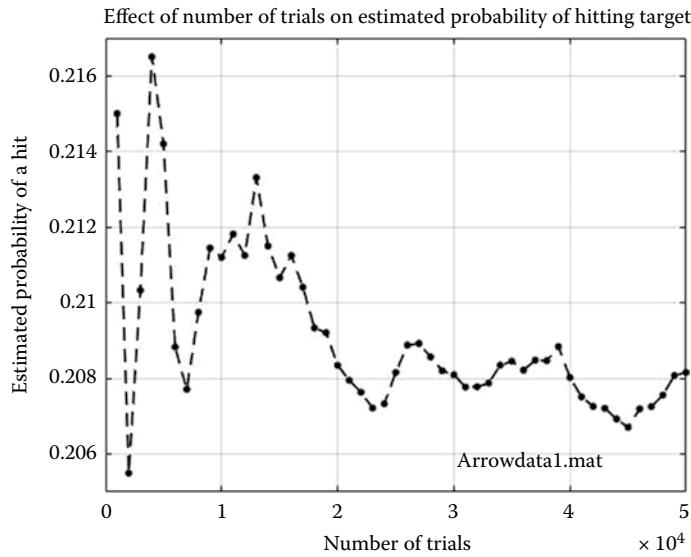


FIGURE 5.117 Estimated probability of a hit after 1000, 2000, ..., 50,000 trials.

estimated probability of hitting the target computed after 100, 200, ..., 1000 trials from the first data file “arrowdata1.mat.”

After 1000 trials, an estimate of the true (unknown) probability of hitting the target, under the conditions given for $\theta(0)$ and $v(0)$, is accurate to one place after the decimal point. Figure 5.117 is a similar plot showing the estimated probability of hitting the target computed after every 1000 trials. The estimated probability of a hit based on 50,000 trials is now accurate to three places after the decimal point.

Figure 5.118 shows arrow and target locations from four of the random trials. Note the correlation between the height of the arrow and the initial speed $v(0)$. The arrow is located at a higher elevation when the initial speed is greater. Also, Δ , the separation between the arrow and target, should be

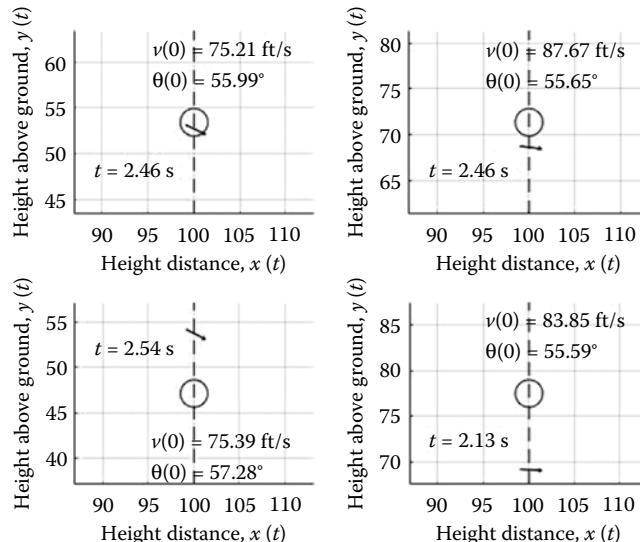


FIGURE 5.118 Arrow and target positions from four runs.

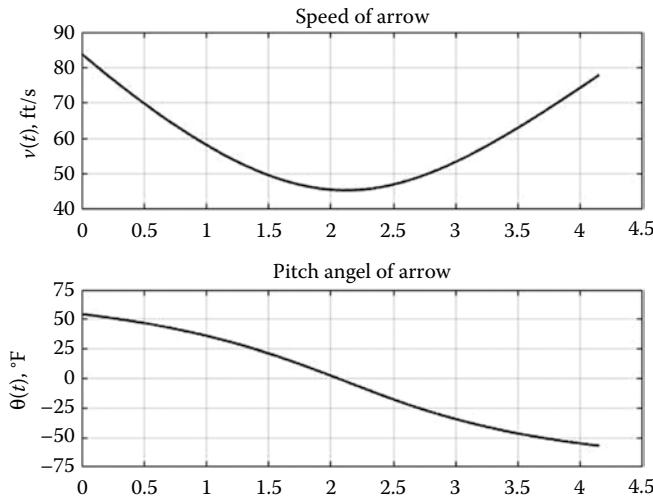


FIGURE 5.119 Arrow speed $v(t)$ and pitch $\theta(t)$ for $v(0) = 83.85 \text{ ft/s}$, $\theta(0) = 54.59^\circ$.

dependent on the angle of departure $\theta(0)$, specifically its relationship to the line of sight angle θ_{LS} (see Figure 5.111).

$$\theta_{LS} = \tan^{-1} \left(\frac{y_T(0)}{x_T} \right) = \tan^{-1} \left(\frac{150}{100} \right) = 0.9828 \text{ rad } (56.31^\circ) \quad (5.153)$$

In the first run (upper left graph in Figure 5.118), the angle of departure (55.99°) is slightly less than the line of sight angle, and the arrow strikes the target just below its center. In the second run (upper right), the initial angle $\theta(0)$ is even less and the arrow passes under the target. In the last two runs, the angle of departure is significantly greater (lower left) and significantly less (lower right) than θ_{LS} , and the corresponding separations are greater and in the expected direction.

The arrow speed $v(t)$ and pitch angle $\theta(t)$ for the last case (lower right corner) are shown in Figure 5.119. When the arrow is directly below the target at $t = 2.13 \text{ s}$, the speed is 45.22 ft/s and the pitch angle is -2.93° .

EXERCISES

In Exercises 5.42 and 5.43, M , B , and K are randomly distributed according to Equations 5.129 through 5.131 with the same limits given in the text.

- 5.42 Use “*Ch5_MonteCarlo_damping_ratio.m*” or write your own program to find and graph the approximate pdf $\hat{f}_{M_{p\omega}}(u)$ and cdf $\hat{F}_{M_{p\omega}}(U)$. Find $\Pr[1.1 \leq M_{p\omega} \leq 1.3]$.
- 5.43 The resonant frequency of a second-order system depends on the damping ratio and natural frequency according to

$$\omega_r = \omega_n \sqrt{1 - 2\zeta^2}, \quad \zeta \leq 0.707$$

- Use Monte Carlo simulation to approximate the true pdf and cdf for ω_r .
- Graph $\hat{f}_{\omega_r}(u)$ and $\hat{F}_{\omega_r}(u)$.
- Estimate $\Pr[\omega_r > \sqrt{2} \text{ rad/s}]$.

- 5.44 Repeat Exercise 5.43 if the mass M is normally distributed with mean $\mu_M = 1$ slug and standard deviation $\sigma_M = 0.25$ slugs. Assume B and K are no longer random, instead $B = 2 \text{ lb s/ft}$ and $K = 4 \text{ lb/ft}$.
- 5.45 Suppose the arrow and target with mass and aerodynamic properties given in the text are dropped from an airplane in level flight at a cruising speed of $v_{cr} = 600 \text{ ft/s}$.
- Find expressions for the terminal velocities of both.
 - Simulate their descent from an altitude of 10,000 ft with zero initial velocity.
 - Plot the acceleration of each during their descent.
- 5.46 Neglecting aerodynamic damping forces and assuming that the initial firing angle of the arrow is equal to the sight angle to the target, perform a simulation study to produce the missing graphs in Figure E5.46:

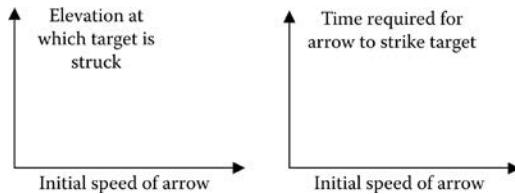


FIGURE E5.46

- 5.47 A boy is throwing rocks, aiming at a circular target with diameter D . The center of the target is x_T ft down range from where he is located (see Figure E5.47). The aerodynamic drag force is proportional to the speed of the rock with drag constant α . The rocks are launched from a height of y_0 at an angle $\varphi(0)$ and initial speed $v(0)$. The weight of the rock is W . The distance downrange where the rock lands is R .

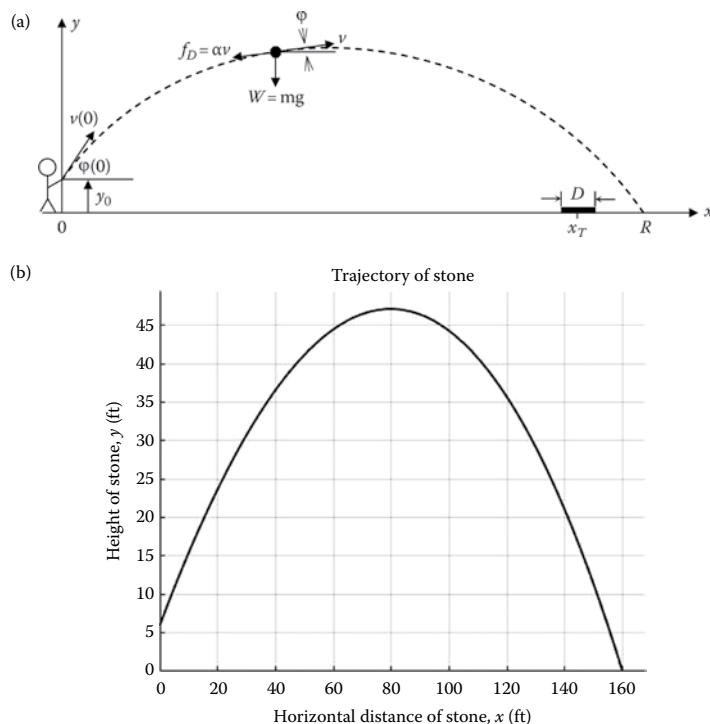


FIGURE E5.47

Baseline system parameter values are

$$y_0 = 6 \text{ ft}, x_T = 160 \text{ ft}, D = 4 \text{ ft}, \alpha = 9 \times 10^{-4} \text{ lb/ft/s}, W_0 = 0.5 \text{ lb}, \\ \varphi(0) = 45^\circ, \text{ and } v(0) = 75 \text{ ft/s}$$

- a. Write the equations comprising the mathematical model of the system in state variable form $\dot{\underline{x}} = f(\underline{x}, \underline{u})$ where the state vector $\underline{x} = [x \ \dot{x} \ y \ \dot{y}]$.
 - b. Use Simulink to simulate the system under baseline conditions, and verify the stone trajectory shown in figure of Exercise 5.47:
 - c. The boy picks up a rock, the weight of which is uniformly distributed between 0.25 and 0.75 lb, and throws it with initial speed and angle given by the baseline values. Find the probability of the rock landing on the target.
 - d. Prepare a histogram for the random variable $\Delta = |R - x_T|$ and use it to find the empirical probability density function $\hat{f}_\Delta(u), \Delta \geq 0$.
 - e. Repeat parts (d) and (e) if $W = W_0 = 0.5$ lb and $\theta(0) \sim U(40^\circ, 50^\circ)$.
- 5.48 A particle slides without friction along a path given by $y = f(x) = x^{1/2}$ under the influence of gravity as shown in [Figure E5.48](#):

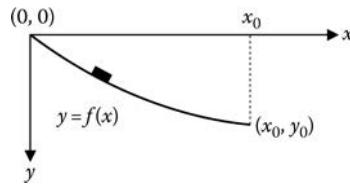


FIGURE E5.48

The time required for the particle to slide down the curve starting from the origin to the point (x_0, y_0) is (Speckhart 1976)

$$t_0 = \frac{1}{\sqrt{2g}} \int_0^{x_0} \sqrt{\frac{1 + (dy/dx)^2}{y}} dx$$

The termination value x_0 is a random variable uniformly distributed between 1 and 5 along the curve. Implement a Monte Carlo experiment culminating in a histogram for the random variable t_0 .

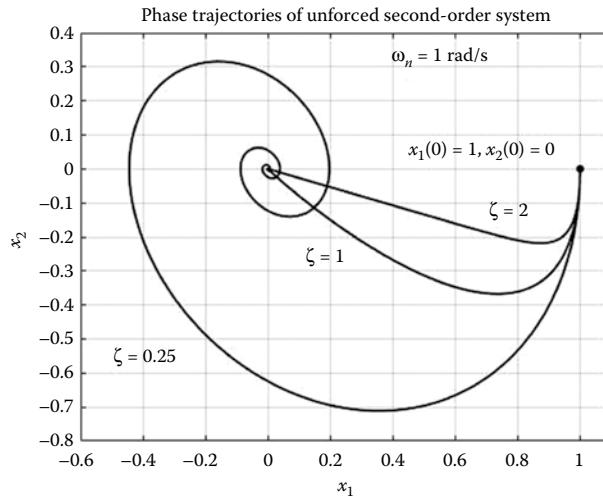
- 5.49 Consider the second-order system $\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = 0$ with initial conditions $y(0) = y_0$, $\dot{y}(0) = 0$. Introduce state variables $x_1 = y$, $x_2 = \dot{y}$. Phase plots for an underdamped ($\zeta = 0.25$), critically damped ($\zeta = 1$), and overdamped ($\zeta = 2$) case with $\omega_n = 1$ rad/s and $y_0 = 1$ are shown in [Figure E5.49](#):

- a. Plot a histogram for the distance from the initial point $x_1(0) = 1$, $x_2(0) = 0$ to the steady-state equilibrium point $x_1(\infty) = 0$, $x_2(\infty) = 0$ along the trajectories in state space if the damping ratio is uniformly distributed between 0 and 2.

Note that the distance from the initial point $(1,0)$ to the point $[x_1(t), x_2(t)]$ along the trajectory is given by

$$s(t) = \int_0^t (\dot{x}_1^2 + \dot{x}_2^2)^{1/2} dt$$

- i. Repeat part (a) for the case where $\zeta = 0.25$, and the natural frequency ω_n is uniformly distributed between 0 and 100 rad/s.

**FIGURE E5.49**

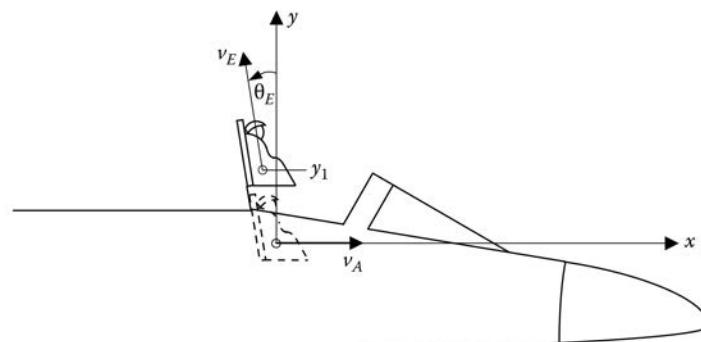
- ii. Repeat part (a) for the case where $\zeta = 1$, and the natural frequency ω_n is uniformly distributed between 0 and 12.5 rad/s.
- iii. Repeat part (a) for the case where $\zeta \sim U(0, 2)$, $\omega_n \sim U(0, 100)$, and $y_0 \sim U(0, 1)$.

5.11 CASE STUDY: PILOT EJECTION

Several benchmark applications of continuous-time simulation using analog and digital computers have been around for decades. Simulation of a pilot and seat ejected from a fighter aircraft falls in this category (Korn and Wait 1978). The system is shown in [Figure 5.120](#).

When forced to eject, the combination of pilot and seat trajectory is controlled by a set of guide rails until it is clear of the plane. The ejection velocity v_E is constant along a direction θ_E from the y axis of the plane. Ejection occurs when the pilot and seat have traveled a vertical distance y_i .

After ejection from the aircraft, the pilot and seat follow a ballistic trajectory subject to an aerodynamic drag force and its own weight. The equations of motion can be developed in the $x-y$ coordinate system or $n-t$ coordinate system, where n and t refer to directions normal and tangential to the flight of the pilot and seat as shown in [Figure 5.121](#). Summing forces in the n and t directions,

**FIGURE 5.120** Diagram of pilot ejection.

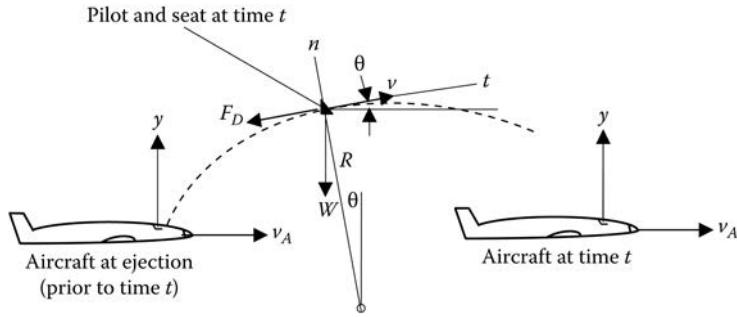


FIGURE 5.121 Trajectory of pilot and seat after ejection.

$$\sum F_t = ma_t \quad (5.154)$$

$$\Rightarrow -F_D - W \sin \theta = mv \quad (5.155)$$

$$\sum F_n = ma_n \quad (5.156)$$

$$\Rightarrow -W \cos \theta = m \frac{v^2}{R} \quad (5.157)$$

where R is the instantaneous radius of curvature of the pilot and seat trajectory. The plane is assumed to be traveling in a horizontal direction at constant speed v_A .

The forward velocity v and angular velocity $\dot{\theta}$ are related by

$$v = R\dot{\theta} \quad (5.158)$$

Solving for R in Equation 5.158 and substituting the result in Equation 5.157 give

$$-W \cos \theta = mv\dot{\theta} \quad (5.159)$$

With $W = mg$ and state variables v and $\dot{\theta}$, the state derivatives are obtained from Equations 5.155 and 5.159 as

$$\dot{v} = \begin{cases} 0, & 0 \leq y < y_1 \\ -\frac{F_D}{m} - g \sin \theta, & y \geq y_1 \end{cases} \quad (5.160)$$

$$\dot{\theta} = \begin{cases} 0, & 0 \leq y < y_1 \\ -\frac{g \cos \theta}{v}, & y \geq y_1 \end{cases} \quad (5.161)$$

The intervals $0 \leq y < y_1$ and $y \geq y_1$ correspond to before and after ejection.

Additional state variables x and y , the relative coordinates of the pilot and seat with respect to the moving aircraft, are needed to view its trajectory with respect to the plane in order to determine if it safely clears the plane's rear vertical stabilizer. The state derivatives are expressed as (see Figure 5.121)

$$\dot{x} = v \cos \theta - v_A \quad (5.162)$$

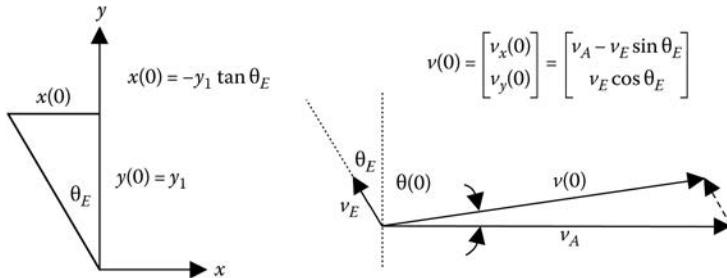


FIGURE 5.122 Initial states $x(0)$, $y(0)$, $v(0)$, and $\theta(0)$ at ejection ($t = 0$).

$$\dot{y} = v \sin \theta \quad (5.163)$$

It is convenient to start the simulation, that is, integrating the state derivatives, at the moment of ejection. The initial conditions are obtained with the help of Figure 5.122.

The initial states $v(0)$ and $\theta(0)$ are computed from

$$v(0) = [v_x^2(0) + v_y^2(0)]^{1/2} \quad (5.164)$$

$$\Rightarrow v(0) = [(v_A - v_E \sin \theta_E)^2 + (v_E \cos \theta_E)^2]^{1/2} \quad (5.165)$$

$$\theta(0) = \tan^{-1} \left[\frac{v_y(0)}{v_x(0)} \right] \quad (5.166)$$

$$\Rightarrow \theta(0) = \tan^{-1} \left(\frac{v_E \cos \theta_E}{v_A - v_E \sin \theta_E} \right) \quad (5.167)$$

Finally, the drag force F_D is obtained from

$$F_D = \frac{1}{2} C_D \rho A v^2 \quad (5.168)$$

where

C_D is the drag coefficient

ρ is the density of air

A is the surface area of the pilot and seat normal to the velocity vector

A simulation study is required to investigate the combinations of aircraft speed v_A and altitude h associated with safe ejection, that is, pilot and seat clear the rear vertical stabilizer by a predetermined amount. First, we shall simulate a single case where $v_A = 500$ ft/s and $h = 0$ (sea level). A Simulink diagram is shown in Figure 5.123.

Baseline numerical values of the system parameters are $\theta_E = 15^\circ$, $v_E = 40$ ft/s, $m = 8$ slugs, $A = 10$ ft², $C_D = 1$, and $y_1 = 4$ ft. The “Lookup Table” contains air density ρ (slug ft²) vs. altitude h (ft) data points from sea level to 60,000 ft.

The pilot and seat trajectory relative to the aircraft is obtained by calling the Simulink model “ejection_seat.mdl” from the M-file “Ch5_eject.m” using the command “sim('ejection_seat').” Figure 5.124 illustrates the relative separation between the pilot and seat combination and

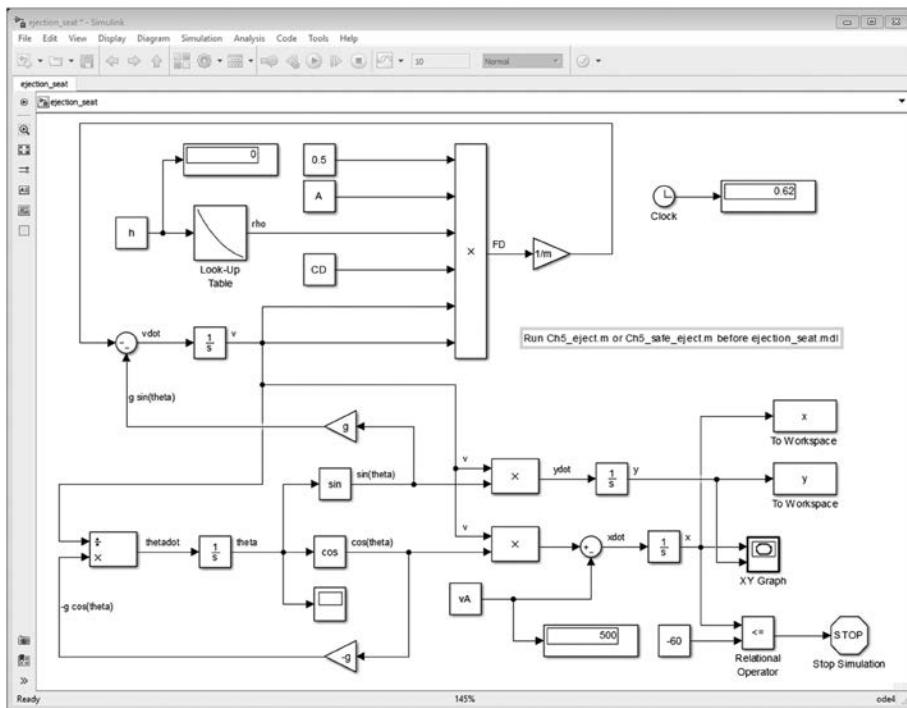


FIGURE 5.123 Simulink diagram of pilot ejection.

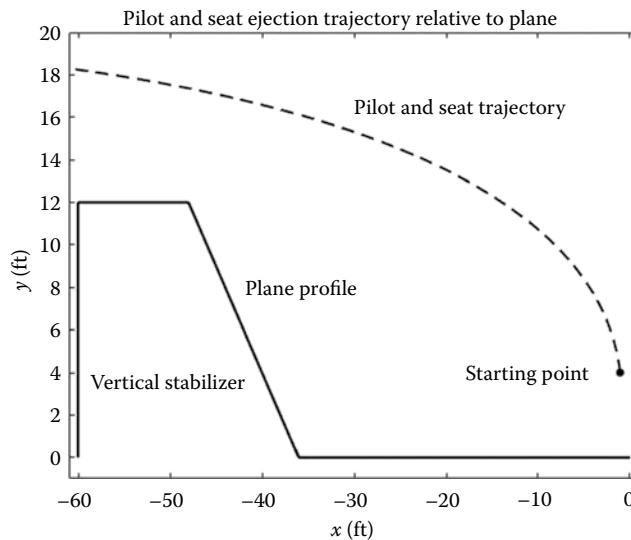


FIGURE 5.124 Plot of pilot and seat trajectory relative to the aircraft ($h = 0 \text{ ft}$, $v_A = 500 \text{ ft/s}$).

the plane during the time when the pilot and seat are located above the plane. The pilot and seat safely clear the vertical stabilizer.

At a given altitude h , the pilot and seat trajectory will safely clear the stabilizer provided the aircraft cruising speed v_A falls within a range of values. At slow speeds, the exit velocity is insufficient to propel the pilot and seat safely over the stabilizer, while at very high speeds, the excessive drag force and backward velocity (relative to the plane) produce a similar outcome.

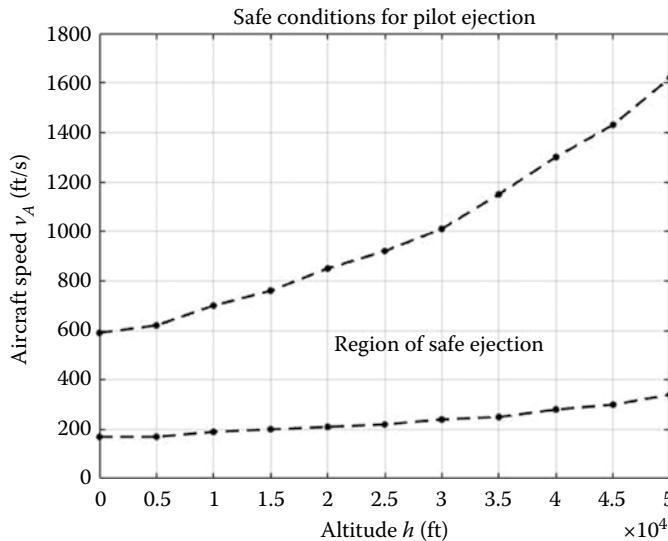


FIGURE 5.125 Lower and upper aircraft speeds at a given altitude for safe ejection.

A simulation study was performed to determine a region of safe ejection conditions, that is, altitude and speed combinations resulting in a clearance of 5 ft when the pilot and seat are directly over the back part of the rear stabilizer. The M-file “*Ch5_safe_eject.m*” calls the simulation model for altitudes from zero to 50,000 ft (in increments of 5,000 ft) and finds the range of aircraft speeds for a safe ejection. The result is shown in [Figure 5.125](#).

[Figure 5.126](#) shows a plot of $y(t)$, the height of the pilot and seat combination above the plane, corresponding to the safe ejection trajectory shown in [Figure 5.124](#). The lower graph shows $\theta(t)$, the angle between the velocity vector and the horizontal. Can you locate the point on each plot where the pilot and seat are located at the rear of the plane?

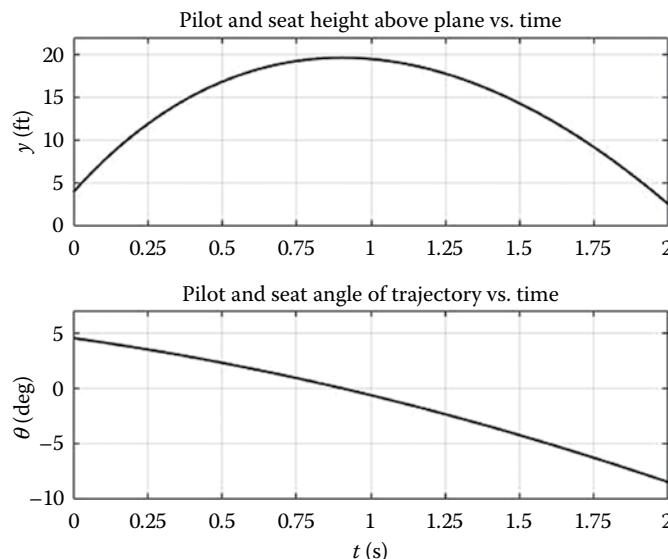


FIGURE 5.126 Pilot and seat height above plane and trajectory after ejection.

EXERCISES

- 5.50 With respect to the ballistic trajectory of the pilot and seat,
- Develop an alternate mathematical model using x , y coordinates. The states are x , \dot{x} , y , and \dot{y} .
 - Prepare a Simulink diagram for simulating the trajectory following ejection.
 - Run the simulation for the same conditions as in [Figure 5.124](#) and compare results.
 - Suppose the aircraft is cruising at 30,000 ft in level flight when ejection occurs. Simulate pilot and seat trajectories corresponding to $v_A = 500, 600, \dots, 1200$ ft/s. Plot the entire set of trajectories (with respect to the plane) on the same axes with the plane profile similar to [Figure 5.124](#). Are the results consistent with the safe ejection conditions portrayed in [Figure 5.125](#)?
- 5.51 Use either $n-t$ or $x-y$ coordinate systems to model the pilot and seat trajectory and obtain plots of
- x vs. t
 - y vs. t
 - θ vs. t
- when ejection occurs from 50,000 ft at a speed of 900 ft/s.
- 5.52 Reexamine the limiting plane speeds for a safe ejection from 25,000 ft as the mass of the pilot and seat varies from 8 slugs to 12 slugs. How important is the combined mass of the pilot and seat with respect to the limiting plane speeds at 25,000 ft?
- 5.53 Obtain new curves for lower and upper safe ejection speeds in terms of altitude if the criterion for a safe ejection is that the pilot and seat simply clear the rear vertical stabilizer. Use the baseline value for $m = 8$ slugs.
- 5.54 Modify the code in M-file “*Ch5_safe_eject.m*” to check whether the pilot and seat have cleared the rear stabilizer over its entire length of 48–60 ft back from the point of ejection. How does this affect the curves in [Figure 5.125](#)?

5.12 CASE STUDY: KALMAN FILTERING

Estimations of the Moon and planetary orbits were performed by early pioneers such as Kepler, Legendre, and Gauss. More recent estimation algorithms have been developed in an effort to obtain the optimal estimate of a dynamic object, the Kalman filter being the most popular. In this case study, the continuous-time Kalman filter, the steady-state Kalman filter, and the discrete-time Kalman filter are applied to the trajectory of an asteroid. First, the algorithms of the different filters will be presented in summary form, and then simulations will be run in Simulink for comparison.

5.12.1 CONTINUOUS-TIME KALMAN FILTER

The state equations of a continuous dynamic system are given by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} + \mathbf{w} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{v}\end{aligned}\tag{5.169}$$

where

- \mathbf{x} is the state vector
- \mathbf{u} is the input vector
- \mathbf{y} is the output vector
- \mathbf{A} is the system matrix

\mathbf{B} is the input matrix

\mathbf{C} is the output matrix

In the state equations, \mathbf{w} and \mathbf{v} are zero-mean, uncorrelated, continuous-time, white noise with process covariance matrix \mathbf{Q}_c and measurement covariance matrix \mathbf{R}_c , respectively. Mathematically,

$$\mathbf{w} \sim (0, \mathbf{Q}_c)$$

$$\mathbf{v} \sim (0, \mathbf{R}_c)$$

$$E[\mathbf{w}\mathbf{w}^T] = \mathbf{Q}_c \delta_{ij} \quad (5.170)$$

$$E[\mathbf{v}\mathbf{v}^T] = \mathbf{R}_c \delta_{ij}$$

$$E[\mathbf{v}\mathbf{w}^T] = 0$$

The algorithm of the continuous-time Kalman filter is given by

$$\mathbf{K} = \mathbf{P}\mathbf{C}^T \mathbf{R}_c^{-1}$$

$$\dot{\hat{\mathbf{x}}} = A\dot{\hat{\mathbf{x}}} + \mathbf{B}\mathbf{u} + \mathbf{K}(\mathbf{y} - \mathbf{C}\dot{\hat{\mathbf{x}}}) \quad (5.171)$$

$$\dot{\mathbf{P}} = -\mathbf{P}\mathbf{C}^T \mathbf{R}_c^{-1} \mathbf{C}\mathbf{P} + \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{Q}_c$$

where the last equation in 5.171 is referred to as the Riccati equation. The algorithm is initialized with the expectation values of the state and state covariance

$$\begin{aligned} \hat{\mathbf{x}}(0) &= E[\mathbf{x}(0)] \\ \mathbf{P}(0) &= E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T] \end{aligned} \quad (5.172)$$

5.12.2 STEADY-STATE KALMAN FILTER

In the case of the steady-state Kalman filter, the system dynamics do not change with respect to time; therefore, $\dot{\mathbf{P}} = 0$, so that the Riccati equation of 5.171 becomes

$$0 = -\mathbf{P}\mathbf{C}^T \mathbf{R}_c^{-1} \mathbf{C}\mathbf{P} + \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T + \mathbf{Q}_c \quad (5.173)$$

5.12.3 DISCRETE-TIME KALMAN FILTER

The state equations of a discrete dynamic system are given by

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \end{aligned} \quad (5.174)$$

where

- \mathbf{F}_{k-1} is the system matrix
- \mathbf{G}_{k-1} is the input matrix
- \mathbf{H}_{k-1} is the output matrix

In this case, \mathbf{w}_{k-1} and \mathbf{v}_{k-1} are zero-mean, uncorrelated, discrete-time, white noise with process covariance matrix \mathbf{Q}_k and measurement covariance matrix \mathbf{R}_k , respectively. Mathematically,

$$\begin{aligned}\mathbf{w}_k &\sim (0, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim (0, \mathbf{R}_k) \\ E[\mathbf{w}_k \mathbf{w}_j^T] &= \mathbf{Q}_k \delta_{k-j} \\ E[\mathbf{v}_k \mathbf{v}_j^T] &= \mathbf{R}_k \delta_{k-j} \\ E[\mathbf{w}_k \mathbf{v}_j^T] &= 0\end{aligned}\tag{5.175}$$

The algorithm of the discrete-time Kalman filter is given by

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^- + \mathbf{G}_{k-1} \mathbf{u}_{k-1} \\ \mathbf{P}_k^- &= \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T\end{aligned}\tag{5.176}$$

and is initialized with the expectation values of the state and state covariance

$$\begin{aligned}\hat{\mathbf{x}}_0^+ &= E[\mathbf{x}_0] \\ \mathbf{P}_0^+ &= E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+) (\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T]\end{aligned}\tag{5.177}$$

5.12.4 SIMULINK SIMULATIONS

The three different Kalman filters (continuous, steady-state, and discrete) are used to estimate the kinematics (position and velocity) of an incoming meteorite. It is assumed that the meteorite is tracked with a radar system that picks up the object at a range of 200,000 m with a velocity of 5000 m/s. The measurement error \mathbf{R} of the radar tracking station is 100 m. The process noise statistics \mathbf{Q} in range, velocity, and acceleration are 1 m, 0.1 m/s, and 0.1 m/s², respectively. Since the initial conditions of the meteorite are unknown, the diagonal elements of the state covariance matrix \mathbf{P} are large. The meteorite is tracked for 30 s at a frequency of 10 Hz.

Figure 5.127 shows a Simulink diagram for estimating the range of the meteorite with a continuous-time Kalman filter. (In most cases, element blocks retained their default names for ease of locating them in the Simulink library. A few subsystem names were changed to reflect their contents.) At the top of the continuous-time Kalman filter hierarchy, two major subsystems are shown: (1) the actual range of the meteorite corrupted by noise and (2) the estimated range containing the continuous-time Kalman filter elements. To run this model, execute the MATLAB M-file *Ch5_CTKF_Model_Data.m*.

By double clicking on the “Actual” subsystem, Figure 5.128 shows the elemental blocks that calculate the kinematics of the meteorite $y = y_0 + v_0 t + 1/2 a t^2$ and $v = v_0 + a t$ where

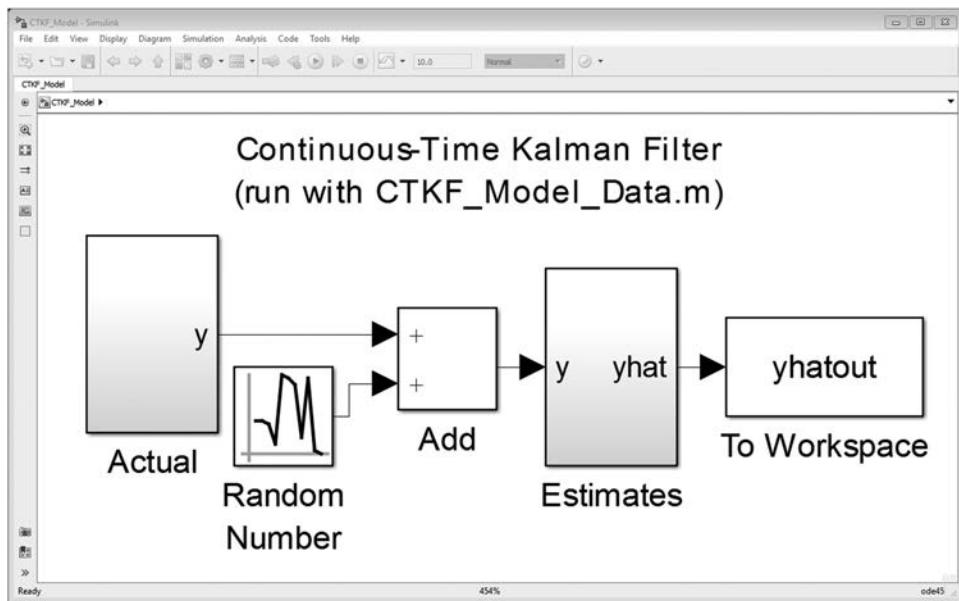


FIGURE 5.127 Top view of the continuous-time Kalman filter.

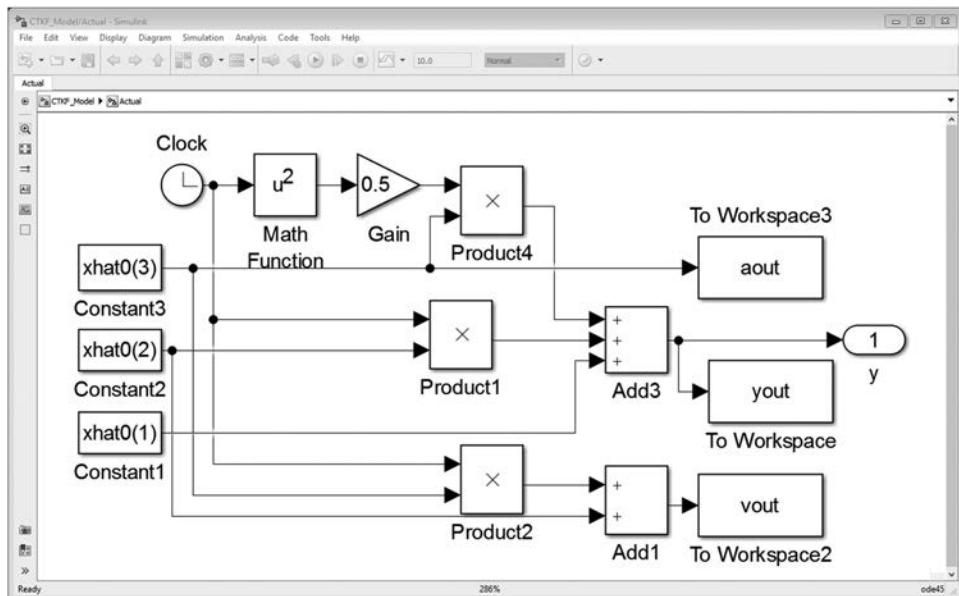


FIGURE 5.128 The “Actual” subsystem.

the initial conditions are represented by $xhat0$, a vector defined in the MATLAB M-file “*Ch5_DTKF_Model_Data.m*”.

Returning to the top-level view and then double clicking on the “Estimates” subsystem, Figure 5.129 shows the elemental blocks of the continuous-time Kalman filter algorithm, Equation 5.171. The integrator block requires the initial conditions $xhat0$ defined in the MATLAB M-file. For legibility, the computation of the state covariance matrix P is placed into its own subsystem.

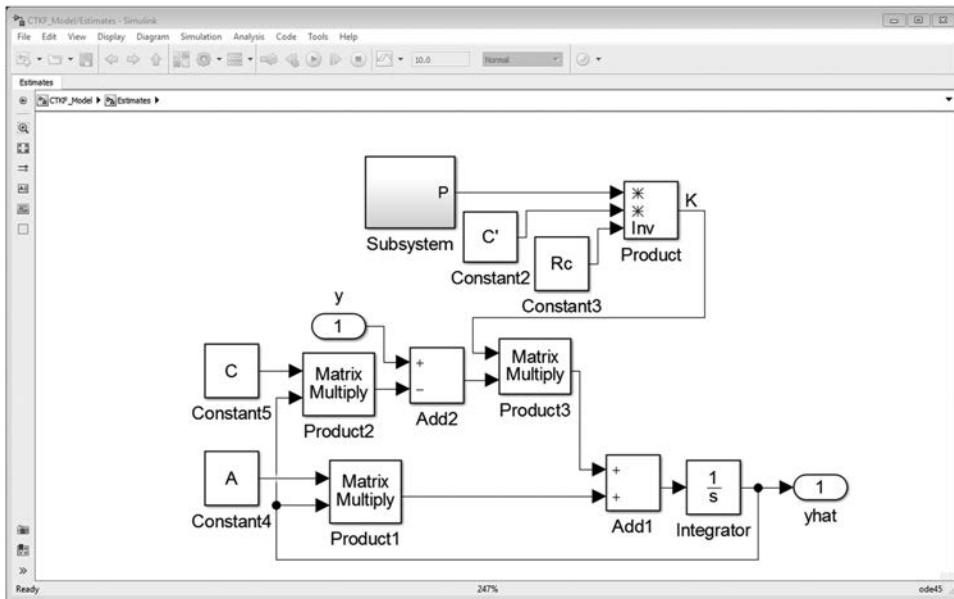


FIGURE 5.129 The continuous-time Kalman filter algorithm.

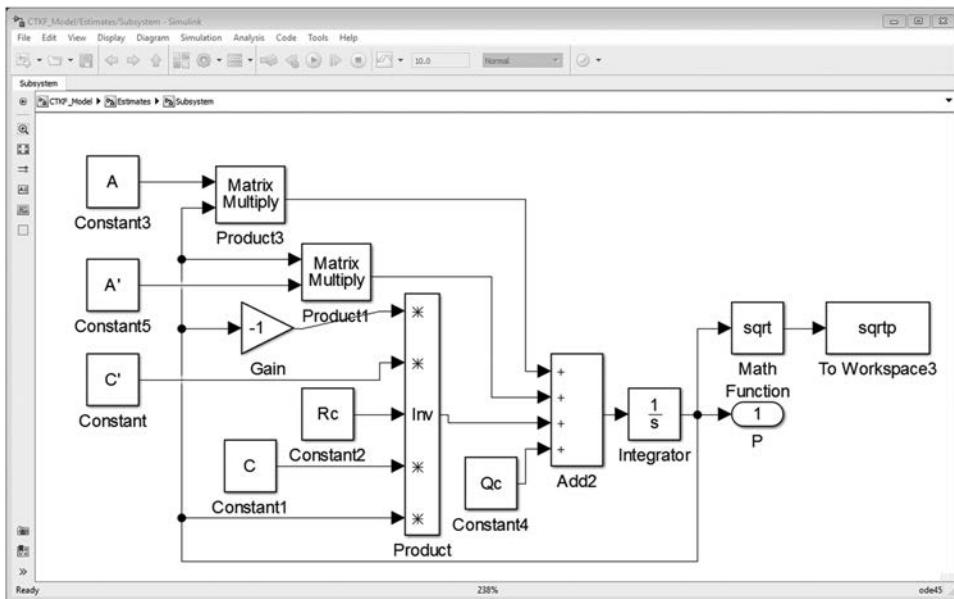


FIGURE 5.130 Simulink diagram of the continuous-time Kalman filter.

By double clicking on the “P” subsystem, Figure 5.130 shows the elemental blocks that update the state covariance matrix P , Equation 5.171. The integrator in this subsystem requires the initial conditions P_0 defined in the M-file.

Simulating the model by executing the MATLAB M-file *Ch5_CTKF_Model_Data.m* created the following plots. Figure 5.131 shows the actual range R and the estimated range R_{hat} of the meteorite vs. time. The meteorite is picked up at a range of 200,000 m and tracked for 30 s. Over this time period, the meteorite traveled approximately 150,000 m. The continuous-time Kalman

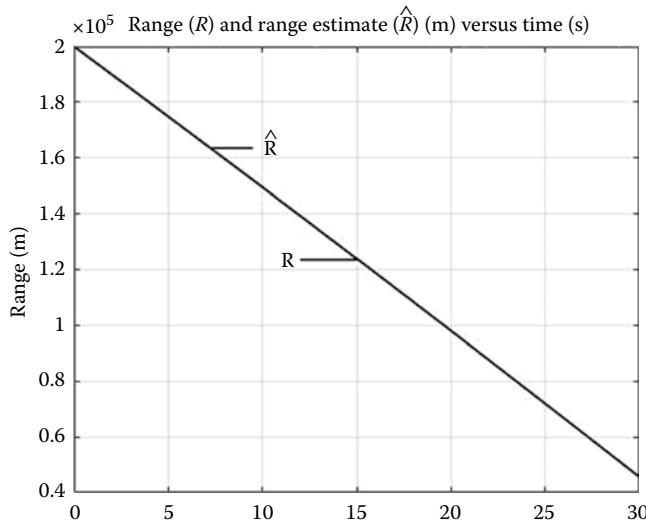


FIGURE 5.131 Plot of range and range estimates (m) vs. time (s).

filter performs very well, such that it is difficult to see any differences between the actual range and the estimated range.

Figure 5.132 shows the actual velocity V and the estimated velocity \hat{V} of the meteorite vs. time. The continuous-time Kalman filter takes approximately 10 s for transients to settle before obtaining reasonable velocity estimates.

Figure 5.133 shows the actual acceleration A and the estimated acceleration \hat{A} of the meteorite vs. time. It is unnecessary to estimate the acceleration of gravity, but it is shown here for completeness. Again, the transients take approximately 10 s to settle before obtaining reasonable estimates.

Figure 5.134 shows the range error, the difference between the actual range and the estimated range, vs. time. In theory, the range error should be bounded by the standard deviation of the 1,1

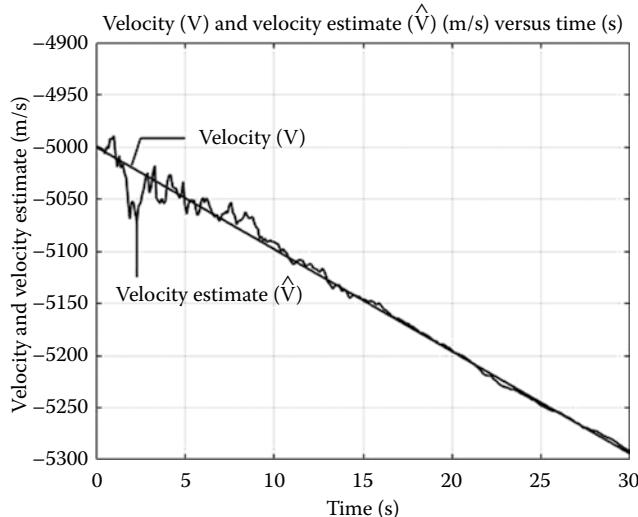


FIGURE 5.132 Plot of velocity and velocity estimates (m/s) vs. time (s).

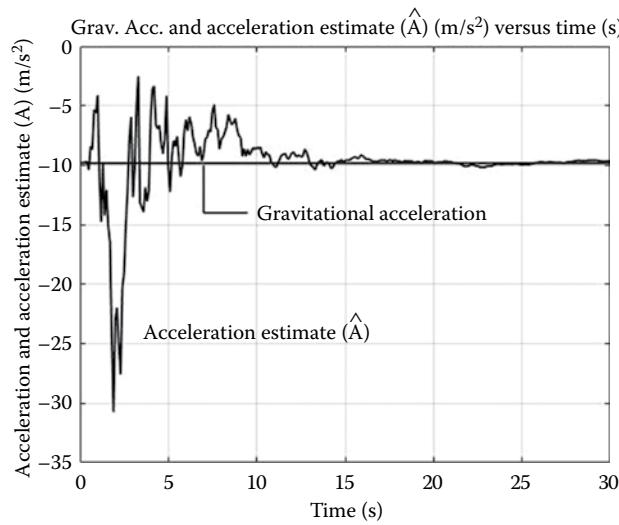


FIGURE 5.133 Plot of acceleration and acceleration estimates (m/s/s) vs. time (s).

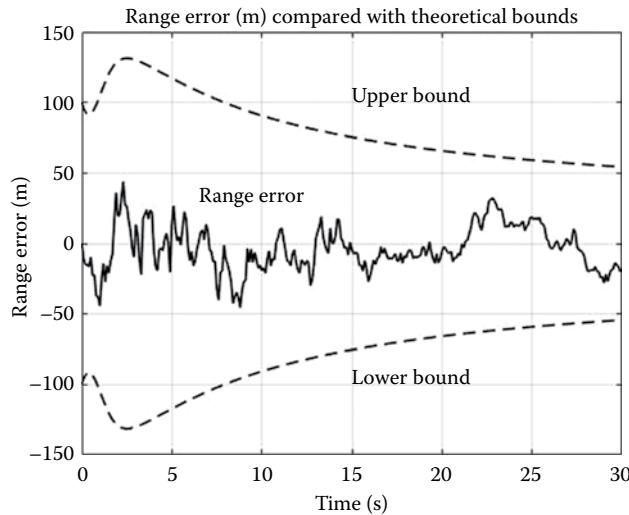


FIGURE 5.134 Plot of range error vs. time.

element of the state covariance matrix, which it is. It appears as if the maximum range error at any given time is about 50 m. Recall (Figure 5.131) that the meteorite traveled roughly 150,000 m over 30 s. An error of 50 m, even at the end of the 30 s when the meteorite is at a range of 50,000 m, is 0.1%.

Figure 5.135 shows the velocity error, the difference between the actual velocity and the estimated velocity, vs. time. In this case, the velocity error should be bounded by the standard deviation of the 2,2 element of the state covariance matrix, which it is. After the filter transients settle out, the maximum velocity error appears to be less than 10 m/s. Recall (Figure 5.132) that the meteorite obtained a speed of roughly 5300 m/s over 30 s. An error of 10 m/s is less than 0.2%.

This concludes the implementation and analysis of the continuous-time Kalman filter as applied to the range and velocity estimates of an incoming meteorite.

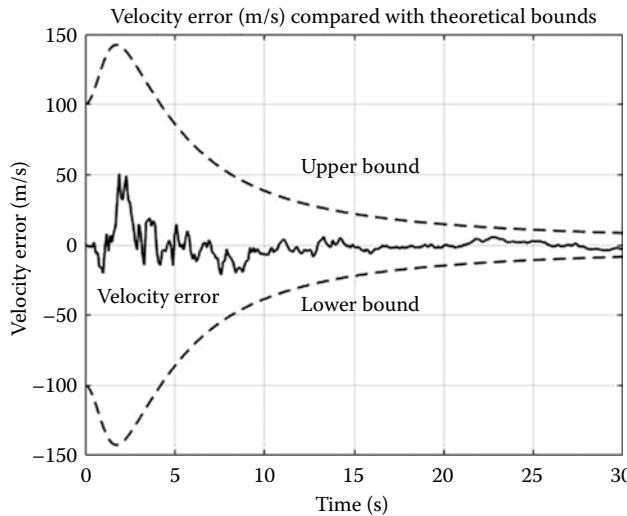


FIGURE 5.135 Plot of velocity error vs. time.

Next, the steady-state Kalman filter is applied to the same problem for comparison with the continuous-time Kalman filter. The only difference between the two models is the calculation of the state covariance matrix P . In the continuous-time algorithm, the Riccati equation is time dependent; for the steady-state algorithm, the Riccati equation is independent of time, Equation 5.173. With regard to model structure, the top-level diagram and “Actual” subsystem diagram are the same for the steady-state Kalman filter as they were for the continuous-time Kalman filter. However, the “Estimates” subsystem reflects the difference with regard to the Riccati equation, which is represented by a constant element block called “SSP” seen in Figure 5.136.

Simulating the model by executing the MATLAB M-file SSCTKF_Model_Data.m created the following plots. Figure 5.137 shows the actual range R and the estimated range $Rhat$ of the meteorite vs. time. From this plot, it appears that the steady-state Kalman filter performs just as well as the continuous-time Kalman filter. As before, it is difficult to see any differences between the actual range and the estimated range.

Figure 5.138 shows the actual velocity V and the estimated velocity $Vhat$ of the meteorite vs. time. From this plot, it can be seen that the steady-state Kalman filter performs better than the continuous filter in estimating the velocity of the meteorite. Obviously missing from this plot are the transients associated with the time-dependent state covariance updates. The steady-state Kalman filter eliminates the need to perform this calculation—which may be significant for an application where real-time processing is limited.

Figure 5.139 shows the actual acceleration A and the estimated acceleration $Ahat$ of the meteorite vs. time. As mentioned before, it is unnecessary to estimate the acceleration of gravity, but it is shown for completeness. Again, there are no transients with the steady-state Kalman filter.

Figure 5.140 shows the range error, the difference between the actual range and the estimated range, vs. time. Again, the range error is bounded by the standard deviation of the 1,1 element of the state covariance matrix, which is constant. The maximum range error at any given time is negligible for the steady-state Kalman filter.

Figure 5.141 shows the velocity error, the difference between the actual velocity and the estimated velocity, vs. time. The velocity error is bounded by the standard deviation of the 2,2 element of the state covariance matrix, which is constant. Here, too, the maximum velocity error at any given time is negligible for the steady-state Kalman filter.

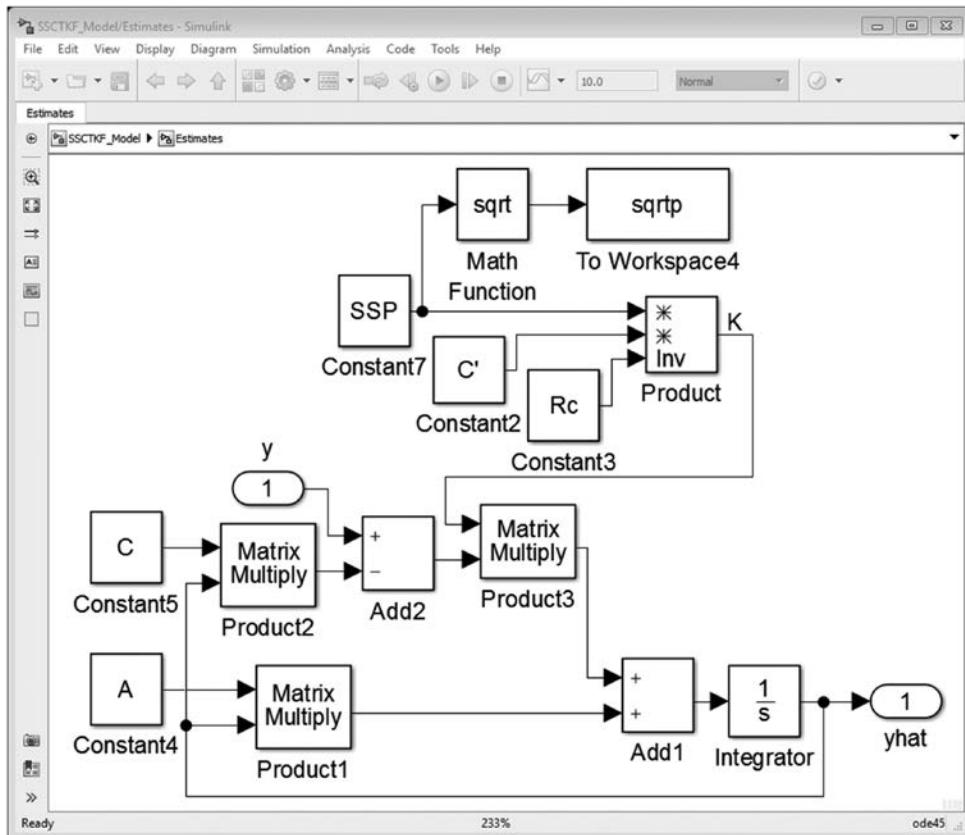


FIGURE 5.136 The steady-state Kalman filter algorithm.

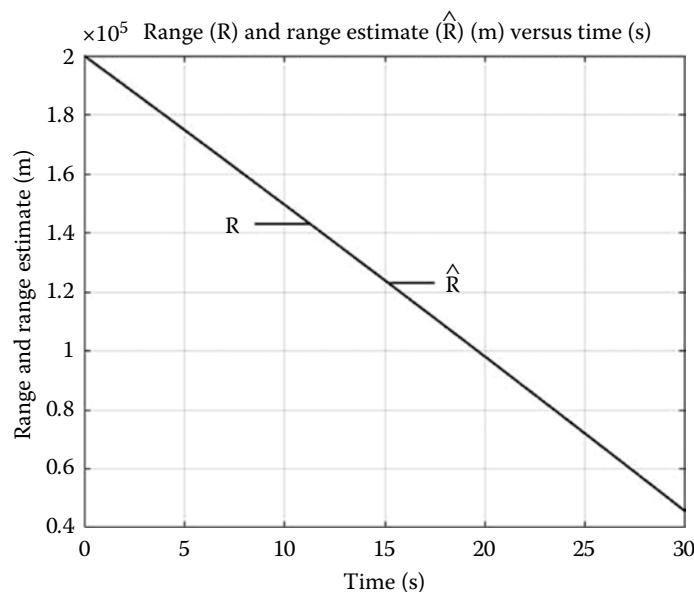


FIGURE 5.137 Plot of range and range estimates (m) vs. time (s).

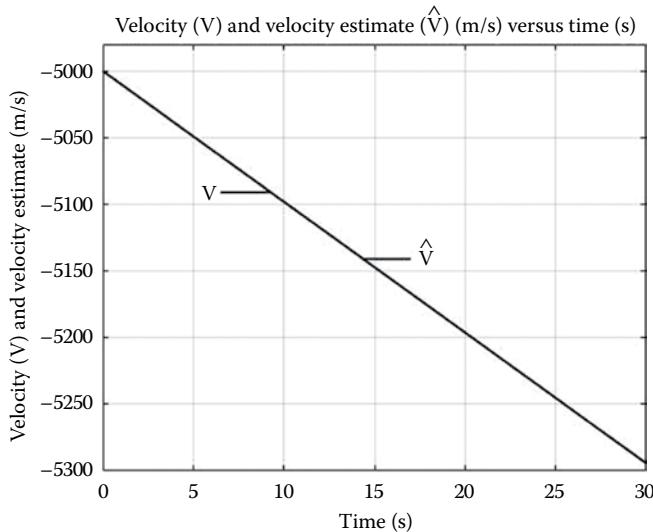


FIGURE 5.138 Plot of velocity and velocity estimates (m/s) vs. time (s).

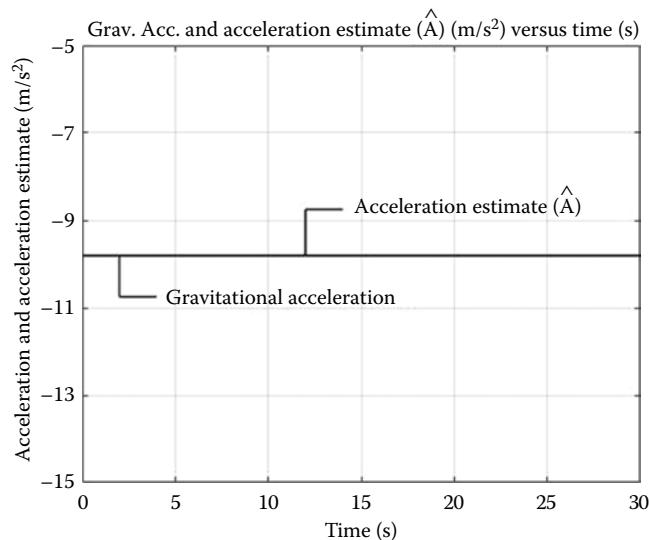


FIGURE 5.139 Plot of acceleration and acceleration estimates (m/s/s) vs. time (s).

This concludes the implementation and analysis of the steady-state Kalman filter as applied to the range and velocity estimates of an incoming meteorite.

Next, the discrete-time Kalman filter is applied to the same problem for comparison with the continuous-time Kalman filter. The dynamic system of the meteorite kinematics are discretized, Equation 5.174, and then simulated with the discrete-time Kalman filter algorithm, Equation 5.176. At this time, a few comments regarding the algorithm are in order. The first two equations of the algorithm \hat{x}_k^- and P_k^- are known as the a priori state and state covariance estimates, respectively. They take the name “a priori” because the calculations are performed *before* the meteorite’s state is measured. The third equation of the algorithm K_k is the Kalman gain. The last two equations of the algorithm \hat{x}_k^+ and P_k^+ are known as the a posteriori state and state covariance estimates,

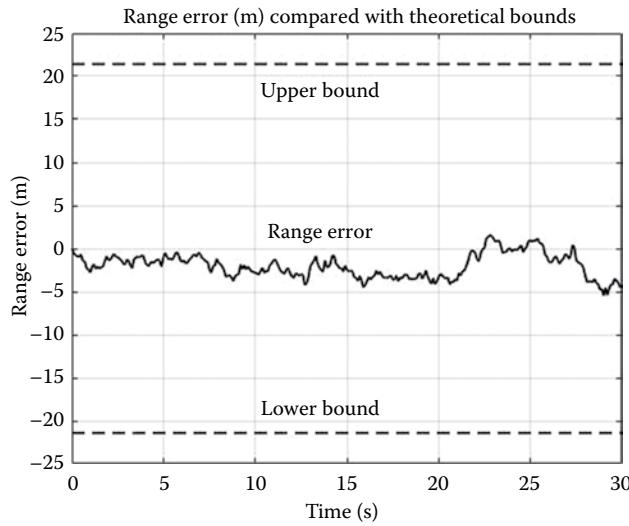


FIGURE 5.140 Plot of range error vs. time.

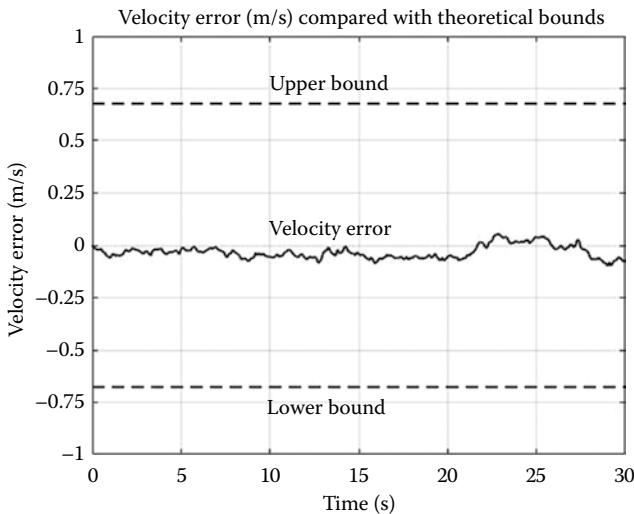


FIGURE 5.141 Plot of velocity error vs. time.

respectively. They take the name “a posteriori” because the calculations are performed *after* the meteorite’s state is measured.

As in the previous two cases, the top-level diagram and “Actual” subsystem diagram are the same for the discrete-time Kalman filter. However, the “Estimates” subsystem, shown in Figure 5.142, shows the Simulink diagram for the discrete-time Kalman filter algorithm. From this view, the a priori state and state covariance, the Kalman gain, and the a posteriori state and state covariance subsystems are clearly represented.

By double clicking on the “a priori state” subsystem, Figure 5.143 shows the elemental blocks that calculate the a priori state estimate of the algorithm. The initial conditions are represented by $xm0$, a vector defined in the corresponding MATLAB M-file.

Returning to the top-level view and then double clicking on the “a priori covariance” subsystem, Figure 5.144 shows the elemental blocks that calculate the a priori state covariance estimate of the

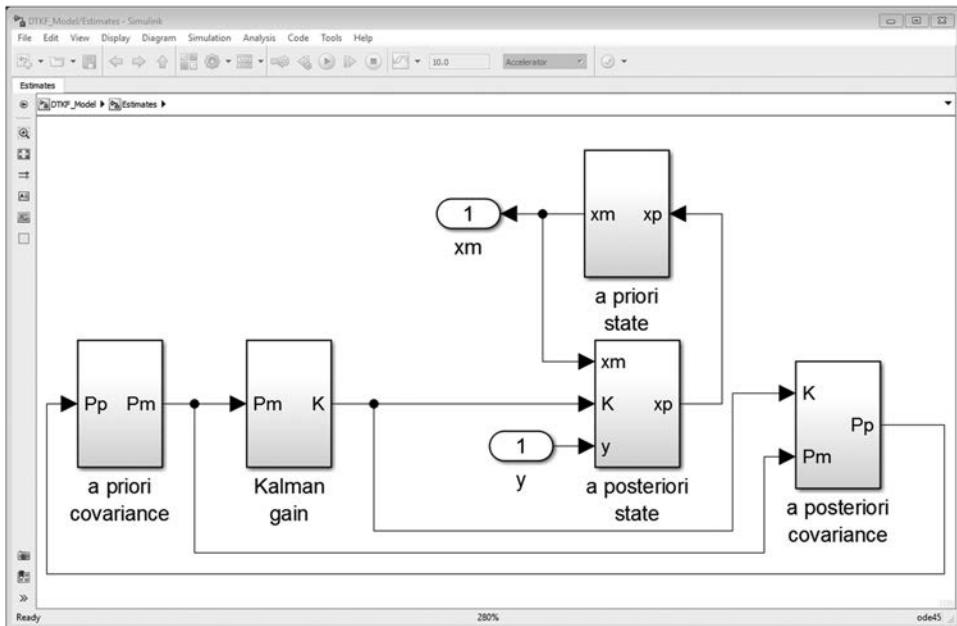


FIGURE 5.142 The discrete-time Kalman filter algorithm.

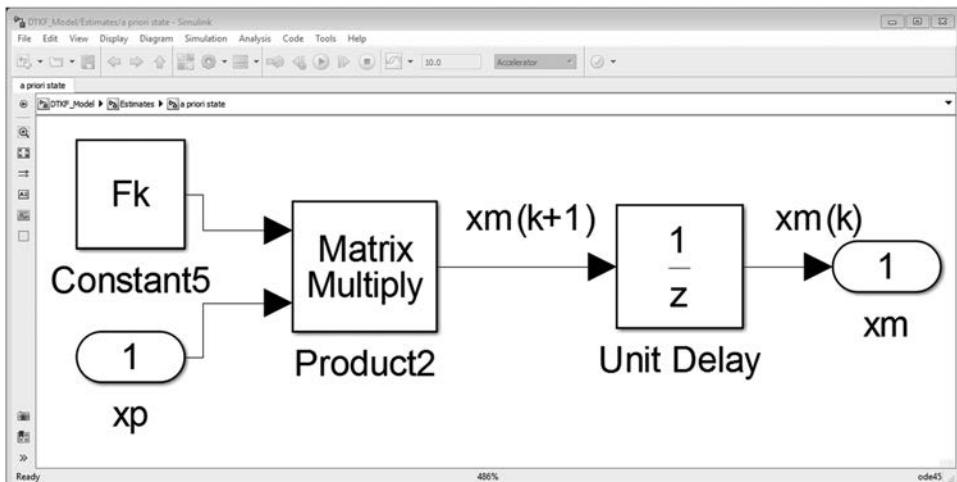


FIGURE 5.143 The “a priori state” subsystem.

algorithm. The initial conditions are represented by P_{m0} , a matrix defined in the corresponding MATLAB M-file.

Returning to the top-level view and then double clicking on the “Kalman gain” subsystem, [Figure 5.145](#) shows the elemental blocks that calculate the Kalman gain of the algorithm.

By double clicking on the “a posteriori state” subsystem, [Figure 5.146](#) shows the elemental blocks that calculate the a posteriori state estimate of the algorithm.

Returning to the top-level view and then double clicking on the “a posteriori covariance” subsystem, [Figure 5.147](#) shows the elemental blocks that calculate the a posteriori state covariance estimate of the algorithm.

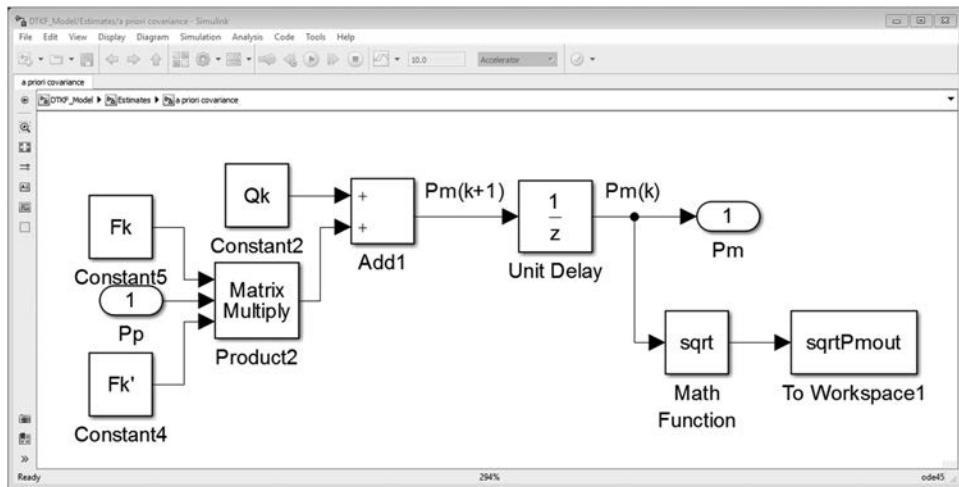


FIGURE 5.144 The “a priori covariance” subsystem.

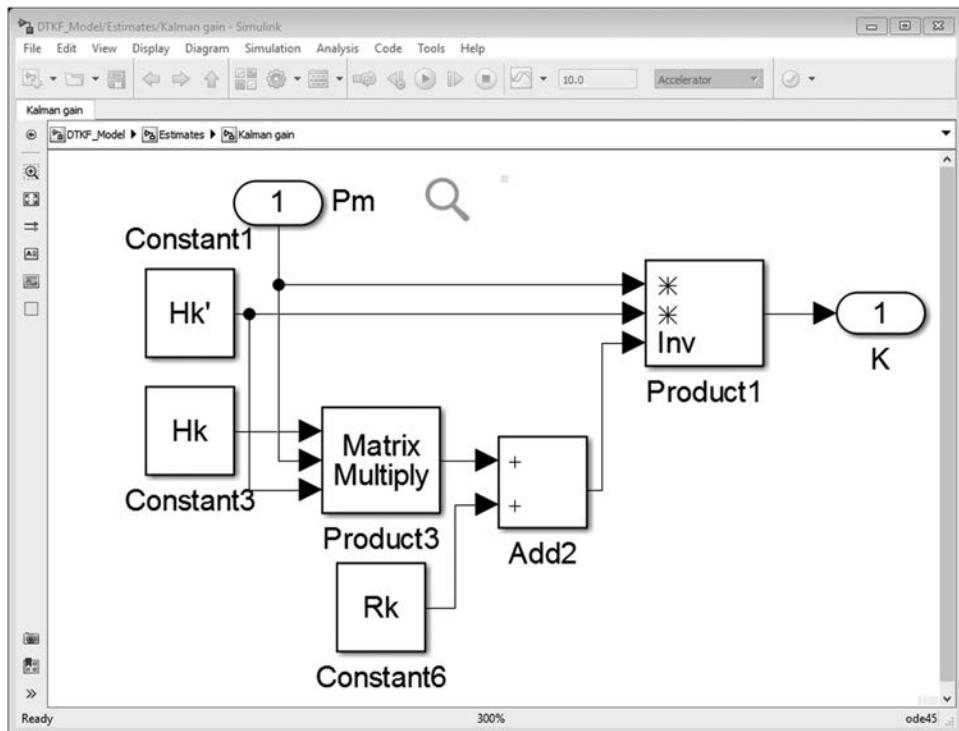


FIGURE 5.145 The “Kalman gain” subsystem.

Simulating the model by executing the MATLAB M-file Ch5_DTKF_Model_Data.m created the following plots. Figure 5.148 shows the actual range R and the estimated range R_{hat} of the meteorite vs. time. The meteorite is picked up at a range of 200,000 m and tracked for 30 s. Over this time period, the meteorite traveled approximately 150,000 m. Like the previous two filters, the discrete-time Kalman filter performs very well. Indeed, it is difficult to see any differences between the actual range and the estimated range.

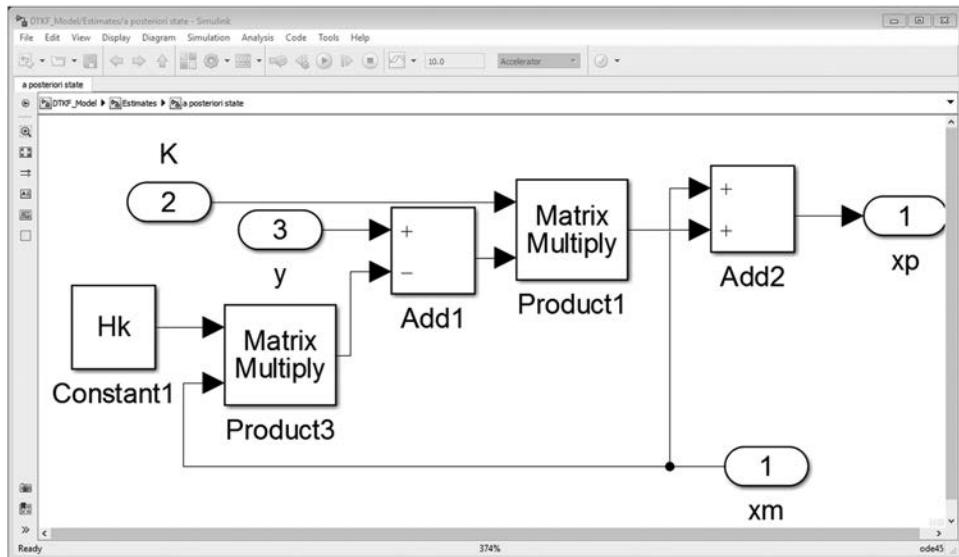


FIGURE 5.146 The “a posteriori state” subsystem.

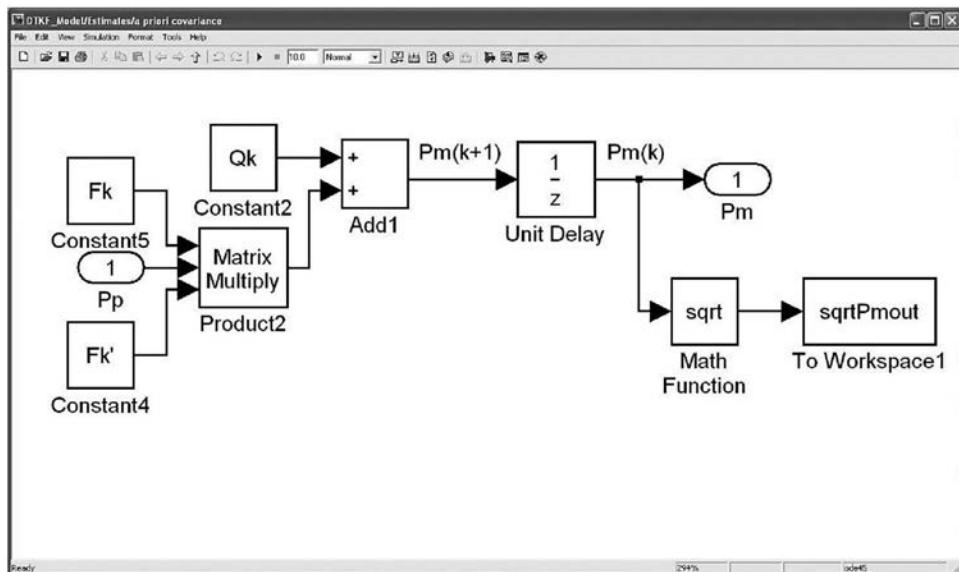


FIGURE 5.147 The “a posteriori covariance” subsystem.

Figure 5.149 shows the actual velocity V and the estimated velocity \hat{V} of the meteorite vs. time. The discrete-time Kalman filter takes approximately 10 s for transients to settle before obtaining reasonable velocity estimates. This is similar to the behavior of the continuous-time Kalman filter.

Figure 5.150 shows the actual acceleration A and the estimated acceleration \hat{A} of the meteorite vs. time. The transients take approximately 15 s to settle before obtaining reasonable estimates, 5 s more than the continuous-time Kalman filter.

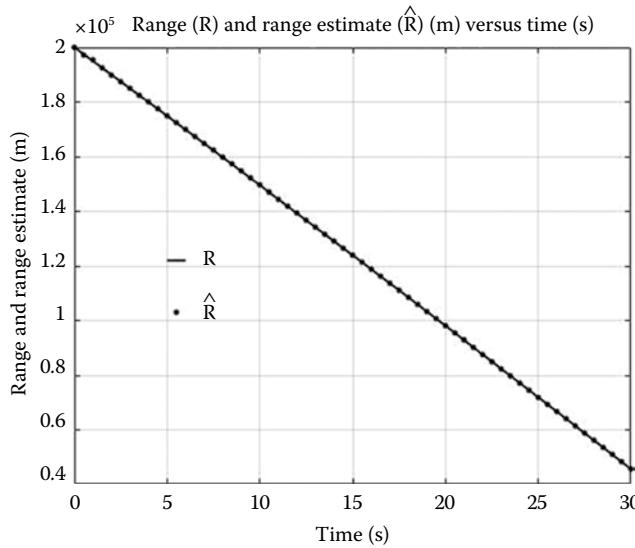


FIGURE 5.148 Plot of range and range estimates (m) vs. time (s).

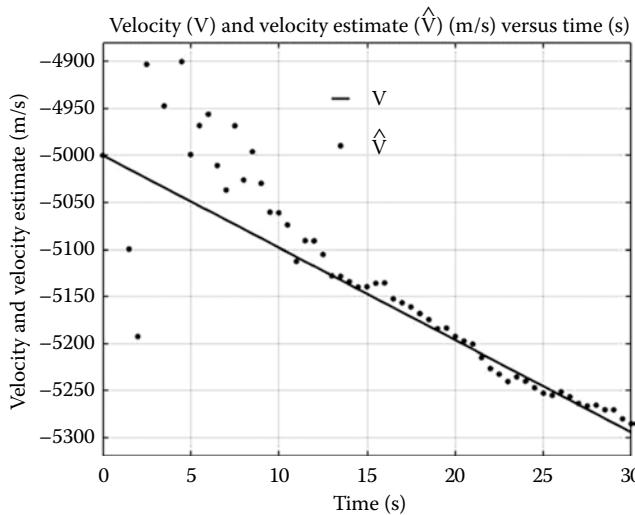


FIGURE 5.149 Plot of velocity and velocity estimates (m/s) vs. time (s).

Figure 5.151 shows the range error, the difference between the actual range and the estimated range, vs. time. In theory, the range error should be bounded by the standard deviation of the 1,1 element of the state covariance matrix. For the discrete-time Kalman filter, a few data points lie outside this theoretical limit, but only marginally. Recall (Figure 5.149) that the meteorite traveled roughly 150,000 m over 30 s. An error of 100 m, even at the end of the 30 s when the meteorite is at a range of 50,000 m, is 0.2%.

Figure 5.152 shows the velocity error, the difference between the actual velocity and the estimated velocity, vs. time. Again, in theory, the velocity error should be bounded by the standard

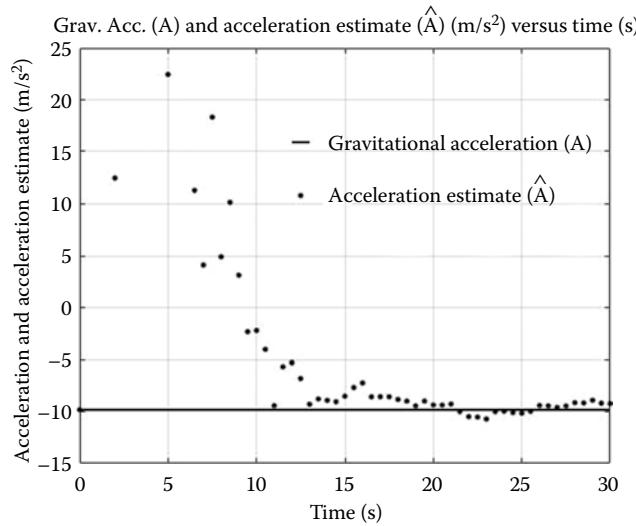


FIGURE 5.150 Plot of acceleration and acceleration estimates ($\text{m}/\text{s}/\text{s}$) vs. time (s).

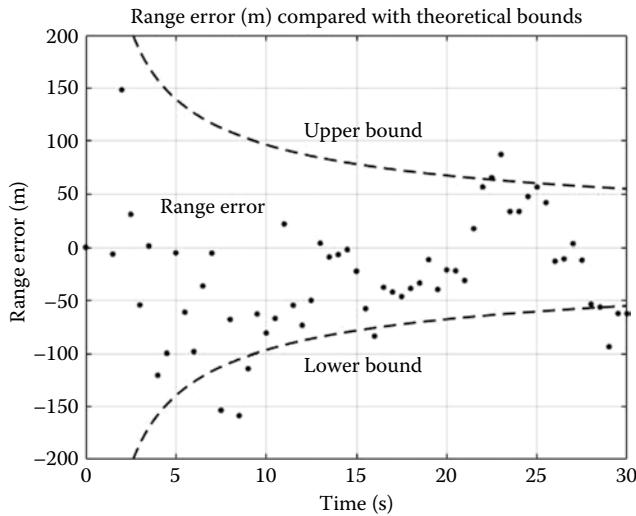


FIGURE 5.151 Plot of range error vs. time.

deviation of the 2,2 element of the state covariance matrix. After the discrete-time Kalman filter transients settle out, the maximum velocity error appears to be less than 10 m/s. Recall (Figure 5.149) that the meteorite obtained a speed of roughly 5300 m/s over 30 s. An error of 10 m/s is less than 0.2%.

5.12.5 SUMMARY

Three different Kalman filters (continuous, steady-state, and discrete) were used to estimate the kinematics (position and velocity) of an incoming meteorite. Once filter transients settled out, both the continuous-time and discrete-time Kalman filters provided acceptable results with regard to meteorite range and velocity estimation as evidenced by comparing the range and velocity errors

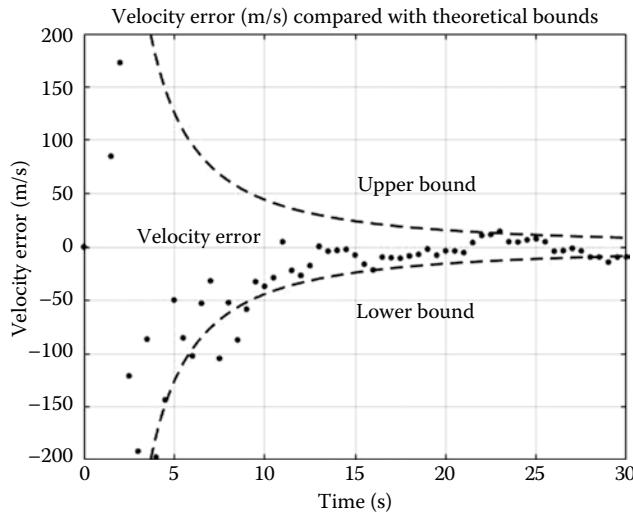


FIGURE 5.152 Plot of velocity error vs. time.

with actual range and velocity magnitudes. If real-time processing poses limitations, it is recommended to use the steady-state Kalman filter.

EXERCISES

- 5.55 Develop the steady-state Kalman filter for the discrete model.

Hint: Combine the a priori and the a posteriori equations into a single equation and note in the steady-state, $\hat{x}_k = \hat{x}_{k-1} = \hat{x}$, $P_k^- = P_{k-1}^- = P^-$ in the a priori case or $\hat{x}_k^+ = \hat{x}_{k-1}^+ = \hat{x}^+ P_k^+ = P_{k-1}^+ = P^+$ in the a posteriori case.

5.13 CASE STUDY: CASCADED TANKS WITH FLOW LOGIC CONTROL

The system shown below consists of two tanks arranged in a series configuration (Figure 5.153).

The first tank has two input flow sources.

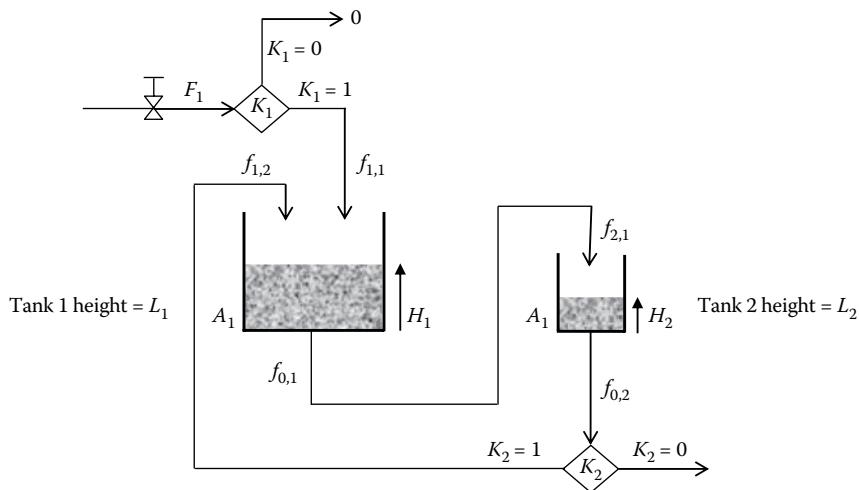


FIGURE 5.153 Cascaded tanks.

1. The external input $f_{1,1}$ which is zero if the parameter $K_1 = 0$, or a flowrate F_1 when $K = 1$,

$$f_{1,1} = \begin{cases} 0, & K_1 = 0 \\ F_1, & K_1 = 1 \end{cases} \quad (5.178)$$

The constant flowrate F_1 is regulated by a control valve which is in the wide open position over the time interval $t_0 \leq t \leq t_1$ and closed at all other times. A constant flow of magnitude “A” occurs when the valve is in the open position. Hence,

$$F_1 = \begin{cases} 0, & t < t_0 \\ A, & t_0 \leq t \leq t_1 \\ 0, & t > t_1 \end{cases} \quad (5.179)$$

2. A second input $f_{1,2}$ which depends on the value of parameter K_2 according to

$$f_{1,2} = \begin{cases} 0, & K_2 = 0 \\ f_{0,2}, & K_2 = 1 \end{cases} \quad (5.180)$$

where $f_{0,2}$ is the outflow from the bottom of the second tank.

The flowrates $f_{0,1}$ and $f_{0,2}$ from the bottom of Tank 1 and 2, respectively are

$$f_{0,1} = c_1 H_1^{1/2}, \quad (5.181)$$

$$f_{0,2} = c_2 H_2^{1/2} \quad (5.182)$$

The flow into the top of the second tank is the discharge flow from the bottom of the first tank,

$$f_{2,1} = f_{0,1} \quad (5.183)$$

Setting $f_1 = f_{1,1} + f_{1,2}$ the overflow from each tank (not shown in figure) is

$$f_{1,s} = \begin{cases} 0, & H_1 < L_1 \\ f_1 - (f_{0,1})_{\max}, & H_1 = L_1 \quad \text{where } (f_{0,1})_{\max} = c_1 L_1^{1/2} \end{cases} \quad (5.184)$$

$$f_{2,s} = \begin{cases} 0, & H_2 < L_2 \\ f_{2,1} - (f_{0,2})_{\max}, & H_2 = L_2 \quad \text{where } (f_{0,2})_{\max} = c_2 L_2^{1/2} \end{cases} \quad (5.185)$$

Baseline parameter values are:

$$A_1 = 4 \text{ ft}^2, A_2 = 4 \text{ ft}^2$$

$$L_1 = 20 \text{ ft}, L_2 = 10 \text{ ft}$$

$$c_1 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, c_2 = 2 \text{ ft}^3/\text{min per ft}^{1/2}$$

$$H_1(0) = 0 \text{ ft}, H_2(0) = 0 \text{ ft}$$

$$t_0 = 0 \text{ min}, t_1 = 20 \text{ min}$$

$$A = 15 \text{ ft}^3/\text{min}$$

$$K_1 = 1, K_2 = 0$$

A Simulink diagram for simulating the cascaded tank system dynamics is shown in Figure 5.154. Scopes for viewing the variables $f_{1,1}$, $f_{1,2}$, $f_1 f_{0,1}$, $f_{2,1}$, $f_{0,2}$, $f_{1,s}$, $f_{2,s}$, H_1 , H_2 are included. In addition, blocks are present for generating

1. $V_{1,in}$: the cumulative flow (ft^3) into Tank 1 from the external source from $t = 0$ to $t = t_{final}$
2. $V_{1,spill}$: the cumulative overflow (ft^3) from Tank 1 from $t = 0$ to $t = t_{final}$
3. $V_{2,spill}$: the cumulative overflow (ft^3) from Tank 2 from $t = 0$ to $t = t_{final}$
4. $V_{2,out}$: the cumulative flow (ft^3) out of Tank 2 not returned to Tank 1 from $t = 0$ to $t = t_{final}$

The Simulink model “cascaded_tanks.mdl” is called from the MATLAB script file “Ch5_call_cascaded_tanks.m” using the baseline values and ‘ode1 (Euler)’ integrators with a step size of 0.005 s. Simulation results are presented in Figures 5.155–5.157.

Note how the level in the first tank increases until it reaches the tank height of $L_1 = 20 \text{ ft}$. The tank level $H_1(t)$ and outflow $F_{0,1}(t)$ begin decreasing simultaneous with the cessation of inflow $f_1(t)$ at $t = 20 \text{ s}$.

Note the level $H_2(t)$ rising to the tank height of 10 ft and remaining there until the outflow $f_{0,2}(t)$ equals the inflow $f_{0,1}(t)$, at which time the level begins to fall.

Note the eventual emptying of both tanks as a result of the outflow from the second tank leaving the system as opposed to recycling back to the first tank.

Figure 5.158 depicts the tank levels $H_1(t)$ and $H_2(t)$ for the same baseline conditions with the exception of $K_2 = 1$ which makes the outflow from the second tank an input to the first tank. The tanks are now interacting as a result of the recycled flow.

Note the steady-state levels of both tanks $H_1(\infty) = H_2(\infty) = L_2 = 10 \text{ ft}$. With 40 ft^3 of fluid in each tank, the spillover from both tanks is the difference between the 300 ft^3 of fluid from the external source and the 80 ft^3 of fluid in the tanks at steady-state. According to the simulation, the spillover amounts are 159.5 ft^3 and 60.5 ft^3 from Tanks 1 and 2, respectively.

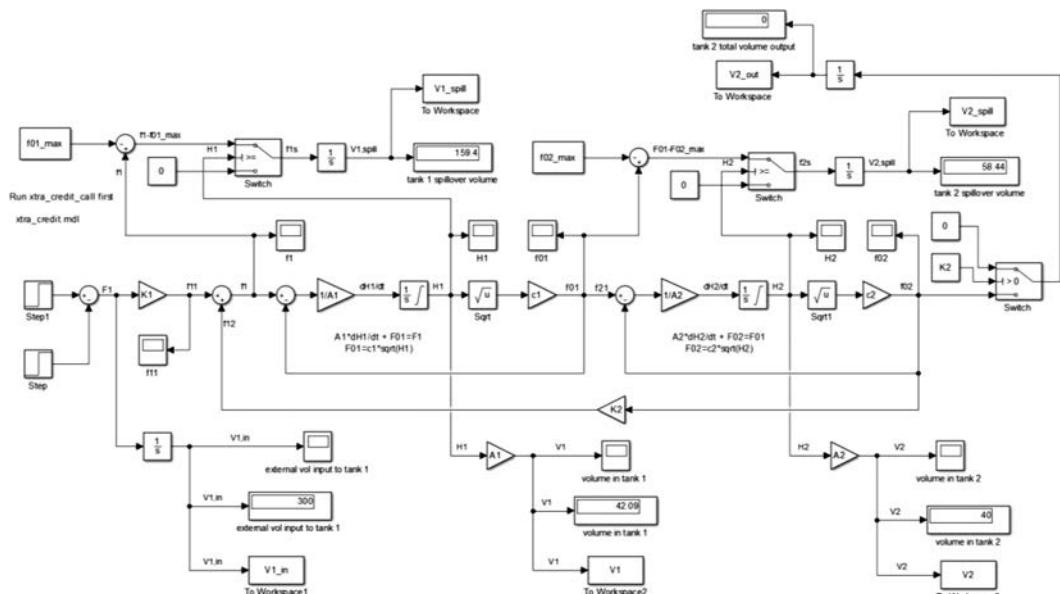


FIGURE 5.154 Simulink diagram for simulation of cascaded tanks.

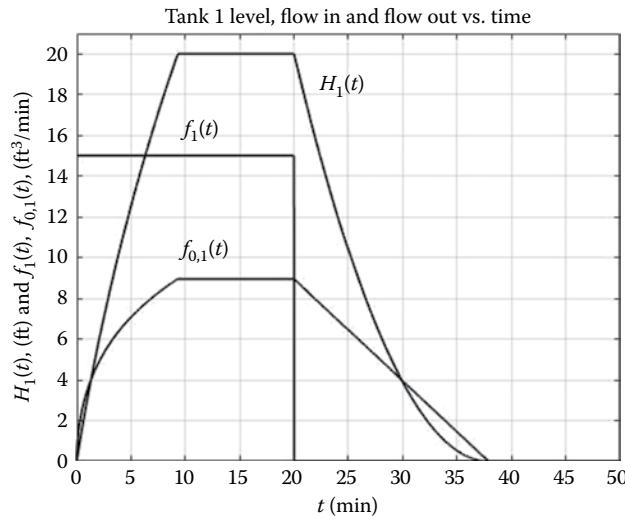


FIGURE 5.155 $H_1(t)$, $f_1(t)$, and $f_{0,1}(t)$ vs. t .

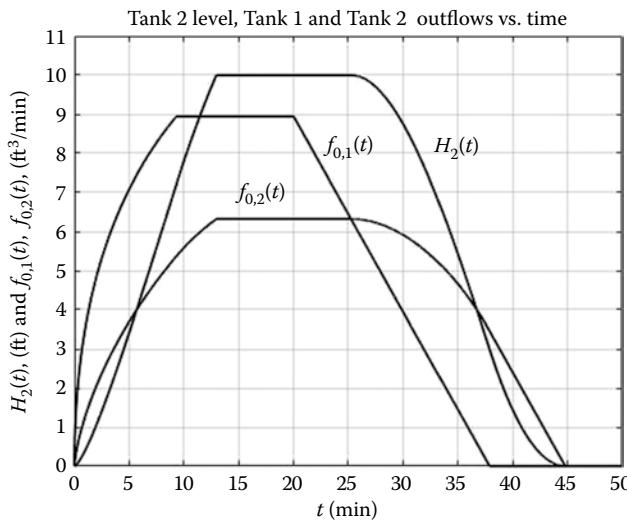


FIGURE 5.156 $H_2(t)$, $f_{0,1}(t)$ and $f_{0,2}(t)$ vs. t .

Additional simulations with parameter values listed in Cases I–V below were run for a period of 30 min, less than the time required for steady-state to be achieved.

Case I:

$$\begin{aligned}A_1 &= 4 \text{ ft}^2, L_1 = 20 \text{ ft}, c_1 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_1(0) = 0 \text{ ft}, \\A_2 &= 4 \text{ ft}^2, L_2 = 10 \text{ ft}, c_2 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_2(0) = 0 \text{ ft} \\t_0 &= 0 \text{ min}, t_1 = 20 \text{ min}, A = 8 \text{ ft}^3/\text{min} \\K_1 &= 1, K_2 = 0\end{aligned}$$

Case II:

$$\begin{aligned}A_1 &= 4 \text{ ft}^2, L_1 = 20 \text{ ft}, c_1 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_1(0) = 0 \text{ ft}, \\A_2 &= 4 \text{ ft}^2, L_2 = 10 \text{ ft}, c_2 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_2(0) = 0 \text{ ft}\end{aligned}$$

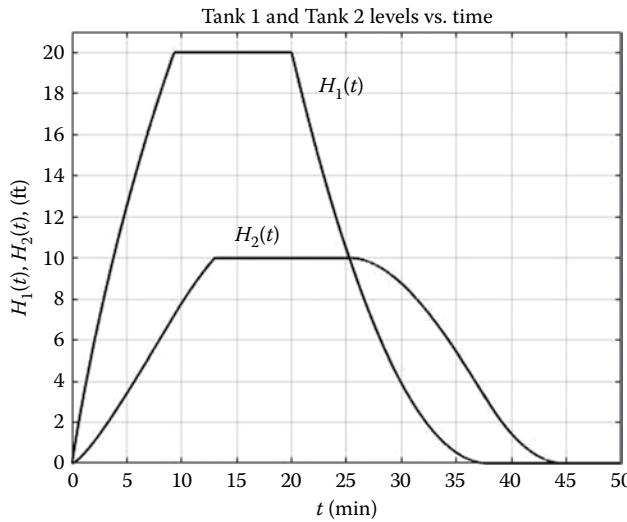


FIGURE 5.157 $H_1(t)$ and $H_2(t)$ vs. t .

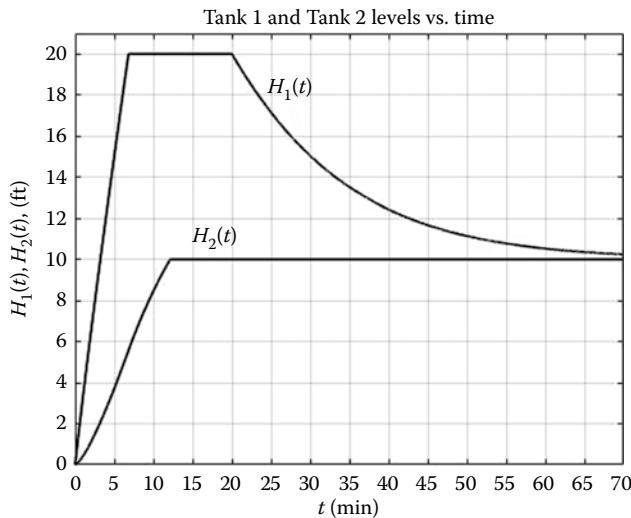


FIGURE 5.158 $H_1(t)$ and $H_2(t)$ vs. t with recycling of flow from Tank 2 to Tank 1.

$$t_0 = 0 \text{ min}, t_1 = 20 \text{ min}, A = 20 \text{ ft}^3/\text{min}$$

$$K_1 = 1, K_2 = 0$$

Case III:

$$A_1 = 5 \text{ ft}^2, L_1 = 20 \text{ ft}, c_1 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_1(0) = 10 \text{ ft},$$

$$A_2 = 3 \text{ ft}^2, L_2 = 10 \text{ ft}, c_2 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_2(0) = 5 \text{ ft}$$

$$t_0 = 0 \text{ min}, t_1 = 20 \text{ min}, A = 20 \text{ ft}^3/\text{min}$$

$$K_1 = 1, K_2 = 0$$

Case IV:

$$A_1 = 4 \text{ ft}^2, L_1 = 20 \text{ ft}, c_1 = 2.25 \text{ ft}^3/\text{min per ft}^{1/2}, H_1(0) = 0 \text{ ft},$$

$$A_2 = 4 \text{ ft}^2, L_2 = 10 \text{ ft}, c_2 = 1.5 \text{ ft}^3/\text{min per ft}^{1/2}, H_2(0) = 0 \text{ ft}$$

$$t_0 = 0 \text{ min}, t_1 = 20 \text{ min}, A = 25 \text{ ft}^3/\text{min}$$

$$K_1 = 1, K_2 = 0$$

TABLE 5.5
Summary of Simulation Results for Cases I–V

	$V_1(0)$	$V_2(0)$	$V_{1,in}$		$V_1(t_{final})$	$V_2(t_{final})$	$V_{2,out}$	$V_{1,spill}$	$V_{2,spill}$	
Case	a	b	c	a+b+c	d	e	f	g	h	d+e+f+g+h
I	0	0	160	160	4.2	22.7	133.1	0	0	160.0
II	0	0	300	300	15.6	35.0	159.5	64.6	25.3	300.0
III	50	15	400	465	30.6	27.9	184.3	175.9	46.3	465.0
IV	0	0	500	500	11.0	39.2	128.7	232.3	88.7	499.9
V	0	0	600	600	60.2	40.0	0.0	455.1	44.7	599.9

Case V:

$$A_1 = 4 \text{ ft}^2, L_1 = 20 \text{ ft}, c_1 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_1(0) = 0 \text{ ft},$$

$$A_2 = 4 \text{ ft}^2, L_2 = 10 \text{ ft}, c_2 = 2 \text{ ft}^3/\text{min per ft}^{1/2}, H_2(0) = 0 \text{ ft}$$

$$t_0 = 0 \text{ min}, t_1 = 20 \text{ min}, A = 30 \text{ ft}^3/\text{min}$$

$$K_1 = 1, K_2 = 1$$

The following dynamic variables are returned from the simulation for post-simulation analysis.

- a. $V_1(0)$, the initial volume of fluid in Tank 1 (ft^3).
- b. $V_2(0)$, the initial volume of fluid in Tank 2 (ft^3).
- c. $V_{1,in}$, the cumulative flow (ft^3) into Tank 1 from the external source from $t = 0$ to $t = t_{final}$.
- d. $V_1(t_{final})$, the final volume of fluid in Tank 1 (ft^3) at $t = t_{final}$.
- e. $V_2(t_{final})$ the final volume of fluid in Tank 2 (ft^3) at $t = t_{final}$.
- f. $V_{2,out}$, the cumulative flow (ft^3) out of Tank 2 not returned to Tank 1 from $t = 0$ to $t = t_{final}$.
- g. $V_{1,spill}$, the cumulative overflow (ft^3) from Tank 1 from $t = 0$ to $t = t_{final}$.
- h. $V_{2,spill}$, the cumulative overflow (ft^3) from Tank 1 from $t = 0$ to $t = t_{final}$.

Results are displayed in [Table 5.5](#). All volumes are in ft^3 .

The significance of the columns in the table are as follows.

a+b: $V_1(0) + V_2(0)$, total volume of fluid initially in both tanks.

c: $V_{1,in}$, total volume of fluid introduced into the first tank from the external source.

d+e: $V_1(t_{final}) + V_2(t_{final})$, total volume of fluid in both tanks at the end of the simulation.

f: $V_{2,out}$, total volume of fluid exiting the system from the second tank.

g+h: $V_{1,spill} + V_{2,spill}$, total volume of fluid spillover from both tanks.

Conservation of fluid volume requires

$$V_1(0) + V_2(0) + V_{1,in} = V_1(t_{final}) + V_2(t_{final}) + V_{2,out} + V_{1,spill} + V_{2,spill} \quad (5.186)$$

Note the agreement between the numerical values in the column labeled a+b+c and the column labeled d+e+f+g+h assuring the accuracy of the simulation results.

6 Intermediate Numerical Integration

6.1 INTRODUCTION

We continue our exposition of numerical integration introduced in [Chapter 3](#). Additional algorithms to approximate the solution of differential equation models of continuous-time systems will be examined. In previous chapters, there was no mention of how to quantify the degree of accuracy one could expect with the simple Euler and trapezoidal integrators. Truncation errors are introduced in this chapter as a way of remedying this omission.

This chapter introduces two broad classifications of numerical integrators known as one-step methods and multistep formulas and presents a case for when to use each type. Adaptive techniques for changing the integration step size when using one-step methods are discussed.

Later on, a property of system models referred to as “stiffness” is explored along with ways of dealing with it to make sure accurate and stable simulations result. Numerical stability is mentioned only briefly near the end of the chapter; however, more will be mentioned about this important property when we revisit numerical integration in [Chapter 8](#).

This chapter concludes with a case study that relies on one of the numerical integration methods introduced earlier in the chapter.

6.2 RUNGE-KUTTA (RK) (ONE-STEP METHODS)

One-step methods refer to a family of numerical integration algorithms designed to update the current state across an interval of time, called the integration step, in such a way that the state derivative function is evaluated at one or more points of the interval. In contrast, multistep methods incorporate computed state values from previous intervals in the process of updating the state.

Our discussion of one-step methods begins with an autonomous system involving a single state variable $x = x(t)$ with state derivative function $f(t, x)$.

$$\frac{dx}{dt} = f(t, x) \quad (6.1)$$

The state derivative function could be written $f(t, x, u)$ when there are external inputs present. The reason for choosing a first-order system is simple. Dynamic system models are typically higher than first order; however, the differential equations comprising an n th-order model can be recast as a set of coupled first-order differential equations for the state derivatives $\dot{x}_1(t), \dot{x}_2(t), \dots, \dot{x}_n(t)$ in terms of the state variables $x_1(t), x_2(t), \dots, x_n(t)$ and when present, inputs $u_1(t), u_2(t), \dots, u_r(t)$. The algorithms derived for numerical integration of Equation 6.1 are easily extended to the case of more than one state variable.

Suppose $x(t_i)$, the solution to Equation 6.1 at time $t = t_i$, were known and denoted x_i for short. A way of approximating $x_{i+1} = x(t_{i+1})$, the state $x(t)$ at $t = t_{i+1} = t_i + T$, is needed. The approximation is written as $x_A(i + 1)$ (see [Figure 6.1](#)).

We can proceed along a line whose slope is φ_i (see [Figure 6.1](#)) starting from the point (t_i, x_i) on the solution $x(t)$ and terminating when $t = t_{i+1}$. This leads to

$$x_A(i + 1) = x_i + T\varphi_i \quad (6.2)$$

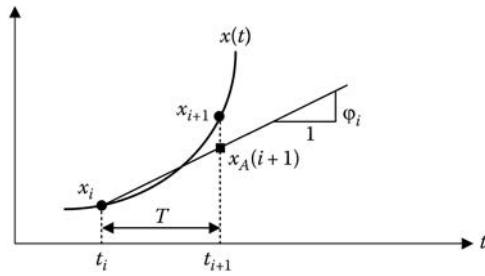


FIGURE 6.1 Graphical representation of calculation for new state $x_A(i + 1)$.

The slope φ_i is a suitably chosen approximation to the state derivative function $f(t, x)$ over the interval $t_i \leq t \leq t_{i+1}$. We shall return to this notion of a line with slope φ_i from (t_i, x_i) to $[t_{i+1}, x_A(i + 1)]$ momentarily.

6.2.1 TAYLOR SERIES METHOD

Consider the Taylor Series expansion of the function $x(t)$ shown in Figure 6.1.

Expanding the function $x(t)$ in a Taylor Series about the point t_i ,

$$x_{i+1} = x_i + \frac{d}{dt}x(t_i)T + \frac{1}{2!} \frac{d^2}{dt^2}x(t_i)T^2 + \frac{1}{3!} \frac{d^3}{dt^3}x(t_i)T^3 + \dots \quad (6.3)$$

Equation 6.3 can be expressed in terms of the state derivative function,

$$f(t, x) = \frac{d}{dt}x(t) \quad (6.4)$$

$$\Rightarrow x_{i+1} = x_i + f(t_i, x_i)T + \frac{1}{2!} \frac{d}{dt}f(t_i, x_i)T^2 + \frac{1}{3!} \frac{d^2}{dt^2}f(t_i, x_i)T^3 + \dots \quad (6.5)$$

The derivatives $(d/dt)f(t_i, x_i)$, $(d^2/dt^2)f(t_i, x_i)$, and so forth can be obtained from the chain rule. For example, the first derivative is

$$\frac{d}{dt}f(t_i, x_i) = \frac{\partial}{\partial t}f(t_i, x_i) + \frac{\partial}{\partial x}f(t_i, x_i)\frac{d}{dt}x(t_i) \quad (6.6)$$

$$= f_t(t_i, x_i) + f_x(t_i, x_i)f(t_i, x_i) \quad (6.7)$$

where

$$f_t(t_i, x_i) = \frac{\partial}{\partial t}f(t_i, x_i), \quad f_x(t_i, x_i) = \frac{\partial}{\partial x}f(t_i, x_i) \quad (6.8)$$

Substituting Equation 6.7 into Equation 6.5 yields

$$x_{i+1} = x_i + Tf(t_i, x_i) + \frac{T^2}{2}[f_t(t_i, x_i) + f_x(t_i, x_i)f(t_i, x_i)] + \dots \quad (6.9)$$

Truncating Equation 6.9 after the second term produces the explicit Euler integrator

$$x_A(i+1) = x_i + Tf(t_i, x_i) \quad (6.10)$$

which would normally be written as

$$x_A(i+1) = x_A(i) + Tf[t_i, x_A(i)] \quad (6.11)$$

since x_i is known only at the initial point $(0, x_0)$.

Truncating Equation 6.9 after the third term results in a more accurate approximation of the true value x_{i+1} , namely,

$$x_A(i+1) = x_A(i) + Tf[t_i, x_A(i)] + \frac{T^2}{2} \{f_t[t_i, x_A(i)] + f_x[t_i, x_A(i)]f[t_i, x_A(i)]\} \quad (6.12)$$

The Taylor Series method can be used to obtain difference equations such as Equations 6.11 and 6.12 for updating the discrete-time state $x_A(i)$. However, it is rarely attempted because expressions for the higher-order derivatives of $f(t, x)$ are often complex functions involving higher-order partial derivatives of $f(t, x)$. What is needed is an algorithm for computing $x_A(i+1)$ with comparable accuracy to the truncated Taylor Series without requiring partial derivatives of $f(t, x)$.

6.2.2 SECOND-ORDER RUNGE-KUTTA METHOD

Recalling our previous discussion of φ_i , the slope of the line from the point (t_i, x_i) to $[t_{i+1}, x_A(i+1)]$ in Figure 6.1, suppose we choose it to be a weighted sum of the state derivative $f(t, x)$ evaluated at several points on the interval. In particular, if φ_i is a weighted average of $f(t, x)$ at two points on the interval $t_i \leq t \leq t_{i+1}$, the result is

$$\varphi_i = a_1 k_1 + a_2 k_2 \quad (0 \leq a_1 \leq 1, 0 \leq a_2 \leq 1, a_1 + a_2 = 1) \quad (6.13)$$

where k_1 is the state derivative function $f(t, x)$ at (t_i, x_i) , that is,

$$k_1 = f(t_i, x_i) \quad (6.14)$$

and k_2 is the state derivative function $f(t, x)$ at $[t_i + pT, x_i + qTf(t_i, x_i)]$, that is,

$$k_2 = f[t_i + pT, x_i + qTf(t_i, x_i)], \quad (0 \leq p \leq 1, 0 \leq q \leq 1) \quad (6.15)$$

Lines with slopes k_1 and k_2 are shown in Figure 6.2.

From Equations 6.14 and 6.15,

$$k_2 = f[t_i + pT, x_i + qTk_1] \quad (6.16)$$

indicating that k_2 can be determined once k_1 is known. The weights a_1 and a_2 as well as the constants p and q are to be determined.

Substituting Equation 6.13 into Equation 6.2 gives

$$x_A(i+1) = x_i + T(a_1 k_1 + a_2 k_2) \quad (6.17)$$

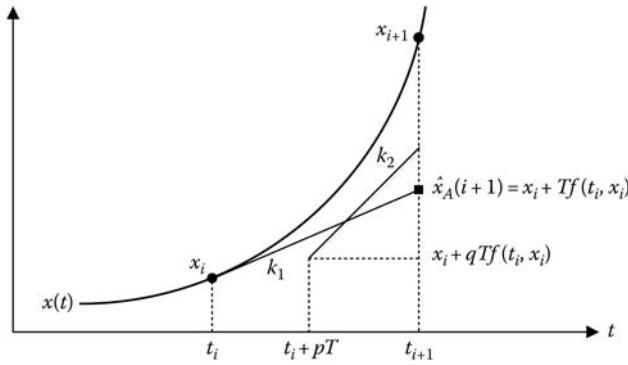


FIGURE 6.2 Representation of $\varphi_i = a_1 k_1 + a_2 k_2$ as weighted sum of $f(t, x)$ at two points.

The derivative function $f(t, x)$ can be expanded in a two-dimensional Taylor Series about the point (t_i, x_i) as follows:

$$\begin{aligned} f(t_i + \Delta t, x_i + \Delta x) &= f(t_i, x_i) + f_t(t_i, x_i)\Delta t + f_x(t_i, x_i)\Delta x \\ &\quad + \frac{1}{2}[f_{tt}(t_i, x_i)\Delta t^2 + 2f_{tx}(t_i, x_i)\Delta t\Delta x + f_{xx}(t_i, x_i)\Delta x^2] + \dots \end{aligned} \quad (6.18)$$

Letting $\Delta t = \alpha pT$, $\Delta x = qTf(t_i, x_i)$ in Equation 6.18 makes k_2 in Equation 6.16 equal to

$$\begin{aligned} k_2 &= f(t_i, x_i) + f_t(t_i, x_i)pT + f_x(t_i, x_i)qTf(t_i, x_i) \\ &\quad + \frac{1}{2}\{f_{tt}(t_i, x_i)(pT)^2 + 2f_{tx}(t_i, x_i)(pT)[qTf(t_i, x_i)] + f_{xx}(t_i, x_i)[qTf(t_i, x_i)]^2\} + \dots \end{aligned} \quad (6.19)$$

Substituting Equation 6.14 for k_1 and Equation 6.19 for k_2 into Equation 6.17 results in

$$\begin{aligned} x_A(i+1) &= x_A(i) + Ta_1 f(t_i, x_i) + Ta_2 [f(t_i, x_i) + f_t(t_i, x_i)pT + f_x(t_i, x_i)qTf(t_i, x_i)] \\ &\quad + Ta_2 \{f_{tt}(t_i, x_i)(pT)^2 + 2f_{tx}(t_i, x_i)(pT)[qTf(t_i, x_i)] + f_{xx}(t_i, x_i)[qTf(t_i, x_i)]^2\} + \dots \end{aligned} \quad (6.20)$$

Simplifying Equation 6.20 by collecting terms involving powers of T leads to

$$x_A(i+1) = x_i + (a_1 + a_2)Tf(t_i, x_i) + a_2 T^2 [pf_t(t_i, x_i) + qf_x(t_i, x_i)f(t_i, x_i)] + \dots \quad (6.21)$$

Equating the right-hand sides of Equations 6.9 and 6.21 gives

$$a_1 + a_2 = 1, \quad a_2 p = \frac{1}{2}, \quad a_2 q = \frac{1}{2} \quad (6.22)$$

The first three terms in Equation 6.3 comprise the second-order truncated Taylor Series expansion of $x(t)$ about the point t_i , that is,

$$x_2(t_i + T) = x(t_i) + \frac{d}{dt}x(t_i)T + \frac{1}{2!} \frac{d^2}{dt^2}x(t_i)T^2 \quad (6.23)$$

where the subscript “2” indicates that the Taylor Series is truncated after the term containing T^2 . Hence, by choosing the constants a_1 , a_2 , p , and q according to Equation 6.22, we can be certain that the computed state $x_A(i + 1)$ in Equation 6.21 achieves comparable accuracy as the second-order truncated Taylor Series.

There are, however, an infinite number of solutions to the three equations in four unknowns in Equation 6.22. Numerical integrators based on the use of Equation 6.17 with a_1 , a_2 , p , and q satisfying the constraints in Equation 6.22 are referred to as second-order RK or RK-2 integrators.

6.2.3 TRUNCATION ERRORS

The local truncation error ε_T is the difference between the exact solution $x(t_i + T)$ and the approximate solution $x_A(i + 1)$ obtained by the Taylor Series method or some other numerical approximation technique such as the RK-2 integrators. Hence,

$$\varepsilon_T = x(t_i + T) - x_A(i + 1) \quad (6.24)$$

For the approximation based on the second-order truncated Taylor Series method, Equation 6.24 becomes

$$\varepsilon_T = x(t_i + T) - \left[x_i + f(t_i, x_i)T + \frac{1}{2!} \frac{d}{dt} f(t_i, x_i)T^2 \right] \quad (6.25)$$

Thus, the local truncation error reduces to the sum of all the terms in the Taylor Series expansion for $x(t_i + T)$ beginning with the term containing T^3 . That is,

$$\varepsilon_T = \frac{1}{3!} \frac{d^3}{dt^3} x(t_i)T^3 + \frac{1}{4!} \frac{d^4}{dt^4} x(t_i)T^4 + \dots \quad (6.26)$$

Since the first term on the right-hand side of Equation 6.26 is generally the dominant term (magnitude-wise), the local truncation error is proportional to T^3 and is said to be of order T^3 , denoted $\varepsilon_T \sim O(T^3)$. The global truncation error E_T is the accumulation of individual truncation errors incurred in the process of numerically integrating over several intervals. It turns out that E_T is proportional to T^2 or equivalently $E_T \sim O(T^2)$.

It is important to distinguish between the order of the local truncation error and its actual value for a particular numerical integrator. We should not expect to find the numerical value of ε_T in the process of computing $x_A(i)$, $i = 0, 1, 2, \dots$. Were that possible, the exact solution $x(t_i)$, $i = 0, 1, 2, \dots$ could be computed from Equation 6.24.

We have seen that RK-2 integrators achieve comparable accuracy to the second-order truncated Taylor Series method and, as a result, are referred to as second-order accurate. The local truncation error $\varepsilon_T \sim O(T^3)$ regardless of how we solve for a_1 , a_2 , p , and q in Equation 6.22. The numerical value of ε_T will, however, be sensitive to the particular RK-2 integrator.

Knowing $\varepsilon_T \sim O(T^3)$ and $E_T \sim O(T^2)$ for RK-2 integrators makes the consequence of adjusting the integration step size predictable. For example, halving the step size reduces the local and global truncation errors by a factor of 1/8 and 1/4 respectively. For the explicit Euler integrator (RK-1), $\varepsilon_T \sim O(T^2)$ and $E_T \sim O(T)$ implying the local truncation error are reduced by 1/4 while the global truncation is approximately 1/2 as large when the step size is halved.

We now investigate two possible choices for the set of constants a_1 , a_2 , p , and q .

Solution I: $a_1 = a_2 = 1/2$ and $p = q = 1$

From Equations 6.2 and 6.13, the RK-2 integrator becomes

$$x_A(i+1) = x_i + \frac{T}{2}(k_1 + k_2) \quad (6.27)$$

Since x_i is unknown after the initial step, it must be replaced by $x_A(i)$ in Equation 6.27 to yield the difference equation for a numerical integrator. Using the definitions for k_1 and k_2 in Equations 6.14 and 6.15 and remembering that $p = q = 1$ give

$$x_A(i+1) = x_A(i) + \frac{T}{2}\{f[t_i, x_A(i)] + f[t_i + T, x_A(i) + Tf[t_i, x_A(i)]]\} \quad (6.28)$$

Denoting $x_A(i) + Tf[t_i, x_A(i)]$ by $\hat{x}_A(i+1)$ in Equation 6.28 gives

$$x_A(i+1) = x_A(i) + \frac{T}{2}\{f[t_i, x_A(i)] + f[t_i + T, \hat{x}_A(i+1)]\} \quad (6.29)$$

You should recognize $\hat{x}_A(i+1)$ as the explicit Euler estimate of x_{i+1} in Equation 6.11 (see [Figure 6.2](#)). Hence, the explicit Euler (an RK-1 integrator) establishes the second point $[t_i + T, \hat{x}_A(i+1)]$ for evaluating the derivative function, and the average derivative function or slope is then used to update the state according to Equation 6.29.

The RK-2 integrator of Equation 6.29 is the improved Euler or Heun's method introduced in Section 3.6. At that time, it was developed using a geometrical argument instead of the formal approach presented here.

The second solution for the constants a_1 , a_2 , p , and q will also look familiar.

Solution II: $a_1 = 0$, $a_2 = 1$ and $p = q = 1/2$.

From Equations 6.2 and 6.13, the RK-2 integrator is

$$x_A(i+1) = x_i + Tk_2 \quad (6.30)$$

As in the case of the improved Euler integrator, the difference equation for $x_A(i)$ results from replacing x_i by $x_A(i)$ in Equation 6.30 giving

$$x_A(i+1) = x_A(i) + Tf\left[t_i + \frac{T}{2}, x_A(i) + \frac{T}{2}f[t_i, x_A(i)]\right] \quad (6.31)$$

Introducing the notation

$$x_A\left(i + \frac{1}{2}\right) = x_A(i) + \frac{T}{2}f[t_i, x_A(i)] \quad (6.32)$$

implies the new state $x_A(i + 1)$ is calculated according to

$$x_A(i+1) = x_A(i) + Tf\left[t_i + \frac{T}{2}, x_A\left(i + \frac{1}{2}\right)\right] \quad (6.33)$$

Equation 6.33 is identical to the modified Euler integrator in Section 3.6.

In summary, the Taylor Series method (second order and higher) for approximating $x(t_i + T)$ requires the derivative function $f(t_i, x_i)$ as well as its derivatives (see Equation 6.5). RK-2 integrators produce estimates of x_{i+1} to the same accuracy as the first three terms in Equation 6.5 without requiring the total derivative (d/dt) $f(t, x)$. The price is an extra derivative function evaluation $f(t, x)$.

The following example illustrates use of the Taylor Series method and the RK-2 integrators. Results are compared with the first-order explicit Euler (RK-1) integrator and the exact solution.

EXAMPLE 6.1

The object shown in Figure 6.3 is initially at rest and then subjected to a constant force $f(t) = \bar{F}$, $t \geq 0$. The motion of the object is opposed by the damper force $f_D(t) = \alpha v(t)$. The contents of the object are leaking so that the object's mass diminishes from its initial value m_0 to a final mass m_f .

At a given time t , the mass of the object is given by

$$m(t) = \begin{cases} m_0 - ct, & 0 \leq t \leq \frac{(m_0 - m_f)}{c} \\ m_f, & t > \frac{(m_0 - m_f)}{c} \end{cases} \quad (6.34)$$

- Find an expression for the state derivative function $f(t, v)$ while the mass of the object is still decreasing.
- Find the difference equation for updating the state $v_A(i)$ using the second-order Taylor Series method.
- Find the difference equation for updating the state $v_A(i)$ using the RK-1 explicit Euler integrator.
- Find the difference equation for updating the state $v_A(i)$ using the RK-2 improved Euler integrator.
- Find the difference equation for updating the state $v_A(i)$ using the RK-2 modified Euler integrator.
- Find the exact solution for the state $v(t)$.
- Numerical values of the system parameters are $m_0 = 1$ slug, $m_f = 0.2$ slugs, $c = 0.05$ slugs/min, and $\alpha = 0.25$ lb/ft/min and the external force is $\bar{F} = 10$ lb. Tabulate and graph the results when $T = 0.5$ min.
- The differential equation model for the system is

$$m(t) \frac{dv}{dt} = \bar{F} - \alpha v \quad (6.35)$$

Solving for the derivative function,

$$\frac{dv}{dt} = f(t, v) = \frac{\bar{F} - \alpha v}{m_0 - ct}, \quad 0 \leq t \leq \frac{(m_0 - m_f)}{c} \quad (6.36)$$

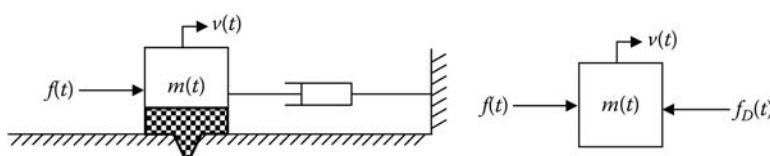


FIGURE 6.3 Moving object with decreasing mass.

b. From Equation 6.9,

$$v_{i+1} = v_i + Tf(t_i, v_i) + \frac{T^2}{2}[f_t(t_i, v_i) + fv(t_i, v_i)f(t_i, v_i)] + \dots \quad (6.37)$$

Partial differentiation of Equation 6.36 gives

$$f_t(t_i, v_i) = (\bar{F} - \alpha v_i) \frac{c}{(m_0 - ct_i)^2} \quad (6.38)$$

$$f_v(t_i, v_i) = \frac{-\alpha}{m_0 - ct_i} \quad (6.39)$$

Substituting Equations 6.36, 6.38, and 6.39 into Equation 6.37 yields

$$v_{i+1} = v_i + T \left[\frac{\bar{F} - \alpha v_i}{m_0 - ct_i} \right] + \frac{T^2}{2} \left[\frac{c(\bar{F} - \alpha v_i)}{(m_0 - ct_i)^2} - \frac{\alpha}{(m_0 - ct_i)} \frac{\bar{F} - \alpha v_i}{m_0 - ct_i} \right] + \dots \quad (6.40)$$

Truncating Equation 6.40 after the T^2 term, replacing v_i by $v_A(i)$, v_{i+1} by $v_A(i+1)$, and setting $t = iT$ lead to the difference equation

$$v_A(i+1) = v_A(i) + \left[\frac{\bar{F} - \alpha v_A(i)}{m_0 - ciT} \right] T + \frac{(c - \alpha)}{2} \left[\frac{\bar{F} - \alpha v_A(i)}{(m_0 - ciT^2)} \right] T^2 \quad (6.41)$$

c. The RK-1 explicit Euler integrator is

$$\hat{v}_A(i+1) = \hat{v}_A(i) + Tf[t_i, \hat{v}_A(i)] \quad (6.42)$$

$$= \hat{v}_A(i) + T \left[\frac{\bar{F} - \alpha \hat{v}_A(i)}{m_0 - ciT} \right] \quad (6.43)$$

d. The RK-2 improved Euler integrator, Equation 6.29, is

$$v_A(i+1) = v_A(i) + \frac{T}{2} \{ f[t_i, v_A(i)] + f[t_i + T, \hat{v}_A(i+1)] \} \quad (6.44)$$

$$= v_A(i) + \frac{T}{2} \left[\frac{\bar{F} - \alpha v_A(i)}{m_0 - ciT} + \frac{\bar{F} - \alpha \hat{v}_A(i+1)}{m_0 - c(i+1)T} \right] \quad (6.45)$$

e. The RK-2 modified Euler integrator, Equations 6.32 and 6.33, is

$$v_A\left(i + \frac{1}{2}\right) = v_A(i) + \frac{T}{2} f[t_i, v_A(i)] \quad (6.46)$$

$$= v_A(i) + \frac{T}{2} \left[\frac{\bar{F} - \alpha v_A(i)}{m_0 - ciT} \right] \quad (6.47)$$

$$v_A(i+1) = v_A(i) + Tf \left[t_{i+1/2} v_A \left(i + \frac{1}{2} \right) \right] \quad (6.48)$$

$$= v_A(i) + T \left[\frac{\bar{F} - \alpha v_A \left(i + \frac{1}{2} \right)}{m_0 - c \left(i + \frac{1}{2} T \right)} \right] \quad (6.49)$$

f. The exact solution for $v(t)$ is obtained from Equation 6.36 by integration.

$$\int_{v(0)}^v \frac{dv'}{\bar{F} - \alpha v'} = \int_0^t \frac{dt'}{m_0 - ct'} \quad (6.50)$$

$$\Rightarrow v(t) = \frac{\bar{F}}{\alpha} - \left[\frac{\bar{F}}{\alpha} - v(0) \right] \left[1 - \frac{ct}{m_0} \right]^{\alpha/c}, \quad 0 \leq t \leq \frac{(m_0 - m_f)}{c} \quad (6.51)$$

g. For the numerical values given, results from the Taylor Series method, the three numerical integrators, and the exact solution are tabulated in [Table 6.1](#) at 1 min intervals after the first two steps.

[Figure 6.4](#) contains a graph of the four numerical integrators and the exact solution. Both the table and figure confirm the improved accuracy possible with the use of the Taylor Series method and RK-2 integration compared to the explicit Euler (RK-1) integrator.

TABLE 6.1

Taylor Series Method, RK-1 (Explicit Euler), RK-2 (Improved Euler), RK-2 (Modified Euler) with $T = 0.5$ min, and Exact Solution

i	$t_i = iT$	Taylor Series Method $v_A(i)$	RK-1 Explicit Euler $\hat{v}_A(i)$	RK-2 Improved Euler $v_A(i)$	RK-2 Modified Euler $v_A(i)$	Exact Solution
0	0	0	0	0	0	0
1	0.5	4.75	5.0	4.7436	4.7468	4.7562
2	1	9.0375	9.4872	9.0257	9.0317	9.0488
4	2	16.3617	17.0828	16.3421	16.3520	16.3804
6	3	22.2287	23.0849	22.2045	22.2168	22.2518
8	4	26.8677	27.7584	26.8415	26.8548	26.8928
10	5	30.4826	31.3371	30.4562	30.4696	30.5078
12	6	33.2532	34.0256	33.2280	33.2408	33.2772
14	7	35.3370	36.0013	35.3139	35.3256	35.3588
16	8	36.8704	37.4162	36.8501	36.8604	36.8896
18	9	37.9706	38.3992	37.9534	37.9622	37.9869
20	10	38.7368	39.0575	38.7226	38.7298	38.7500
22	11	39.2515	39.4792	39.2403	39.2460	39.2619
24	12	39.5826	39.7345	39.5741	39.5785	39.5904
26	13	39.7843	39.8783	39.7782	39.7814	39.7899
28	14	39.8991	39.9519	39.8948	39.8970	39.9028
30	15	39.9586	39.9847	39.9559	39.9573	39.9609

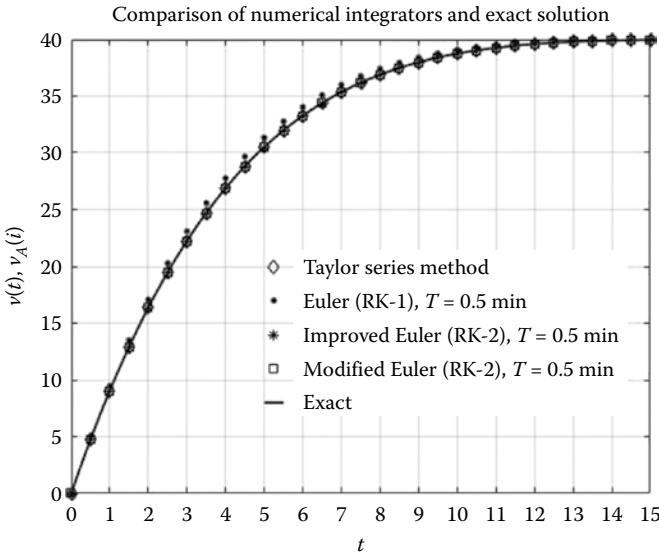


FIGURE 6.4 Comparison of numerical integrators and exact solution for Example 6.1.

Knowing the exact solution, we can check the results obtained from the Taylor Series method. For the numerical values given, the exact solution in Equation 6.51 becomes

$$v(t) = 40 - 40(1 - 0.05t)^5, \quad 0 \leq t \leq 16 \quad (6.52)$$

The second-order truncated Taylor Series $v_2(t)$ about the point $t = 0$ is

$$v_2(T) = v(0) + \frac{d}{dt}v(0)T + \frac{1}{2} \frac{d^2}{dt^2}v(0)T^2 \quad (6.53)$$

Setting $v(0)$ to zero, differentiating Equation 6.52 to find the first two derivatives and substituting the results into Equation 6.53 give

$$\begin{aligned} v_2(T) &= 10T + \frac{1}{2}(-2)T^2 \\ \Rightarrow v_2(0.5) &= 10(0.5) - (0.5)^2 = 4.75 \end{aligned} \quad (6.54)$$

which agrees with the value in [Table 6.1](#).

6.2.4 HIGH-ORDER RUNGE-KUTTA METHODS

Higher-order RK formulas are derived in the same manner as the RK-2 integrators. For RK-3 integration, the formula for updating the state $x_A(i)$, is

$$x_A(i+1) = x_A(i) + T(a_1k_1 + a_2k_2 + a_3k_3) \quad (6.55)$$

where k_1 , k_2 , and k_3 are derivative function evaluations at specific points. There are now three constants p , q , and r , which determine the points at which the derivatives are to be evaluated. Matching coefficients of powers of T in the expression for $x_A(i+1)$ using Equation 6.55 with the truncated

Taylor Series for $x(t)$ through the T^3 term generates four equations in the six unknowns a_1 , a_2 , and a_3 and p , q , and r .

One particular solution leads to the frequently used RK-3 integration formula

$$x_A(i+1) = x_A(i) + \frac{T}{6}(k_1 + 4k_2 + k_3) \quad (6.56)$$

where

$$k_1 = f[t_i, x_A(i)] \quad (6.57)$$

$$k_2 = f\left[t_i + \frac{1}{2}T, x_A(i) + \frac{1}{2}k_1 T\right] \quad (6.58)$$

$$k_3 = f[t_i + T, x_A(i) - k_1 T + 2k_2 T] \quad (6.59)$$

The local truncation error of an RK-3 integrator $\varepsilon_T \sim O(T^4)$ and the global truncation error $E_T \sim O(T^3)$.

Fourth-order RK formulas are the most common of all the RK numerical integrators for reasons we shall discuss shortly. The derivation is patterned after the approach used for the lower-order RK methods. Flexibility in the choice of several parameters results in a family of RK-4 integrators. A popular RK-4 integrator is illustrated in [Figure 6.5](#).

The derivative function evaluations are computed according to

$$k_1 = f[t_i, x_A(i)], \quad x_A\left(i + \frac{1}{2}\right) = x_A(i) + \frac{T}{2}k_1 \quad (6.60)$$

$$k_2 = f\left[t_{i+1/2}, x_A\left(i + \frac{1}{2}\right)\right], \quad \hat{x}_A\left(i + \frac{1}{2}\right) = x_A(i) + \frac{T}{2}k_2 \quad (6.61)$$

$$k_3 = f\left[t_{i+1/2}, \hat{x}_A\left(i + \frac{1}{2}\right)\right], \quad \hat{x}_A(i+1) = x_A(i) + Tk_3 \quad (6.62)$$

$$k_4 = f[t_{i+1}, \hat{x}_A(i+1)] \quad (6.63)$$

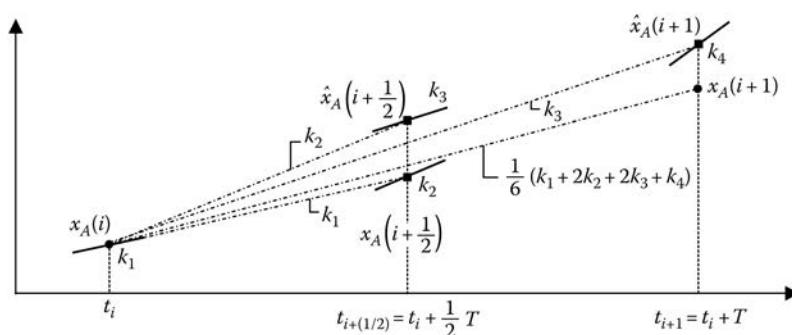


FIGURE 6.5 Illustration of an RK-4 integrator.

and the updated state $x_A(i + 1)$ is obtained from

$$x_A(i + 1) = x_A(i) + \frac{T}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (6.64)$$

Note that of the four required derivative evaluations, one is at the beginning of the interval, two occur at the midpoint, and the last one takes place at the end of the interval. The algorithm is straightforward to program because of the sequential nature in the calculations of k_1 , k_2 , k_3 , and k_4 .

RK-1 through RK-4 (and higher) integrators are incorporated in simulation and numerical analysis software packages. MATLAB® and Simulink® offer a choice of RK-1 through RK-5 integrators.

6.2.5 LINEAR SYSTEMS: APPROXIMATE SOLUTIONS USING RK INTEGRATION

The special case of linear system models is worth looking at in some detail. Suppose the derivative function in Equation 6.1 is linear in x , that is,

$$\frac{dx}{dt} = f(t, x) = ax \quad (6.65)$$

Applying RK-1, RK-2, RK-3, and RK-4 integrators to the linear system in Equation 6.65 produces the following difference equations for updating the state $x_A(i)$:

$$\text{RK-1: } x_A(i + 1) = (1 + aT)x_A(i) \quad (6.66)$$

$$\text{RK-2: } x_A(i + 1) = \left[1 + aT + \frac{1}{2!}(aT)^2 \right] x_A(i) \quad (6.67)$$

$$\text{RK-3: } x_A(i + 1) = \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 \right] x_A(i) \quad (6.68)$$

$$\text{RK-4: } x_A(i + 1) = \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 + \frac{1}{4!}(aT)^4 \right] x_A(i) \quad (6.69)$$

The general solutions to Equations 6.66 through 6.69 are easily obtained by recursion. The results are

$$\text{RK-1: } x_A(i) = (1 + aT)^i x(0) \quad (6.70)$$

$$\text{RK-2: } x_A(i) = \left[1 + aT + \frac{1}{2!}(aT)^2 \right]^i x(0) \quad (6.71)$$

$$\text{RK-3: } x_A(i) = \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 \right]^i x(0) \quad (6.72)$$

$$\text{RK-4: } x_A(i) = \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 + \frac{1}{4!}(aT)^4 \right]^i x(0) \quad (6.73)$$

where $x(0)$ is the initial condition. In general, an RK-m integrator applied to the linear system model, Equation 6.65, results in

$$x_A(i) = \left[\sum_{k=0}^m \frac{(aT)^k}{k!} \right]^i x(0) \quad (6.74)$$

$$= \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 + \cdots + \frac{1}{m!}(aT)^m \right]^i x(0) \quad (6.75)$$

In the case of more than a single state variable, that is, $\dot{x} = Ax$ a similar result applies for RK-*m* integrators.

$$\underline{x}_A(i) = \left[\sum_{k=0}^m \frac{(TA)^k}{k!} \right]^i \underline{x}(0) \quad (6.76)$$

$$= \left[I + TA + \frac{1}{2!}(TA)^2 + \frac{1}{3!}(TA)^3 + \cdots + \frac{1}{m!}(TA)^m \right]^i \underline{x}(0) \quad (6.77)$$

Equation 6.76 for the explicit Euler integrator ($m = 1$) as well as the improved and modified Euler integrators ($m = 2$) was first introduced in Section 3.6.

The discrete-time signal $x_A(i)$, $i = 0, 1, 2, 3, \dots$ is intended to approximate the continuous-time state $x(t)|_{t=iT}$, $i = 0, 1, 2, 3, \dots$. The solution $x(t)$ to Equation 6.65 is

$$x(t) = x(0)e^{at}, \quad t \geq 0 \quad (6.78)$$

$$\Rightarrow x(iT) = x(0)e^{aiT} = x(0)(e^{aT})^i \quad (6.79)$$

Expanding e^{aT} in a Taylor Series about zero, Equation 6.79 becomes

$$x(iT) = \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 + \cdots + \frac{1}{m!}(aT)^m + \cdots \right]^i x(0) \quad (6.80)$$

From Equations 6.74 and 6.80 with $i = 1$, the $m + 1$ terms in the approximate value $x_A(1)$ are identical to the first $m + 1$ terms of the infinite series expression for $x(T)$.

After one step, the local truncation error of an RK integrator is

$$\varepsilon_T = x(T) - x_A(1) \quad (6.81)$$

For an RK-*m* integrator,

$$\varepsilon_T = e^{aT} x(0) - \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 + \cdots + \frac{1}{m!}(aT)^m \right] x(0) \quad (6.82)$$

Replacing e^{aT} in Equation 6.82 by its Taylor Series expansion leads to

$$\varepsilon_T = \left[\frac{1}{(m+1)!} (aT)^{m+1} + \frac{1}{(m+2)!} (aT)^{m+2} \dots \right] x(0) \quad (6.83)$$

and, therefore, $\varepsilon_T \sim O(T^{m+1})$ as expected. All RK- m integrators are said to be of m th order, not to be confused with their local truncation error, which is of order $m + 1$, that is, $\varepsilon_T \sim O(T^{m+1})$. The m th order reference stems from the high-order term in the truncated Taylor Series. For an RK- m integrator, the global truncation error $E_T \sim O(T^m)$.

RK-1 through RK-4 integrators require one to four derivative function evaluations per step. RK integrators of order higher than four are not as efficient. For example, an RK-5 integrator requires six derivative function evaluations per step for comparable agreement with the fifth-order Taylor Series expansion of the solution. A penalty of one additional derivative function evaluation per step is the price incurred in moving from an RK-4 integrator with $\varepsilon_T \sim O(T^5)$ to an RK-5 integrator with $\varepsilon_T \sim O(T^6)$. The computational effort during each integration step results primarily from evaluating the derivative function. Hence, the penalty is nontrivial.

Worse yet, RK-6 integrators require eight derivative function evaluations to achieve a local truncation error $\varepsilon_T \sim O(T^7)$. RK-4 methods are popular because they are the highest order one-step integrators that do not require more derivative function evaluations than their order.

6.2.6 CONTINUOUS-TIME MODELS WITH POLYNOMIAL SOLUTIONS

The Taylor Series method for finding $x_A(i+1)$ starting from the point (t_i, x_i) on the solution $x(t)$ is

$$x_A(i+1) = x_i + \frac{d}{dt} x(t_i)T + \frac{1}{2!} \frac{d^2}{dt^2} x(t_i)T^2 + \dots + \frac{1}{m!} \frac{d^m}{dt^m} x(t_i)T^m \quad (6.84)$$

where the total derivatives $(d^2/dt^2)x(t_i)$, $(d^3/dt^3)x(t_i)$, ..., $(d^m/dt^m)x(t_i)$ are computed from partial derivatives of the derivative function $f(t_i, x_i)$.

Suppose the exact solution is the m th-order polynomial

$$x(t) = a_0 + a_1 t + a_2 t^2 + \dots + a_m t^m \quad (6.85)$$

The exact solution at $t = t_{i+1}$ is

$$x(t_{i+1}) = a_0 + a_1 t_{i+1} + a_2 t_{i+1}^2 + \dots + a_m t_{i+1}^m \quad (6.86)$$

With $x_A(0)$ set equal to $x(0)$, Equations 6.84 and 6.86 produce identical results at the discrete points 0, T , $2T$, In other words,

$$x_A(i+1) = x(t_{i+1}), \quad i = 0, 1, 2, \dots \quad (6.87)$$

Proof for the case when $m = 2$ follows. Starting with

$$x_A(i+1) = x_i + \frac{d}{dt} x(t_i)T + \frac{1}{2!} \frac{d^2}{dt^2} x(t_i)T^2 \quad (6.88)$$

The two derivatives in Equation 6.88 are obtained from the exact solution for $x(t)$ in Equation 6.85 with $m = 2$. Substituting them into Equation 6.88 and simplifying give

$$x_A(i+1) = x_i + (a_1 + 2a_2 t_i)T + \frac{1}{2}(2a_2)T^2 \quad (6.89)$$

$$= (a_0 + a_1 t_i + a_2 t_i^2) + a_1 T + 2a_2 t_i T + a_2 T^2 \quad (6.90)$$

$$= a_0 + a_1(t_i + T) + a_2(t_i + T)^2 \quad (6.91)$$

$$= x(t_{i+1}) \quad (6.92)$$

The proof is similar for higher-order polynomial solutions.

In Example 6.1, the exact solution $v(t)$ in Equation 6.52 is a fifth-order polynomial. Hence, the Taylor Series method using the fifth-order truncated Taylor Series would agree with the exact solution at $0, T, 2T, \dots$. However, in Example 6.1, a second-order Taylor Series was used to generate the discrete-time values $v_A(1), v_A(2), v_A(4), v_A(6), \dots, v_A(30)$ shown in [Table 6.1](#). This explains the discrepancy between the discrete-time values and the exact solution $v(t_1), v(t_2), v(t_4), v(t_6), \dots, v(t_{30})$ shown in the last column of the table.

In general, when $x(t)$ is an m th-order polynomial, unlike the m th-order Taylor Series method, RK- m integrators will not generate the true solution values $x(t_1), x(t_2), x(t_3), \dots$. Different RK- m integrators will produce different discrete-time solutions; however, they achieve comparable accuracy with the Taylor Series method in the sense that the local truncation errors are the same order of magnitude. A similar result holds for RK integrators and the truncated Taylor Series method when both are the same order and less than m . In that case, the Taylor Series method will no longer be exact. The following example illustrates this point.

EXAMPLE 6.2

In Example 6.1, if we change the value of α from 0.25 to 0.1, the exact solution to Equation 6.35 becomes

$$v(t) = 100 - \frac{1}{4}(20 - t)^2 \quad (6.93)$$

Approximate the solution for $v(t)$ using the second-order Taylor Series, RK-1 integration, and both RK-2 integrators with a step size of $T = 0.5$. Compare results with the exact solution.

The state derivative function, Equation 6.36, is given by

$$f(t_i, v_i) = \frac{\bar{F} - \alpha v_i}{m_0 - ct_i} = \frac{10 - 0.1v_i}{1 - 0.05t_i} = 2 \left(\frac{100 - v_i}{20 - t_i} \right) \quad (6.94)$$

The first partials in Equations 6.38 and 6.39 become

$$f_t(t_i, v_i) = 2 \left[\frac{100 - v_i}{(20 - t_i)^2} \right] \quad (6.95)$$

$$f_v(t_i, v_i) = \frac{-2}{(20 - t_i)} \quad (6.96)$$

and the Taylor Series method for calculating the approximation to $v(t + T)$ is

$$v_A(i+1) = v_A(i) + Tf[t_i, v_A(i)] + \frac{T^2}{2} \{f_t[t_i, v_A(i)] + f_v[t_i, v_A(i)]f_t[t_i, v_A(i)]\} \quad (6.97)$$

TABLE 6.2
Comparison of Taylor Series Method, RK-1, RK-2, and Exact Solution

<i>i</i>	<i>t_i</i>	Taylor Series <i>v_A(i)</i>	RK-1 Explicit <i>v_A(i)</i>	RK-2 Improved <i>v_A(i)</i>	RK-2 Modified <i>v_A(i)</i>	Exact <i>v(t_i)</i>
0	0	0	0	0	0	0
1	0.5	4.9375	5.0000	4.9359	4.9367	4.9375
2	1.0	9.7500	9.8718	9.7468	9.7484	9.7500
4	2.0	19.0000	19.2308	18.9938	18.9970	19.0000
6	3.0	27.7500	28.0769	27.7410	27.7455	27.7500
8	4.0	36.0000	36.4103	35.9883	35.9942	36.0000
10	5.0	43.7500	44.2308	43.7357	43.7430	43.7500

From Equations 6.94 through 6.97 with $i = 0$, $t_i = t_0 = 0$ and $v_A(i) = v_A(0) = v(0) = 0$,

$$v_A(1) = 10T - \frac{1}{4}T^2 \quad (6.98)$$

For a step size of $T = 0.5$, $v_A(1) = 4.9375$. The exact solution $v(T)|_{T=0.5}$ is computed from

$$v(t)|_{t=T=0.5} = 100 - \frac{1}{4}(20-t)^2 \Big|_{t=0.5} = 4.9375$$

which agrees with the result from the Taylor Series method.

Results for the Taylor Series method, RK-1, both RK-2 integrators, and the exact solution are tabulated in [Table 6.2](#).

6.2.7 HIGHER-ORDER SYSTEMS

The application of RK numerical integration to higher-order systems is straightforward. The differential equations of an n th-order system model are expressed as a system of first-order differential equations as shown in Equations 6.99 through 6.101.

$$\frac{dx_1}{dt} = f_1(t, x_1, x_2, \dots, x_n) \quad (6.99)$$

$$\frac{dx_2}{dt} = f_2(t, x_1, x_2, \dots, x_n) \quad (6.100)$$

⋮

$$\frac{dx_n}{dt} = f_n(t, x_1, x_2, \dots, x_n) \quad (6.101)$$

Updating the current discrete-time state vector $[x_{1,A}(i), x_{2,A}(i), \dots, x_{n,A}(i)]$ to the new vector $[x_{1,A}(i+1), x_{2,A}(i+1), \dots, x_{n,A}(i+1)]$ with RK- m integration consists of determining, in the proper sequence, the derivatives $k_{j,p}, j = 1, 2, \dots, m$ and $p = 1, 2, 3, \dots, n$. By the proper sequence, we mean $k_{1,1}, k_{1,2}, \dots, k_{1,n}$, followed by $k_{2,1}, k_{2,2}, \dots, k_{2,n}$ up through by $k_{m,1}, k_{m,2}, \dots, k_{m,n}$.

To illustrate, suppose we are dealing with a third-order ($n = 3$) system and choose to implement a fourth-order ($m = 4$) RK-4 integrator to update the discrete-time state. The three derivative functions are each calculated four times in the following order:

$$\left. \begin{aligned} k_{1,1} &= f_1[t_i, x_{1,A}(i), x_{2,A}(i), x_{3,A}(i)] \\ k_{1,2} &= f_2[t_i, x_{1,A}(i), x_{2,A}(i), x_{3,A}(i)] \\ k_{1,3} &= f_3[t_i, x_{1,A}(i), x_{2,A}(i), x_{3,A}(i)] \end{aligned} \right\} \quad (6.102)$$

$$\left. \begin{aligned} k_{2,1} &= f_1[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{1,1}, x_{2,A}(i) + 0.5Tk_{1,2}, x_{3,A}(i) + 0.5Tk_{1,3}] \\ k_{2,2} &= f_2[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{1,1}, x_{2,A}(i) + 0.5Tk_{1,2}, x_{3,A}(i) + 0.5Tk_{1,3}] \\ k_{2,3} &= f_3[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{1,1}, x_{2,A}(i) + 0.5Tk_{1,2}, x_{3,A}(i) + 0.5Tk_{1,3}] \end{aligned} \right\} \quad (6.103)$$

$$\left. \begin{aligned} k_{3,1} &= f_1[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{2,1}, x_{2,A}(i) + 0.5Tk_{2,2}, x_{3,A}(i) + 0.5Tk_{2,3}] \\ k_{3,2} &= f_2[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{2,1}, x_{2,A}(i) + 0.5Tk_{2,2}, x_{3,A}(i) + 0.5Tk_{2,3}] \\ k_{3,3} &= f_3[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{2,1}, x_{2,A}(i) + 0.5Tk_{2,2}, x_{3,A}(i) + 0.5Tk_{2,3}] \end{aligned} \right\} \quad (6.104)$$

$$\left. \begin{aligned} k_{4,1} &= f_1[t_i + T, x_{1,A}(i) + Tk_{3,1}, x_{2,A}(i) + Tk_{3,2}, x_{3,A}(i) + Tk_{3,3}] \\ k_{4,2} &= f_2[t_i + T, x_{1,A}(i) + Tk_{3,1}, x_{2,A}(i) + Tk_{3,2}, x_{3,A}(i) + Tk_{3,3}] \\ k_{4,3} &= f_3[t_i + T, x_{1,A}(i) + Tk_{3,1}, x_{2,A}(i) + Tk_{3,2}, x_{3,A}(i) + Tk_{3,3}] \end{aligned} \right\} \quad (6.105)$$

The components of the state are updated according to

$$x_{1,A}(i+1) = x_{1,A}(i) + \frac{T}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}) \quad (6.106)$$

$$x_{2,A}(i+1) = x_{2,A}(i) + \frac{T}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}) \quad (6.107)$$

$$x_{3,A}(i+1) = x_{3,A}(i) + \frac{T}{6}(k_{1,3} + 2k_{2,3} + 2k_{3,3} + k_{4,3}) \quad (6.108)$$

An example of a second-order system model using RK-4 integration is now presented. The standard form of a linear second-order system is

$$\frac{d^2x}{dt^2} = 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = K\omega_n^2 u \quad (6.109)$$

Letting $x_1 = x$ and $x_2 = dx/dt$ leads to the state equation model

$$\frac{dx_1}{dt} = f_1(t, x_1, x_2, u) = x_2 \quad (6.110)$$

$$\frac{dx_2}{dt} = f_2(t, x_1, x_2, u) = -\omega_n^2 x_1 - 2\zeta\omega_n x_2 + K\omega_n^2 u \quad (6.111)$$

where the last argument u of $f_1(t, x_1, x_2, u)$ and $f_2(t, x_1, x_2, u)$ refers to the system input.

Expressions for the derivatives k_1 , k_2 , k_3 , and k_4 associated with states x_1 and x_2 are

$$k_{1,1} = f_1[t_i, x_{1,A}(i), x_{2,A}(i), u(t_i)] \quad (6.112)$$

$$= x_{2,A}(i) \quad (6.113)$$

$$k_{1,2} = f_2[t_i, x_{1,A}(i), x_{2,A}(i), u(t_i)] \quad (6.114)$$

$$= -\omega_n^2 x_{1,A}(i) - 2\zeta\omega_n x_{2,A}(i) + K\omega_n^2 u(t_i) \quad (6.115)$$

$$k_{2,1} = f_1[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{1,1}, x_{2,A}(i) + 0.5Tk_{1,2}, u(t_i + 0.5T)] \quad (6.116)$$

$$= x_{2,A}(i) + 0.5Tk_{1,2} \quad (6.117)$$

$$k_{2,2} = f_2[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{1,1}, x_{2,A}(i) + 0.5Tk_{1,2}, u(t_i + 0.5T)] \quad (6.118)$$

$$= -\omega_n^2 [x_{1,A}(i) + 0.5Tk_{1,1}] - 2\zeta\omega_n [x_{2,A}(i) + 0.5Tk_{1,2}] + K\omega_n^2 u(t_i + 0.5T) \quad (6.119)$$

$$k_{3,1} = f_1[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{2,1}, x_{2,A}(i) + 0.5Tk_{2,2}, u(t_i + 0.5T)] \quad (6.120)$$

$$= x_{2,A}(i) + 0.5Tk_{2,2} \quad (6.121)$$

$$k_{3,2} = f_2[t_i + 0.5T, x_{1,A}(i) + 0.5Tk_{2,1}, x_{2,A}(i) + 0.5Tk_{2,2}, u(t_i + 0.5T)] \quad (6.122)$$

$$= -\omega_n^2 [x_{1,A}(i) + 0.5Tk_{2,1}] - 2\zeta\omega_n [x_{2,A}(i) + 0.5Tk_{2,2}] + K\omega_n^2 u(t_i + 0.5T) \quad (6.123)$$

$$k_{4,1} = f_1[t_i + T, x_{1,A}(i) + Tk_{3,1}, x_{2,A}(i) + Tk_{3,2}, u(t_i + T)] \quad (6.124)$$

$$= x_{2,A}(i) + Tk_{3,2} \quad (6.125)$$

$$k_{4,2} = f_2[t_i + T, x_{1,A}(i) + Tk_{3,1}, x_{2,A}(i) + Tk_{3,2}, u(t_i + T)] \quad (6.126)$$

$$= -\omega_n^2 [x_{1,A}(i) + Tk_{3,1}] - 2\zeta\omega_n [x_{2,A}(i) + Tk_{3,2}] + K\omega_n^2 u(t_i + T) \quad (6.127)$$

The updated state $[x_{1,A}(i), x_{2,A}(i)]$ is obtained from

$$x_{1,A}(i+1) = x_{1,A}(i) + \frac{T}{6} (k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}) \quad (6.128)$$

$$x_{2,A}(i+1) = x_{2,A}(i) + \frac{T}{6} (k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}) \quad (6.129)$$

An example illustrating the application of Equations 6.110 through 6.129 follows.

EXAMPLE 6.3

A simplified model to predict the levels of a drug in an individual is accomplished using compartmental analysis similar to the iodine model in Section 4.3. After oral ingestion, a drug enters the gastrointestinal tract (compartment 1) and is then distributed into the bloodstream (compartment 2) where it is metabolized and eliminated. State equations describing the drug dynamics in each compartment are

$$\text{Gastrointestinal tract: } \frac{dm_1}{dt} = -c_1 m_1 + u \quad (6.130)$$

$$\text{Bloodstream: } \frac{dm_2}{dt} = -c_1 m_1 - c_2 m_2 \quad (6.131)$$

where

m_1 and m_2 are the amounts of drug in each compartment (mg)

u is the ingestion rate of the drug (mg/min)

c_1 and c_2 are drug distribution and elimination constants of the individual (min^{-1})

The output y is the amount of drug in the bloodstream, that is, m_2 .

- Convert the state equations into a single second-order differential equation relating the output y and input u . Find ζ , ω_n , and K for the second-order system.
- Define $x_1 = y = m_2$ and $x_2 = dy/dt = dm_2/dt$ and simulate the response using classic RK-4 integration with a step size $T = 1$ min for the following conditions:

$$m_1(0) = 0 \text{ mg}, m_2(0) = 0 \text{ mg}, c_1 = 0.06 \text{ min}^{-1}, \text{ and } c_2 = 0.015 \text{ min}^{-1}$$

Assume the drug ingestion rate is given by

$$u(t) = M e^{-t/\tau}, \quad t \geq 0 \quad (M = 5 \text{ mg/min}, \tau = 4 \text{ min}) \quad (6.132)$$

- Find the exact solution for $x_1(t)$.
- Plot the simulated response $x_{VA}(t)$ and the exact solution $x_1(t)$ on the same graph.
- Elimination of m_1 from Equations 6.130 and 6.131 is easily accomplished by Laplace transforming the equations and algebraically solving for $M_2(s)$, which is also $Y(s)$.

$$(s + c_1)M_1(s) = U(s) \quad (6.133)$$

$$(s + c_2)M_2(s) = c_1 M_1(s) \quad (6.134)$$

$$Y(s) = M_2(s) = \frac{c_1}{s + c_2} M_1(s) \quad (6.135)$$

$$= \frac{c_1}{s + c_2} \left[\frac{1}{s + c_1} U(s) \right] \quad (6.136)$$

Inverse Laplace transformation of $Y(s)$ leads to the differential equation

$$\frac{d^2y}{dt^2} + (c_1 + c_2) \frac{dy}{dt} + c_1 c_2 y = c_1 u \quad (6.137)$$

Comparing Equation 6.137 with the standard form of Equation 6.109 yields

$$\omega_n = (c_1 c_2)^{1/2}, \quad \zeta = \frac{c_1 + c_2}{2(c_1 c_2)^{1/2}}, \quad K = \frac{1}{c_2} \quad (6.138)$$

Solving for the second-order system parameters,

$$\omega_n = (c_1 c_2)^{1/2} = [(0.06)(0.015)]^{1/2} = 0.03 \text{ rad/min}$$

$$\zeta = \frac{c_1 + c_2}{2(c_1 c_2)^{1/2}} = \frac{0.06 + 0.015}{2(0.03)} = 1.25$$

$$K = \frac{1}{c_2} = \frac{1}{0.015} = 66.67 \text{ min}^{-1}$$

- b. The RK-4 calculations follow the procedure outlined in Equations 6.112 through 6.129. The integration step begins with the k_1 derivative evaluation for each state, that is,

$$\begin{aligned} k_{1,1} &= x_{2,A}(0) = x_2(0) = 0 \\ k_{1,2} &= -\omega_n^2 x_{1,A}(0) - 2\zeta \omega_n x_{2,A}(0) + K \omega_n^2 u(0) \\ &= -\omega_n^2 x_1(0) - 2\zeta \omega_n x_2(0) + K \omega_n^2 M \\ &= -0.0009(0) - 2(1.25)(0.03)(0) + \left(\frac{1}{0.015}\right)(0.0009)(5) \\ &= 0.3 \end{aligned}$$

followed by the k_2 derivative evaluation for each state, that is,

$$\begin{aligned} k_{2,1} &= x_{2,A}(0) + \frac{1}{2} k_{1,2} T \\ &= 0 + \frac{1}{2} (0.3)(1) \\ &= 0.15 \end{aligned}$$

$$\begin{aligned} k_{2,2} &= -\omega_n^2 \left[x_{1,A}(i) + \frac{1}{2} k_{1,1} T \right] - 2\zeta \omega_n \left[x_{2,A}(i) + \frac{1}{2} k_{1,2} T \right] + K \omega_n^2 u \left(t_i + \frac{1}{2} T \right) \\ &= -0.0009 \left[0 + \frac{1}{2} (0)(1) \right] - 2(1.25)(0.03) \left[0 + \frac{1}{2} (0.3)(1) \right] \\ &\quad + \left(\frac{1}{0.015}\right)(0.0009)5 e^{-0.5/4} \\ &= 0.2535 \end{aligned}$$

The remaining derivative evaluations are obtained in a similar fashion.

$$k_{3,1} = 0.1267, k_{3,2} = 0.2552, k_{4,1} = 0.2552, k_{4,2} = 0.2144$$

M-file “Ch6_Ex6_3.m” recursively solves the RK-4 difference equations for the discrete-time states $x_{1,A}(i)$ and $x_{2,A}(i)$. Table 6.3 contains selected values of $x_{1,A}(i)$.

- c. The exact solution for $x_1(t) = y(t)$ can be determined by substituting the Laplace transform of $u(t)$ into Equation 6.136,

$$X_1(s) = \frac{c_1}{s + c_2} \left[\frac{1}{s + c_1} \left(\frac{M}{s + (1/\tau)} \right) \right] \quad (6.139)$$

TABLE 6.3
Approximate (RK-4, $T = 1$ min) and Exact Solutions

i	t_i	$x_{1,A}(i)$	$x_1(t_i)$
0	0	0.0	0.0
1	1	0.13477907	0.13477243
2	2	0.48560007	0.48558842
3	3	0.98658006	0.98656465
4	4	1.58751933	1.58750114
5	5	2.25036682	2.25034661
25	25	11.68143041	11.68141034
50	50	11.65359364	11.65357994
100	100	6.24296647	6.24295986
150	150	2.98572189	2.98571876
200	200	1.41218500	1.41218352
250	250	0.66716006	0.66715936
300	300	0.31514864	0.31514831

Inverse Laplace transforming Equation 6.139 gives the exact solution (Figure 6.6),

$$x_1(t) = \frac{MC_1\tau}{(c_1 - c_2)(1 - c_1\tau)(1 - c_2\tau)} \left[(1 - c_1\tau)e^{-c_2 t} - (1 - c_2\tau)e^{-c_1 t} + (c_1 - c_2)\tau e^{-t/\tau} \right] \quad (6.140)$$

- d. Table 6.3 contains values of the discrete-time response $x_{1,A}(i)$ and the exact solution $x_1(t)$ at different times. The approximate and exact solutions agree to four places after the decimal point.

Figure 6.6 contains plots of the discrete-time states $x_{1,A}(i)$ and $x_{2,A}(i)$ and the exact solution $x_1(t)$. Every fourth point of the discrete-time signals is plotted for the sake of clarity.

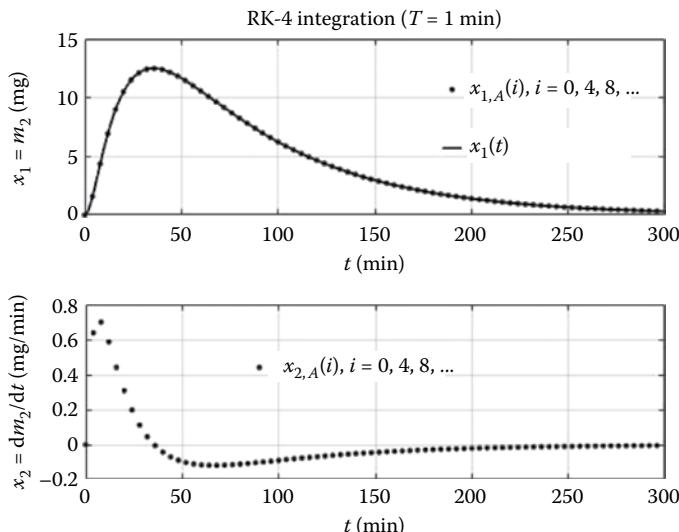


FIGURE 6.6 Discrete-time signals $x_{1,A}(i)$ and $x_{2,A}(i)$ and continuous-time signal $x_1(t)$.

Before we proceed further, it would be useful to know the total amount of drug ingested by the individual. Integrating the rate of drug ingestion over time,

$$M_T = \int_0^{\infty} u(t) dt = \int_0^{\infty} M e^{-t/\tau} dt = M\tau \quad (6.141)$$

For a continuous-time integrator, the derivative function $f(t, x)$ is equal to the input $u(t)$ to the integrator. Hence, M_T in Equation 6.141 can be found for an arbitrary input function $u(t)$ using any of the numerical integrators we have studied. Of course, the upper limit must be finite, presumably the time required for the drug to be fully ingested.

We conclude this section with a simple nonlinear system model.

EXAMPLE 6.4

The cooling of a high-temperature oven is governed by the following differential equation (McClamroch 1980):

$$C \frac{d\tilde{T}}{dt} = -K_c(\tilde{T} - T_0) - K_r(\tilde{T}^4 - T_0^4) \quad (6.142)$$

where

$\tilde{T} = \tilde{T}(t)$ is the oven temperature ($^{\circ}\text{R}$)

T_0 is the surrounding temperature ($^{\circ}\text{R}$)

C is the thermal capacity of the oven

K_c and K_r are convective and radiation heat loss coefficients

Simulate the oven's cooling from an initial temperature of 1000°R if the surrounding temperature is 500°R . Numerical values of the thermal parameters are

$$C = 24 \text{ Btu}/^{\circ}\text{R}, K_c = 8 \text{ Btu/h}/^{\circ}\text{R}, K_r = 2 \times 10^{-8} \text{ Btu/h}/^{\circ}\text{R}^4$$

RK-1 through RK-4 integrators were used to numerically integrate the derivative function

$$f(\tilde{T}) = \frac{d\tilde{T}}{dt} = -\frac{K_c}{C}(\tilde{T} - T_0) - \frac{K_r}{C}(\tilde{T}^4 - T_0^4) \quad (6.143)$$

The results are shown in [Table 6.4](#). The integration step size was chosen for each integration method to make the total number of derivative function evaluations and, hence, the computational effort, roughly the same.

The last column is labeled "Exact"; however, the exact solution is not easily obtained. The numbers in the last column were obtained using RK-4 integration with a small enough step size ($T = 0.005$ h) to generate approximate values in agreement with the exact solution values to at least one place after the decimal point. How can we check this assumption?

[Figure 6.7](#) shows the results of using RK-1 and RK-4 to integrate the derivative function, Equation 6.143. RK-1 is used with a step size $T = 0.3$ h and RK-4 is used with a step size $T = 0.8$ h. Note that RK-4 with $T = 0.8$ h produces more accurate results than RK-1 with $T = 0.3$ h.

EXERCISES

- 6.1 Show that the difference equation resulting from using RK-m integration to approximate the behavior of $dx/dt = f(t, x) = ax$ is

$$x_A(i+1) = \left[1 + aT + \frac{1}{2!}(aT)^2 + \frac{1}{3!}(aT)^3 + \cdots + \frac{1}{m!}(aT)^m \right] x_A(i)$$

6.2 The mass m of a radioactive material in a container decays according to the equation

$$\frac{dm}{dt} = -km \quad \text{subject to } m(0) = m_0$$

where

$$m = m(t)$$

k is a constant for the specific radioactive material

m_0 is the initial mass of radioactive material in the container

TABLE 6.4

Comparison of RK-1, 2, 3, 4 Integrators with Different Step Sizes and Exact Solution

RK-1			RK-2			RK-3			RK-4			"Exact"
$T = 0.1$			$T = 0.2$			$T = 0.3$			$T = 0.4$			
i	t_i	$\tilde{T}_A(i)$	$\tilde{T}_A(i)$									
0	0	1000.0	0	0	1000.0	0	0	1000.0	0	0	1000.0	1000.0
2	0.2	841.0	1	0.2	864.1							859.9
3	0.3	793.1				1	0.3	806.2				813.2
4	0.4	755.6	2	0.4	784.9				1	0.4	774.3	775.5
6	0.6	699.8	3	0.6	730.8	2	0.6	713.5				718.0
8	0.8	660.0	4	0.8	691.1				2	0.8	675.6	676.2
9	0.9	644.0				3	0.9	656.1				659.3
10	1.0	630.1	5	1.0	660.6							644.4
12	1.2	607.0	6	1.2	636.4	4	1.2	617.2	3	1.2	619.1	619.5
16	1.6	573.9	8	1.6	600.8				4	1.6	583.4	583.6
20	2.0	552.1	10	2.0	576.3				5	2.0	559.4	559.6
30	3.0	522.7	15	3.0	540.7	10	3.0	526.2				526.7
40	4.0	510.2	20	4.0	523.3				10	4.0	512.3	512.3
48	4.8	505.4	24	4.8	515.4	16	4.8	506.6	12	4.8	506.7	506.7

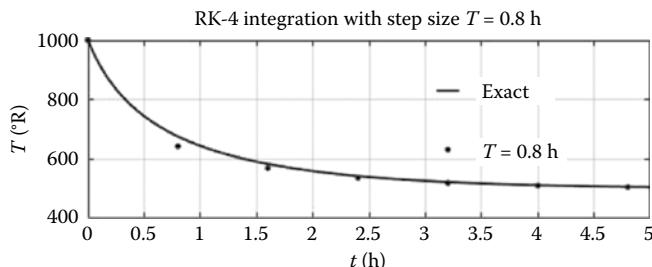
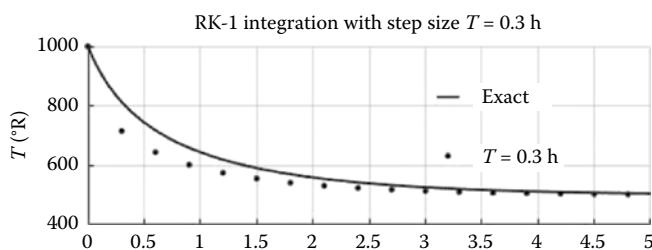


FIGURE 6.7 RK-1 and RK-4 solution of oven cooling with three different step sizes.

The half-life of a radioactive material, $T_{1/2}$, is related to the decay constant k by $T_{1/2} = \ln 2/k$.

Suppose the half-life of a radioactive material is 2 years and there is initially 1 kg of material present in the container.

- Use RK-1 through RK-4 integration to obtain approximations of the mass of radioactive material present in the container every month until less than 0.25 kg of material remains.
- Compare the results from part (a) with the exact solution for $m(t)$.

6.3 The amount of fish in a lake at any time is assumed to obey the following logistic growth model:

$$\frac{dx}{dt} = Kx(M - x) - u$$

where

$x = x(t)$ is the number of fish present

$u = u(t)$ is the rate at which fish are harvested

Nominal values of the system parameters are $K = 2.5 \times 10^{-7}$ (fish-day) $^{-1}$ and $M = 2,00,000$ fish. The lake is initially stocked with 50,000 fish.

- Use RK-4 integration with appropriate step size T to approximate the fish population in the absence of harvesting. Plot the results.
- Plot the exact solution $x(t) = M/[1 - [1 - M/x(0)]e^{-KMt}]$ on the same graph.
- Repeat part (a) for a constant harvesting rate $u = 2750$ fish/day.
- Repeat part (a) for a constant harvesting rate $u = 2250$ fish/day.

6.4 For the system in Example 6.1, let $\alpha = 0.15$, $c = 0.05$, $m_0 = 1$, $\bar{F} = 0$, and $v(0) = 2$. The derivative function is

$$\frac{dv}{dt} = f(t, v) = \frac{-3v}{20-t}, \quad 0 \leq t \leq 16$$

and the exact solution is

$$v(t) = 2(1 - 0.05t)^3, \quad 0 \leq t \leq 16$$

Show that $v(0.5)$ and $v_A(1)$ are identical, where $v_A(1)$ is the result of using RK-3 integration with a step size $T = 0.5$.

- Repeat Example 6.3 using RK-1 and RK-2 integrator with a step size $T = 1$ min. Compare the accuracy of each with the RK-4 method results shown in [Table 6.3](#).
- In Example 6.3, choose the states $x_1 = m_1$ and $x_2 = m_2$ and the outputs $y_1 = x_1$ and $y_2 = x_2$.
 - Find the matrices A , B , C , and D in the continuous-time state variable model of the system.
 - Apply RK-4 integration with $T = 1$ min to obtain approximate solutions for the amount of drug in the gastrointestinal tract and the bloodstream. Compare the results of drug amounts in the bloodstream with results in [Table 6.3](#).
 - Use RK-4 integration with $T = 1$ min to find an approximate solution for the case where $m_1(0) = 20$ mg, $m_2(0) = 0$ mg, and $u(t) = 0$, $t \geq 0$.
 - Verify the results in part (c) by using

$$\underline{x}_A(i) = \left[I + TA + \frac{1}{2!}(TA)^2 + \frac{1}{3!}(TA)^3 + \frac{1}{4!}(TA)^4 \right]^i \underline{x}(0)$$

- 6.7 Approximate the amount of drug ingested by an individual using RK-1 through RK-4 integration (using an appropriate step size T) when the drug ingestion rate is
- $u(t) = M e^{-t/\tau}$, $t \geq 0$ ($M = 5$ mg/min, $\tau = 4$ min)
 - $u(t) = M e^{-t/\tau}$, $t \geq 0$ ($M = 1$ mg/min, $\tau = 45$ min)
 - $u(t) = A \sin(2\pi t/P)$, $0 \leq t \leq P/2$ ($A = 2$ mg/min, $P = 30$ min)
 - $u(t)$ is available in tabular form in the [Table E6.7](#):

TABLE E6.7

t (min)	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
$u(t)$ (mg/min)	0.0	0.4	1.0	3.0	2.2	1.4	0.8	0.4	0.2	0.1	0.0

- 6.8 Since RK-4 integrators require four times the number of derivative function evaluations as RK-1 integrators and RK-2 integrators require twice the number of derivative function evaluations as RK-1 integrators, it is reasonable to compare the three integrators when the computational effort is roughly the same for all three. In other words, if the step size for the RK-1 integrator is T , then the RK-2 and RK-4 integrators should be run with step sizes $2T$ and $4T$, respectively. Simulate the response of the system in Example 6.1 where

$$\frac{dv}{dt} = f(t, v) = 5 \left[\frac{2-v}{20-t} \right], \quad v(0) = 0$$

using RK-1, RK-2 (improved or modified Euler), and the classic RK-4 integrator using step sizes of 0.25, 0.5, and 1 s, respectively. Enter the results in [Table E6.8](#). Comment on the results.

TABLE E6.8

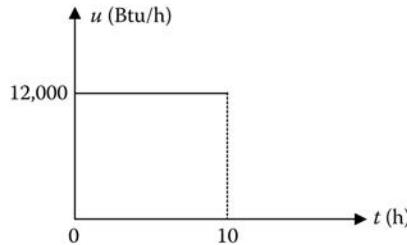
RK-1 ($T = 0.25$)				RK-2 ($T = 0.5$)				RK-4 ($T = 1$)			
i	t_i	$v_A(i)$	$v(t_i)$	i	t_i	$v_A(i)$	$v(t_i)$	i	t_i	$v_A(i)$	$v(t_i)$
0	0	0.00000	0.00000	0	0	0.00000	0.00000	0	0	0.00000	0.00000
4	1			2	1			1	1		
8	2			4	2			2	2		
12	3			6	3			3	3		
16	4			8	4			4	4		
20	5			10	5			5	5		
24	6			12	6			6	6		
28	7			14	7			7	7		
32	8			16	8			8	8		

- 6.9 The model for finding the temperature in the oven of Example 6.4 when there is an internal heat source is

$$C \frac{d\tilde{T}}{dt} = -K_c(\tilde{T} - T_0) - K_r(\tilde{T}^4 - T_0^4) + u$$

where $u = u(t)$ is the heat source. Suppose the oven and its surroundings are in equilibrium at a temperature of 600°R .

- Simulate the transient response of the oven temperature using an RK-2 integrator with step size $T = 0.25$ h when the heat transferred to the oven is as shown in [Figure E6.9](#):

**FIGURE E6.9**

- b. Find the exact solution for $\tilde{T}(t)$ to one place after the decimal point by solving for $\tilde{T}_A(i) \approx (t_i)$, $i = 0, 1, 2, 3, \dots$ using RK-4 integration with step size $T = 0.001$ h. Compare the results with those from part (a) and (b).

6.3 ADAPTIVE TECHNIQUES

The computational efficiency of RK methods can be improved if the step size is allowed to vary during a simulation. A reasonable criterion must be established for determining when it is appropriate to modify the step size and by how much. The criterion is usually based on an estimate of the local truncation error as the simulation progresses with time. If an estimate of the local truncation error is outside an acceptable tolerance, then it is possible to either reduce the step size when the estimated error is too large or quite possibly increase the step size if it appears that the error is unnecessarily small. Techniques for estimating the local truncation error and modifying the step size, if warranted, are referred to as adaptive step size control.

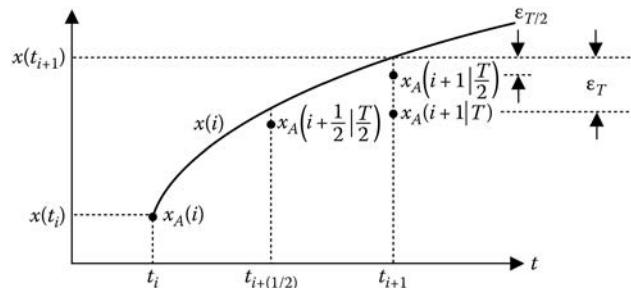
6.3.1 REPEATED RK WITH INTERVAL HALVING

If we use the local truncation error as the basis for determining when the step size needs adjustment, then a method is needed for approximating it. One approach requires that we obtain two estimates of the updated state from an RK integrator and use the difference to estimate the local truncation error. Interval halving refers to the case where the step sizes differ by a factor of 2.

Refer to Figure 6.8 to understand how the method works. Let $x_A(i+1|T)$ be the approximate solution to $\dot{x} = f(t, x)$ at t_{i+1} obtained using a step size of T . Similarly, let $x_A(i+1|T/2)$ be the approximate solution to $\dot{x} = f(t, x)$ at t_{i+1} obtained after two steps using a step size of $T/2$.

Assume $x_A(i)$ is exact, that is, $x_A(i) = x(t_i)$. It follows that

$$x(t_{i+1}) = x_A(i+1|T) + \varepsilon_T \quad (6.144)$$

**FIGURE 6.8** Illustration of interval halving for estimation of local truncation error.

$$x(t_{i+1}) = x_A \left(i + 1 \left| \frac{T}{2} \right. \right) + \varepsilon_{T/2} \quad (6.145)$$

where ε_T and $\varepsilon_{T/2}$ are the local truncation errors in $x_A(i + 1|T)$ and $x_A(i + 1|T/2)$, respectively.

From Equations 6.144 and 6.145,

$$x_A(i + 1|T) + \varepsilon_T = x_A \left(i + 1 \left| \frac{T}{2} \right. \right) + \varepsilon_{T/2} \quad (6.146)$$

Suppose the numerical integrator is an RK-4 with local truncation error $\varepsilon_T \sim O(T^5)$. Then ε_T can be expressed as

$$\varepsilon_T = cT^5 \quad (6.147)$$

and $\varepsilon_{T/2}$, which is the sum of local truncation errors for the two half-intervals, is given by

$$\varepsilon_{T/2} = c \left(\frac{T}{2} \right)^5 + c \left(\frac{T}{2} \right)^5 = 2c \left(\frac{T}{2} \right)^5 = \frac{1}{16} cT^5 = \frac{1}{16} \varepsilon_T \quad (6.148)$$

In reality, c in Equation 6.147 and the two occurrences of c in Equation 6.148 are different and depend on the derivative function and the intervals; however, for suitably small T , the differences are negligible. Eliminating ε_T from Equations 6.146 and 6.148 gives

$$x_A(i + 1|T) + 16\varepsilon_{T/2} = x_A \left(i + 1 \left| \frac{T}{2} \right. \right) + \varepsilon_{T/2} \quad (6.149)$$

Solving for $\varepsilon_{T/2}$ in Equation 6.149 gives

$$\varepsilon_{T/2} = \frac{x_A \left(i + 1 \left| \frac{T}{2} \right. \right) - x_A(i + 1|T)}{15} \quad (6.150)$$

$\varepsilon_{T/2}$ in Equation 6.150 is an estimate of the local truncation error of the RK-4 integrator when the step size is $T/2$. It can be used to adjust the step size in subsequent calculations. For example, Table 6.5 shows a possible approach to step size adjustment using predetermined tolerance limits ε_L and ε_U .

The truncation error $\varepsilon_{T/2}$ can be added to $x_A(i + 1|T/2)$ to obtain a fifth-order accurate estimate of $x(t_i + 1)$, that is, a new estimate $x_A(i + 1)$ with local truncation error $\varepsilon_T \sim O(T^5)$. This gives

$$\varepsilon_L \leq |\varepsilon_{T/2}| \leq \varepsilon_U \quad (6.151)$$

TABLE 6.5
Step Size Adjustment Based on Outcome of $|\varepsilon_{T/2}|$

Outcome	Action (Next Integration Step)
$\varepsilon_L > \varepsilon_{T/2} $	Double current step size
$\varepsilon_L \leq \varepsilon_{T/2} \leq \varepsilon_U$	Keep current step size
$ \varepsilon_{T/2} > \varepsilon_U$	Halve current step size

$$= \frac{16x_A\left(i+1\left|\frac{T}{2}\right.\right) - x_A(i+1|T)}{15} \quad (6.152)$$

The following example includes a one-step numerical integrator with the step size determined by the interval halving method previously described.

EXAMPLE 6.5

A cone-shaped tank is filling with water at a constant rate $F_1(t) = \bar{F}$ as shown in [Figure 6.9](#). The initial level is h_0 . Water evaporates from the tank at a rate proportional to the surface area of liquid. The constant of proportionality is α .

- Find the derivative function in the continuous-time model of the tank.
- For $\bar{F} = \pi \text{ ft}^3/\text{min}$, $\alpha = 0.01 \text{ ft}/\text{min}$, and $h_0 = 10 \text{ ft}$, estimate the local truncation error in the RK-4 estimate $h_A(1)$ when $T = 1$ in using interval halving, that is, find $\varepsilon_{T/2}$.
- Use $\varepsilon_{T/2}$ to obtain a fifth-order accurate estimate of $h(T)$.
- Simulate the tank dynamics for a period of time sufficient to allow the tank level to increase by 90% of the ultimate change in level. Use an adaptive step size based on the algorithm in [Table 6.5](#) with $\varepsilon_L = 10^{-13}$ and $\varepsilon_U = 10^{-11}$.
- Find the analytical solution of the continuous-time model and plot it along with the simulated solution.

- The continuous-time model of the tank is

$$\frac{dV}{dt} = F_1(t) - \alpha S \quad (6.153)$$

where

V is the volume of water in the tank

S is the surface area of water at the top where evaporation occurs

For the conical shape in [Figure 6.9](#),

$$V = \frac{\pi h^3}{12}, \quad S = \frac{\pi h^2}{4} \quad (6.154)$$

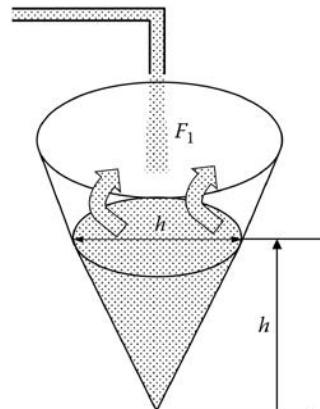


FIGURE 6.9 Conical tank with evaporation.

TABLE 6.6
Results after One Step of Interval Halving Using RK-4 ($T = 1$ min)

	$t_1 = T$	$t_{1/2} = T/2$	$t_1 = T/2 + T/2$
k_1	0.030000000000000	0.030000000000000	0.02988050771064
k_2	0.02988026946101	0.02994006743256	0.02982108077964
k_3	0.02988074623874	0.02994018702867	0.02982119883721
k_4	0.02976202120809	0.02988050764055	0.02976202170051
$h_A(1 T) = 10.02988067543460$	$h_A\left(1\left \frac{T}{2}\right.\right) = 10.01497008471359$	$h_A\left(1\left \frac{T}{2}\right.\right) = 10.02988067543399$	

Substituting Equation 6.154 into Equation 6.153 and solving for the derivative function give

$$\frac{dh}{dt} = \frac{4\bar{F}}{\pi h^2} - \alpha \quad (6.155)$$

b. Using the given values for α and \bar{F} gives

$$\frac{dh}{dt} = \frac{4}{h^2} - 0.01 \quad (6.156)$$

Starting from the initial point $(0, h_0) = (0, 10)$, the results from interval halving after one integration step are given in Table 6.6.

The second column contains the results from a single-step RK-4 integrator with step size $T = 1$ min. The last two columns list the results from two consecutive steps of an RK-4 integrator with step size $T = 0.5$ min. From Equation 6.150, an estimate of the local truncation error in $h_A(1|(T/2))$ is given by

$$\begin{aligned} \varepsilon_{T/2} &= \frac{h_A\left(1\left|\frac{T}{2}\right.\right) - h_A(1|T)}{15} \\ &= \frac{10.02988067543399 - 10.02988067543460}{15} \\ &= -0.40619359727619 \times 10^{-13} \end{aligned}$$

c. From Equation 6.152, the fifth-order accurate estimate of $h(T)$ is

$$\begin{aligned} h_A(1) &= \frac{16h_A\left(1\left|\frac{T}{2}\right.\right) - h_A(1|T)}{15} \\ &= \frac{16(10.02988067543399) - 10.02988067543460}{15} \\ &= 10.02988067543395 \end{aligned}$$

d. The steady-state tank level is easily obtained by setting the derivative function to zero in Equation 6.155 and solving for $h = h(\infty)$.

$$\Rightarrow h(\infty) = \sqrt{\frac{4\bar{F}}{\pi\alpha}} = \sqrt{\frac{4\pi}{\pi(0.01)}} = 20 \text{ ft} \quad (6.157)$$

TABLE 6.7
**Simulation Time Interval and the Constant Step Size T
Using Interval Halving for Adaptive RK-4 Integration**

Time Interval	Step Size, T
$0 \leq t \leq 1$	1
$1 \leq t < 69$	2
$69 \leq t < 209$	4
$209 \leq t < 313$	8
$313 \leq t < 1673$	16

The tank dynamics were simulated using RK-4 integration with interval halving for step size control in “Ch6_Ex6_5.m.” The ultimate change in tank level is $h(\infty) - h(0) = 20 - 10 = 10$ ft. The simulation terminates when $h_A(i)$ exceeds the level $h(0) + 0.9[h(\infty) - h(0)] = 10 + 0.9(10) = 19$ ft.

Table 6.7 summarizes how the step size was changed in accordance with the given tolerances on the estimated local truncation error. Note the significant increase in step size from the starting value of $T = 1$ min as the simulation progresses.

- e. The analytical solution is an implicit function for $h(t)$. The derivation is left as an exercise. The result is

$$100 \left[10 - h(t) - 10 \ln \left(\frac{60 - 3h(t)}{h(t) + 20} \right) \right] = t \quad (6.158)$$

Data points were obtained by increasing $h(t)$ from 10 to 19 ft in small increments, solving for the corresponding value of t , and plotted in Figure 6.10 with t values along the abscissa. The simulated results (every fourth point) are also plotted demonstrating the close agreement with the exact solution. Notice that the step size is progressively increased as the slope, that is, derivative of the solution, gradually decreases.

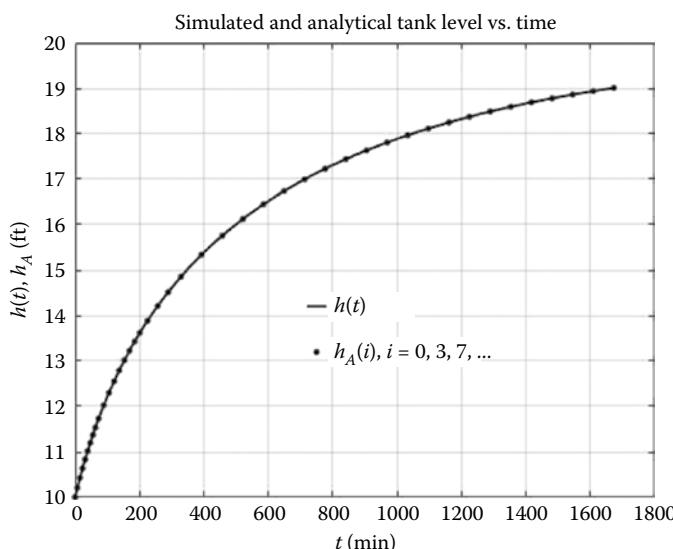


FIGURE 6.10 Results of RK-4 integration with adaptive step size control.

The average of the estimated local truncation errors is

$$\bar{\varepsilon}_{T/2} \Rightarrow \sum_{i=1}^{169} (\varepsilon_{T/2})_i = 1.6653 \times 10^{-12} \quad (6.159)$$

In this example, the simulated time was approximately 1670 min, which would have required 1670 integration steps if the step size had remained constant at $T = 1$ min. With adaptive step size control, the number of integration steps was 169, a nearly 90% reduction. Of course, each of the 169 integration steps requires two passes, one using a full step size and the other using two half-steps. The number of derivative function evaluations for each method is summarized below.

6.3.2 CONSTANT STEP SIZE ($T = 1$ min)

$$\text{Total number of function evaluations} = \frac{1670 \text{ min}}{1 \text{ min/step}} \times 4 \frac{\text{function evaluations}}{\text{step}} = 6680$$

6.3.3 ADAPTIVE STEP SIZE (INITIAL $T = 1$ min)

1. Number of function evaluations (first pass)

$$169 \text{ steps} \times 4 \frac{\text{function evaluations}}{\text{step}} = 676$$

2. Number of new function evaluations (second pass)

$$169 \text{ steps} \times 3 \frac{\text{function evaluations}}{\text{first half interval}} = 507$$

$$169 \text{ steps} \times 4 \frac{\text{function evaluations}}{\text{second half interval}} = 676$$

$$\text{Total number of function evaluations} = 676 + 507 + 676 = 1859.$$

The number of derivative function evaluations has been reduced by over 72%. This comparison is clearly sensitive to the order of the RK integrator used as well as the constant step size T and total simulation time. For example, halving the value of T from 1 to 0.5 min doubles the number of derivative function evaluations in the first case where the step size remains constant. With interval halving and adaptive step size control, the total number of steps would remain nearly the same regardless of the initial step size. Consequently, the total number of derivative function evaluations would remain about the same in either case.

The step size control logic is also significant. The adaptive step size control is typically more complex (Borse 1997; Chapra and Canale 2002) than the simple approach presented here where the new step size is either one half, the same, or twice the current step size.

Since an implicit solution for $h(i)$ is known (Equation 6.158), it is possible to compare the estimated local truncation error with the actual local truncation error, although not in a straightforward manner due to the implicit nature of the solution.

6.3.4 RK-FEHLBERG

In the case of RK-4 integration, the interval halving method requires seven additional derivative function evaluations for the second pass over the two half-intervals. A total of 11 function

evaluations are required for each interval, independent of the interval size. An alternative method for estimating the local truncation error is based on the difference of two different order RK integrators over the same integration time step. By choosing two RK integrators with several common points for the derivative function evaluations, efficiency is improved significantly compared to the interval halving method.

The RK–Fehlberg method employs RK-4 and RK-5 integrators where the four function evaluations k_1, k_2, k_3 , and k_4 of the RK-4 integrator are used in the RK-5 integrator as well. Recall that RK-5 integration methods require six function evaluations per step. Therefore, RK–Fehlberg methods combining RK-4 and RK-5 integrators employ a total of six derivative function evaluations per interval.

A common RK–Fehlberg integrator is given as follows (Rao 2002):

$$k_1 = f[t_i, x_A(i)] \quad (6.160)$$

$$k_2 = f\left[t_i + \frac{1}{4}T, x_A(i) + \frac{1}{4}Tk_1\right] \quad (6.161)$$

$$k_3 = f\left[t_i + \frac{3}{8}T, x_A(i) + \frac{3}{32}Tk_1 + \frac{9}{32}Tk_2\right] \quad (6.162)$$

$$k_4 = f\left[t_i + \frac{12}{13}T, x_A(i) + \frac{1932}{2197}Tk_1 - \frac{7200}{2197}Tk_2 + \frac{7296}{2197}Tk_3\right] \quad (6.163)$$

$$k_5 = f\left[t_i + T, x_A(i) + \frac{439}{216}Tx_1 - 8Tk_2 + \frac{3680}{513}Tk_3 + \frac{845}{4104}Tk_4\right] \quad (6.164)$$

$$k_6 = f\left[t_i + \frac{1}{2}T, x_A(i) - \frac{8}{27}Tx_1 + 2Tk_2 - \frac{3544}{2565}Tk_3 + \frac{1859}{4104}Tk_4 - \frac{11}{40}Tk_5\right] \quad (6.165)$$

The estimate of $x[(i + 1)T]$ using RK-4 integration is

$$x_A(i+1) = x_A(i) + T\left[\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5\right] \quad (6.166)$$

The RK-5 estimate of $x[(i + 1)T]$ and eventual updated state is

$$x_A(i+1) = x_A(i) + T\left[\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\right] \quad (6.167)$$

The local truncation error incurred in the i th integration interval is estimated from the difference of Equations 6.167 and 6.166. Thus,

$$\text{Estimate of } (\varepsilon_T)_i = T\left[\frac{1}{360}k_1 - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{1}{50}k_5 + \frac{2}{55}k_6\right] \quad (6.168)$$

Example 6.6 illustrates the use of RK–Fehlberg integration, specifically, the RK-4 and RK-5 methods previously described.

EXAMPLE 6.6

A motor boat is being driven across a river L ft wide to the opposite side as shown in Figure 6.11. The boat departs from point 0, the origin of an x - y coordinate system, attempting to reach a point H ft upstream. The boat travels at a constant speed v_b mph relative to the water that flows downstream at a speed of v_r mph. The boat is continuously steered in the direction of its intended destination. The boat's heading is given by the angle θ as shown in the diagram. Numerical values of the system parameters are $L = 1000$ ft, $H = 5000$ ft, $v_b = 15$ mph, and $v_r = 5$ mph.

- Choose the state variables as $x(t)$ and $y(t)$ and obtain expressions for the state derivative functions in terms of x and y and the system parameters L , H , v_b , and v_r .
- Use the “ode45” numerical integrator in Simulink and simulate the boat’s x and y position as a function of time. Plot x vs. t , y vs. t , and θ vs. t .
- Plot the steering angle θ vs. horizontal position x .
- Find an expression for the derivative dy/dx in terms of x and y and the system parameters L , H , v_b , and v_r .
- Write a program to implement the RK–Fehlberg method to numerically integrate dy/dx . Adjust the step size when the estimated local truncation error falls outside an acceptable tolerance range. Choose the initial integration step to be $T = 1$ ft.
- Find the exact solution for $y(x)$ and compare it with the simulated results.

- From the diagram, the state derivatives are

$$\frac{dx}{dt} = v_b \cos \theta = v_b \frac{L - x}{[(L - x)^2 + (H - y)^2]^{1/2}} \quad (6.169)$$

$$\frac{dy}{dt} = -v_r + v_b \sin \theta \quad (6.170)$$

$$= -v_r + v_b \frac{H - y}{[(L - x)^2 + (H - y)^2]^{1/2}} \quad (6.171)$$

- A Simulink diagram of the system is shown in Figure 6.12.

Selecting the “ode45” integrator with maximum step size set to 20 and relative tolerance equal to 10^{-6} produces graphs of the state variables $x(t)$ and $y(t)$ and the additional output $\theta(t)$ shown in Figure 6.13.

The exact solutions for $x(t)$, $y(t)$, and $\theta(t)$ were approximated using Simulink’s RK-4 integrator with a very small step size, namely, $T = 0.01$ s. The adaptive step size control quickly adjusts the step size to its maximum value and maintains it at the upper limit until the simulation is nearly complete. Comparison with the “exact” ($T = 0.01$ s) solution shows that the truncation errors are minimal.

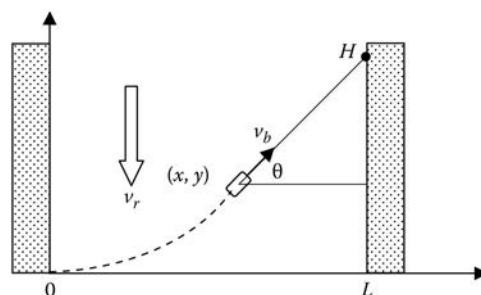


FIGURE 6.11 Boat trajectory crossing the river.

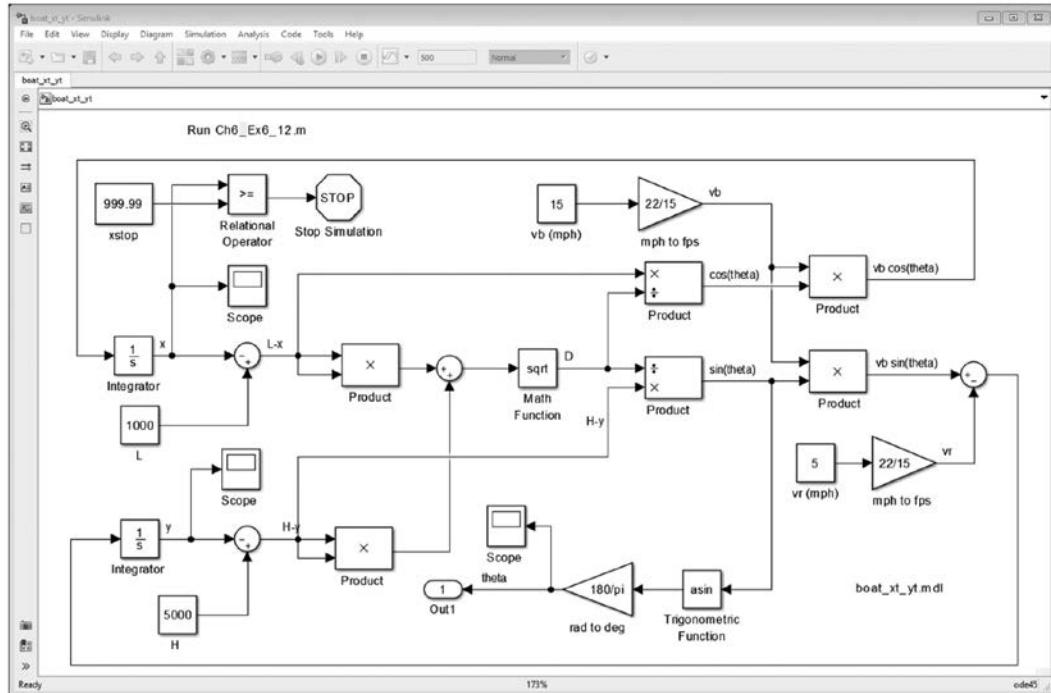


FIGURE 6.12 Simulink diagram of boat crossing.

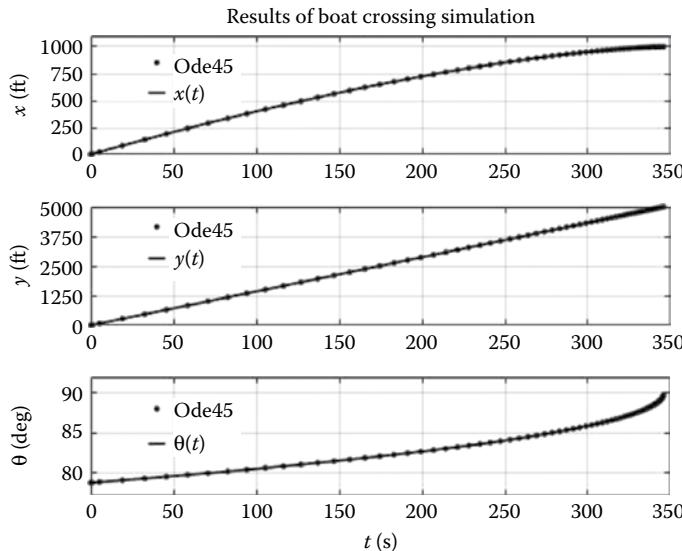


FIGURE 6.13 Time histories of state variables $x(t)$ and $y(t)$ and output $\theta(t)$ using Simulink variable-step integrator “ode45” and “exact” solutions (RK-4 with $T = 0.01$).

- c. A plot of steering angle θ vs. horizontal position x is shown in [Figure 6.14](#). Note that the river current causes the boat to be steered at increasingly greater angles as it approaches the right bank of the river.
- d. States $x(t)$ and $y(t)$ represent a parametric description of the boat’s trajectory $y = y(x)$. The trajectory can be found in one of two ways. First, the parameter t can be eliminated

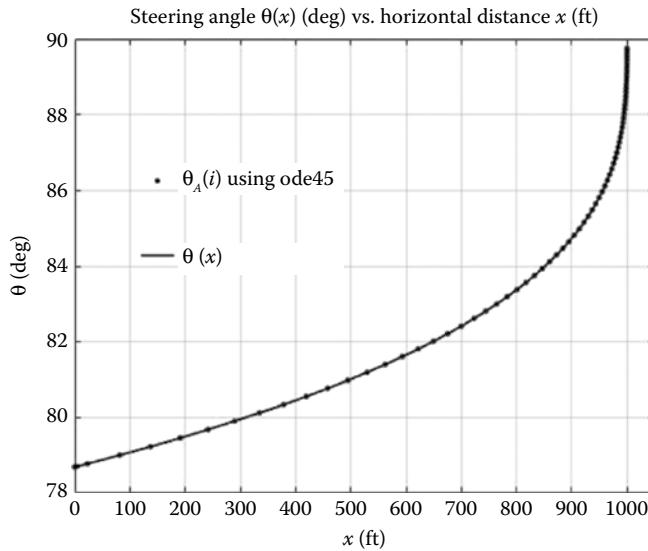


FIGURE 6.14 Steering output θ vs. horizontal location x with Simulink variable-step integrator “ode45” and “exact” solution.

from equations for the states $x(t)$ and $y(t)$. Since we have not developed the solutions for $x(t)$ and $y(t)$, we resort to the second approach, namely, integration of the derivative dy/dx .

Dividing Equation 6.170 by Equation 6.169 gives

$$\frac{dy}{dx} = \frac{-v_r + v_b \sin\theta}{v_b \cos\theta} \quad (6.172)$$

Expressing $\sin\theta$ and $\cos\theta$ in terms of the distances x , y , L , and H (see Figure 6.11) and simplifying the result yield

$$\frac{dy}{dx} = \frac{H-y}{L-x} - \frac{v_r}{v_b} \left[1 + \left(\frac{H-y}{L-x} \right)^2 \right]^{1/2} \quad (6.173)$$

- e. The RK–Fehlberg Equations 6.160 through 6.168 are solved in “Ch6_Ex3_2.m.” A simulation summary is shown in Table 6.8.

TABLE 6.8
Summary of RK–Fehlberg Simulation Results for Boat Crossing

Minimum tolerance	10^{-7} ft
Minimum tolerance	10^{-5} ft
Minimum step size	0.1 ft
Minimum step size	25 ft
Number of integration steps	87
Average step size	11.49 ft
Average estimated local truncation error	1.6427×10^{-4}

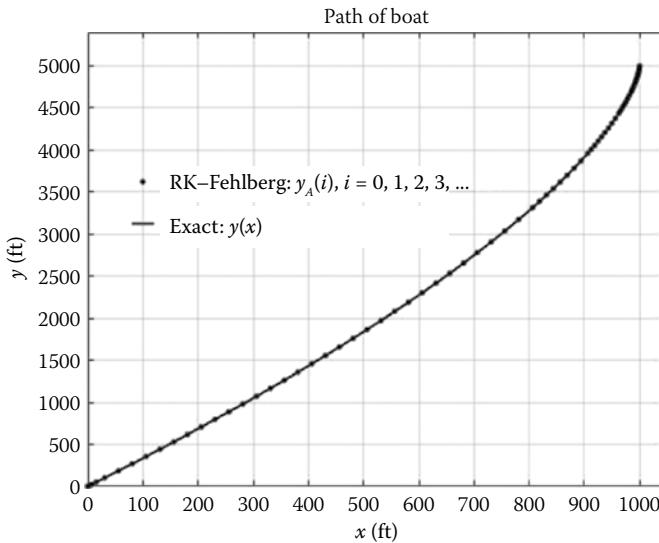


FIGURE 6.15 Simulated (RK–Fehlberg) and exact solutions for $y(x)$ in boat crossing.

f. Derivation of the exact solution to Equation 6.173 is left as an exercise. The result is

$$y(x) = H - \frac{L-x}{2} \left[c(L-x)^{-k} - \frac{(L-x)^k}{c} \right] \quad (6.174)$$

where

$$c = L^{k-1}(H + \sqrt{L^2 + H^2})$$

$$k = v_r/v_b$$

Plots of the approximate solution $y_A(i)$ obtained in part (e) and the exact solution, Equation 6.174, are shown in [Figure 6.15](#).

EXERCISES

6.10 In Example 6.5,

- Find the analytical solution to the continuous-time model of Equation 6.155.
 - Write a program to numerically integrate the derivative function using RK-4 with adaptive control of step size. Compare your results with those in [Tables 6.6](#) and [6.7](#).
 - Experiment with the tolerances used to establish the step size, and plot the results on the same graph with the analytical solution.
 - Is the tank initially in equilibrium? Explain. Find the constant flow in \bar{F}_1 for which the tank is in equilibrium when the height of water is 15 ft.
 - With 15 ft of water in the tank and equilibrium conditions established, the flow in is decreased by 50%. Simulate the response using RK-4 with step size control.
 - For the same conditions as in part (e), find the estimated and true local truncation errors resulting from the use of RK-4 numerical integration when the water level is 14.9 ft.
- Hint:* Find the actual time required for the tank level to reach 14.9 ft, and use that value as the initial integration step size.

6.11 In Example 6.6,

- a. Find the analytical solution $y = y(x)$ to Equation 6.173 repeated as follows:

$$\frac{dy}{dx} = \frac{H-y}{L-x} - \frac{v_r}{v_b} \left[1 + \left(\frac{H-y}{L-x} \right)^2 \right]^{1/2}$$

Hint: Let $\hat{x} = L - x$, $\hat{y} = H - y$, introduce $u = \hat{y}/\hat{x}$, and obtain an implicit solution of the separable differential equation in u .

- b. Compute the estimated and actual local truncation errors using RK-4 with interval halving to adjust the step size. Choose the initial step size $T = 1$ ft.

6.12 Find the trajectory of the boat in Example 6.6, assuming it is steered continuously at the destination point (L, H) , if the river current varies sinusoidally according to

$$v_r(x) = A \sin \frac{\pi}{L} x, \quad 0 \leq x \leq L$$

where $A = 10$ mph.

6.13 In the boat-crossing problem of Example 6.6, suppose the boat steering angle θ is an input to the continuous-time model.

- a. Find the state derivative functions in $dx/dt = f_1(x, y, \theta)$ and $dy/dt = f_2(x, y, \theta)$.

For parts (b) through (d), find the analytical solution and check your answer using Simulink with the “ode45” solver.

- b. The steering angle θ is held constant at the initial heading of the destination point, that is,

$$\theta(t) = \bar{\theta} = \tan^{-1} \left(\frac{H}{L} \right), \quad t \geq 0$$

Find the location of the boat when it reaches the other side. Use the values of v_r , v_b , L , and H from Example 6.6.

- c. The captain wishes to cross the river and reach the opposite shore line at $x = L$, $y = 0$, which is directly across from where he started. Find the constant heading $\bar{\theta}$, which allows him to accomplish this. Assume the river current is constant at $v_r = 6$ mph and the boat moves at a constant speed of $v_b = 24$ mph. Plot the boat’s trajectory.

- d. Make a plot of $y(L)$ vs. $\bar{\theta}$ for $0 \leq \bar{\theta} < \pi/2$ where $y(L)$ is the y coordinate of the location where the boat reaches the opposite side of the river. Assume $v_r = 10$ mph and $v_b = 25$ mph.

- e. The captain observes a large fish swimming upstream at a constant speed of $v_f = 6$ mph in the middle of the river ($x = L/2$). Starting from $(0, 0)$, he begins to steer directly at the fish when it is directly across from him, that is, located at $(L/2, 0)$. Find and plot the boat’s trajectory until it catches up with the fish if the river current is 0 mph and the boat speed is 10 mph.

6.14 A hydraulic accumulator is shown in Figure E6.14. Its purpose is to damp fluctuations in the input flow rate $f_i(t)$ caused by pressure peaks upstream. The flow exits downstream of the accumulator through a linear resistance. The continuous-time model for the pressure $p(t)$ in the accumulator section is (Palm 1983)

$$\frac{A^2}{k} \frac{dp}{dt} = f_i - \frac{1}{R}(p - p_0)$$

where

A is the area of the accumulator plate

k is the spring constant

R is the fluid resistance

The input flow rate is given by

$$f_1(t) = \begin{cases} 0.01 \text{ ft}^3/\text{s}, & t \leq 0 \text{ s} \\ 0.05 \text{ ft}^3/\text{s}, & 0 < t \leq 0.01 \text{ s} \\ 0.01 \text{ ft}^3/\text{s}, & t > 0.01 \text{ s} \end{cases}$$

Numerical values of the system parameters are

$$A = 0.0055 \text{ ft}^2, k = 30 \text{ lb/ft}, R = 10^5 \text{ lb s/ft}^2, p_0 = 14.7 \text{ lb/in.}^2$$

The system is at steady state prior to the pulse input in flow.

- Use the RK–Fehlberg integrator to simulate the transient response of $p(t)$.
- Find the analytical solution for $p(t)$.
- Find the solution for $p(t)$ without the accumulator present.
- Plot the responses from parts (a), (b), and (c) on the same graph.
- Simulate the response for $p(t)$ with Simulink using the “ode45” integrator, and compare the results with those in parts (a) and (b).

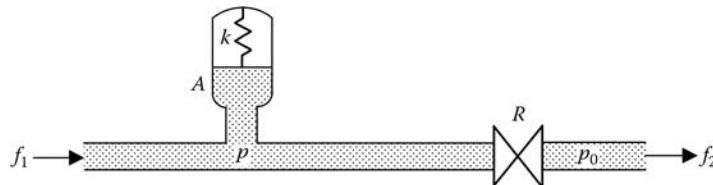


FIGURE E6.14

6.4 MULTISTEP METHODS

RK integrators were classified as one-step methods. The calculations for determining $x_A(i + 1)$, the approximate solution to the continuous-time model

$$\frac{dx}{dt} = f(t, x) \quad (6.175)$$

at $t = t_{i+1}$, relies on the previous estimate $x_A(i)$ and one or more derivative function evaluations on the interval $t_i \leq t \leq t_{i+1}$. The previous state estimate $x_A(i)$ is ignored once $x_A(i + 1)$ has been computed. In contrast, multistep methods exploit knowledge of previous state estimates because they provide information about the local behavior of $x(t)$ that can be used to advance the state.

Formulas for multistep methods are derived by integrating Equation 6.175 from t_i to t_{i+1} ,

$$\int_{x(t_i)}^{x(t_{i+1})} dx = \int_{t_i}^{t_{i+1}} f[t, x(t)] dt \quad (6.176)$$

$$\Rightarrow x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f[t, x(t)] dt \quad (6.177)$$

The integrand $f[t, x(t)]$ is unknown since $x(t)$ is the solution to Equation 6.175.

6.4.1 EXPLICIT METHODS

An m th-order interpolating polynomial $P_m(t)$ that passes through the current derivative $f[t_i, x_A(i)]$ and previous m derivatives $f[t_{i-1}, x_A(i-1)], f[t_{i-2}, x_A(i-2)], \dots, f[t_{i-m}, x_A(i-m)]$ can be used to obtain an approximation of the integral in Equation 6.177 (see Figure 6.16). Replacing the integrand in Equation 6.177 by the interpolating polynomial $P_m(t)$ gives

$$x_A(i+1) = x_A(i) + \int_{t_i}^{t_{i+1}} P_m(t) dt \quad (6.178)$$

where the approximations $x_A(i)$ and $x_A(i+1)$ are used instead of $x(t_i)$ and $x(t_{i+1})$, the actual points on the solution $x(t)$. The integral in Equation 6.178 is equal to the shaded area under the polynomial $P_m(t)$, which has been extrapolated over the current integration interval (t_i, t_{i+1}) .

To illustrate, suppose the polynomial is the linear function passing through $\{t_{i-1}, f[x_A(i-1)]\}$ and $\{t_i, f[x_A(i)]\}$. Then $m = 1$ and

$$P_1(t) = f[t_i, x_A(i)] + \left\{ \frac{f[t_i, x_A(i)] - f[t_{i-1}, x_A(i-1)]}{t_i - t_{i-1}} \right\} (t - t_i) \quad (6.179)$$

Integrating $P_1(t)$ and substituting the result in Equation 6.178 yield after simplifying

$$x_A(i+1) = x_A(i) + \frac{T}{2} \{3f[t_i, x_A(i)] - f[t_{i-1}, x_A(i-1)]\} \quad (6.180)$$

The formula in Equation 6.180 is known as the two-step Adams–Bashforth (AB-2) method. “Two-step” refers to the use of two intervals, (t_{i-1}, t_i) and (t_i, t_{i+1}) , to compute the new state $x_A(i+1)$. Note that the method is explicit since $x_A(i+1)$ does not appear on the right-hand side of Equation 6.180.

The Taylor Series expansion of the derivative function $f(t, x)$ leads to an alternative derivation of Equation 6.180, which also provides an expression for the local truncation error of the AB-2 integrator. The Taylor Series expansion of $x(t)$ about t_i evaluated at t_{i+1} is given by

$$x(t_{i+1}) = x(t_i) + \frac{d}{dt} x(t_i)(t_{i+1} - t_i) + \frac{1}{2} \frac{d^2}{dt^2} x(t_i)(t_{i+1} - t_i)^2 + \dots \quad (6.181)$$

$$= x(t_i) + Tf[t_i, x(t_i)] + \frac{T^2}{2} \frac{d}{dt} f[t_i, x(t_i)] + \dots \quad (6.182)$$

where T is the fixed-step size, that is, $T = t_{i+1} - t_i = t_i - t_{i-1} = \dots$. The Taylor Series expansion of the derivative function $f(t, x)$ about t_i evaluated at t_{i-1} is

$$f[t_{i-1}, x(t_{i-1})] = f[t_i, x(t_i)] + \frac{d}{dt} f[t_i, x(t_i)](t_{i-1} - t_i) + \frac{1}{2} \frac{d^2}{dt^2} f[t_i, x(t_i)](t_{i-1} - t_i)^2 + \dots \quad (6.183)$$

$$= f[t_i, x(t_i)] - T \frac{d}{dt} f[t_i, x(t_i)] + \frac{T^2}{2} \frac{d^2}{dt^2} f[t_i, x(t_i)] + \dots \quad (6.184)$$

Solving for $T \frac{d}{dt} f[t_i, x(t_i)]$ in Equation 6.184 and substituting into Equation 6.182 give

$$x(t_{i+1}) = x(t_i) + \frac{T}{2} \{3f[t_i, x(t_i)] - f[t_{i-1}, x(t_{i-1})]\} + \frac{5}{12} T^3 \frac{d^2}{dt^2} f[t_i, x(t_i)] + \dots \quad (6.185)$$

Truncating Equation 6.185 after the linear term and replacing $x(t_{i-1}), x(t_i), x(t_{i+1})$ with $x_A(i-1), x_A(i), x_A(i+1)$ lead to the AB-2 formula in Equation 6.180. Furthermore, since the first term omitted in Equation 6.185 is of order T^3 , the local truncation error $\varepsilon_T \sim O(T^3)$. The global truncation error $E_T \sim O(T^2)$ and AB-2 is said to be second-order accurate.

More accurate Adams–Bashforth integration formulas exist. It is simply a question of the number of points, that is, $m + 1$ in Figure 6.16, used to establish the interpolating polynomial $P_m(t)$. Several higher-order AB integrators are listed as follows using the simpler notation $f_A(i) = f[t_i, x_A(i)]$, $f_A(i-1) = f[t_{i-1}, x_A(i-1)]$, etc.

$$\text{AB-3: } x_A(i+1) = x_A(i) + \frac{T}{12} [23f_A(i) - 16f_A(i-1) + 5f_A(i-2)] \quad (6.186)$$

$$\text{AB-4: } x_A(i+1) = x_A(i) + \frac{T}{24} [55f_A(i) - 59f_A(i-1) + 37f_A(i-2) - 9f_A(i-3)] \quad (6.187)$$

$$\begin{aligned} \text{AB-5: } x_A(i+1) = x_A(i) + \frac{T}{720} &[1901f_A(i) - 2774f_A(i-1) + 2616f_A(i-2) \\ &- 1274f_A(i-3) + 251f_A(i-4)] \end{aligned} \quad (6.188)$$

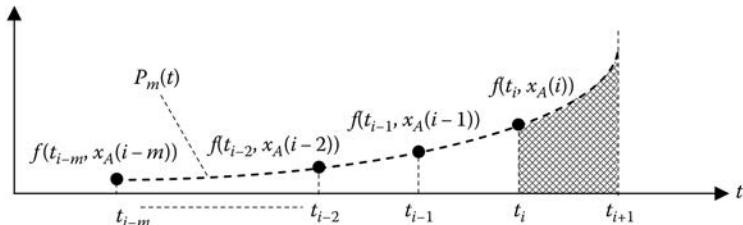
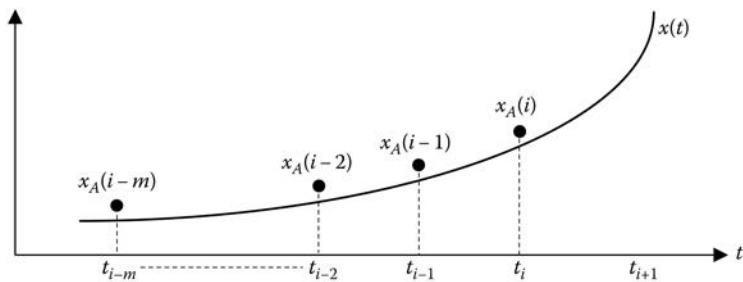


FIGURE 6.16 An m th-order interpolating polynomial for approximating integrand in Equation 6.177.

Local truncation errors for AB integrators are obtained using the Taylor Series expansion approach illustrated for deriving the AB-2 formula. Truncating the respective series to obtain Equations 6.186 through 6.188 results in $(3/8)T^4 \frac{d^3}{dt^3} f[t_i, x(t_i)]$, $(251/720)T^5 \frac{d^4}{dt^4} f[t_i, x(t_i)]$, and $(475/1440)T^6 \frac{d^5}{dt^5} f[t_i, x(t_i)]$ as the first omitted terms in the AB-3, AB-4, and AB-5 formulas. The local and global truncation errors for the third-order accurate AB-3 integrator are $E_T \sim O(T^4)$ and $E_T \sim O(T^3)$, respectively. An m th-order accurate AB- m integrator has a local truncation error $E_T \sim O(T^{m+1})$ and global truncation error $E_T \sim O(T^m)$.

Both AB and RK integrators rely on a weighted sum of derivative function evaluations. In the case of one-step RK integration, the derivative function is evaluated numerous times over a single interval in contrast to the multistep AB integrators, which rely on derivative evaluations from previous intervals.

The m th-order accurate multistep integration formulas are more efficient than one-step methods of identical order because the same derivative function $f_A(i)$ is utilized m times for updating the state over m consecutive intervals. Another way of looking at it is only a single new derivative function evaluation $f_A(i)$ is required to advance the state from $x_A(i)$ to $x_A(i + 1)$. For example, suppose we have just determined the state $x_A(i)$ using AB-3 integration. Since $f_A(i-1)$ and $f_A(i-2)$ are still in memory, only $f_A(i) = f[t_i, x_A(i)]$ is needed to compute the new state $x_A(i + 1)$ in Equation 6.186.

Multistep methods are not self-starting. One approach is to utilize a one-step method for the first several integration steps before transitioning to a multistep formula. Alternatively, a one-step method followed by lower-order multistep methods can be used prior to implementing a specific multistep method. Once again, let us choose the AB-3 integrator for illustration purposes. From Equation 6.186 with $i = 0, 1$

$$x_A(1) = x_A(0) + \frac{T}{12}[23f_A(0) - 16f_A(-1) + 5f_A(-2)] \quad (6.189)$$

$$x_A(2) = x_A(1) + \frac{T}{12}[23f_A(1) - 16f_A(0) + 5f_A(-1)] \quad (6.190)$$

It is impossible to know $f_A(-1)$ and $f_A(-2)$ without knowing $x(-T)$ and $x(-2T)$. Hence, two integrations are performed using a one-step method starting from the known initial point $[0, x(0)]$ to determine $x_A(1)$ and $x_A(2)$. Subsequent state estimates $x_A(3), x_A(4), \dots$ are computed from the AB-3 formula. The “weakest link in the chain” argument dictates the choice of an appropriate one-step method to initiate the numerical solution. In other words, for a third-order accurate AB-3 integrator with local truncation error $\varepsilon_T \sim O(T^4)$, a third-order accurate RK-3 integrator with comparable local truncation error $\varepsilon_T \sim O(T^4)$ is used.

In the second approach, a third-order accurate one-step method can be used to find $x_A(1)$ followed by the second-order accurate multistep AB-2 integrator to determine $x_A(2)$ before switching to AB-3 integration. The first approach is preferred since the AB-2 integrator degrades the accuracy of the numerical solution.

The difference equations for AB-2, AB-3, and so forth are higher order than the first-order differential equation of the continuous-time system given in Equation 6.175. In other words, the resulting discrete-time systems for approximating the first-order continuous-time system dynamics have two or more discrete-time states depending on the order of the AB integrator used. Later, in Chapter 8, we shall see that there is a penalty for implementing higher order (and hence more accurate) multistep integrators to simulate linear continuous-time systems. The penalty takes the form of a constraint imposed on the integration step size in order to assure a stable simulation.

6.4.2 IMPLICIT METHODS

Equations 6.180 and 6.186 through 6.188 are explicit methods since all the terms on the right-hand side have already been computed. There are, however, compelling reasons for using the derivatives

$f[t_{i+1}, x_A(i + 1)]$, $f[t_i, x_A(i)]$, $f[t_{i-1}, x_A(i - 1), \dots, f[t_{i-m+1}, x_A(i - m + 1)]$ instead of $f[t_i, x_A(i)]$, $f[t_{i-1}, x_A(i - 1)]$, $f[t_{i-2}, x_A(i - 2)]$, ..., $f[t_{i-m}, x_A(i - m)]$ (see Figure 6.16) to determine the m th order interpolating polynomial $P_m(t)$. Since our objective is to compute $x_A(i + 1)$, the eventual difference equation will be implicit, that is, $x_A(i + 1)$ will appear on both sides of Equation 6.178.

Using the implicit form in Equation 6.178 yields formulas for the Adams–Moulton implicit numerical integrators given in Equations 6.191 through 6.194.

$$\text{AM-2: } x_A(i + 1) = x_A(i) + \frac{T}{2}[f_A(i + 1) + f_A(i)] \quad (6.191)$$

$$\text{AM-3: } x_A(i + 1) = x_A(i) + \frac{T}{12}[5f_A(i + 1) + 8f_A(i) - f_A(i - 1)] \quad (6.192)$$

$$\text{AM-4: } x_A(i + 1) = x_A(i) + \frac{T}{24}[9f_A(i + 1) + 19f_A(i) - 5f_A(i - 1) + f_A(i - 2)] \quad (6.193)$$

$$\begin{aligned} \text{AM-5: } x_A(i + 1) = x_A(i) + \frac{T}{720} &[251f_A(i + 1) + 646f_A(i) - 246f_A(i - 1) \\ &+ 106f_A(i - 2) - 19f_A(i - 3)] \end{aligned} \quad (6.194)$$

Note that the AM-2 integration formula is the implicit trapezoidal integrator introduced in Section 3.4. If the system model is linear, $f_A(i + 1)$ is a linear function of $x_A(i + 1)$, and an explicit solution for $x_A(i + 1)$ in Equations 6.191 through 6.194 is possible. In general, implicit equations are solved in iterative fashion by numerical methods.

AB- m and AM- m integrators are both m th-order accurate. However, the local truncation error ε_T for the implicit AM- m integrator is less than the comparable explicit AB- m integrator (see Table 6.9). The local truncation errors cannot actually be calculated because the value of \hat{t}_i is unknown except for $t_i \leq \hat{t}_i \leq t_{i+1a}$.

Multistep methods are not well suited for adaptively changing the step size based on the estimated local truncation error. With a change in step size from $x_A(i)$ to $x_A(i + 1)$, some or all of the past values $[x_A(i), f_A(i)]$, $[x_A(i - 1), f_A(i - 1)]$, ..., $[x_A(i - m), f_A(i - m)]$ can no longer be used, defeating the essential reason for using a multistep method in the first place. The use of multistep integration methods is demonstrated in Example 6.7.

TABLE 6.9
Local Truncation Errors for AB- m , AM- m Integrators ($m = 2, 3, 4, 5$)

	Local Truncation Error, ε_T		Local Truncation Error, ε_T
AB-2	$\frac{5}{12}T^3 \frac{d^2}{dt^2} f[\hat{t}_i, x(\hat{t}_i)]$	AM-2	$-\frac{1}{12}T^3 \frac{d^2}{dt^2} f[\hat{t}_i, x(\hat{t}_i)]$
AB-3	$\frac{3}{8}T^4 \frac{d^3}{dt^3} f[\hat{t}_i, x(\hat{t}_i)]$	AM-3	$-\frac{1}{24}T^4 \frac{d^3}{dt^3} f[\hat{t}_i, x(\hat{t}_i)]$
AB-4	$\frac{251}{720}T^5 \frac{d^4}{dt^4} f[\hat{t}_i, x(\hat{t}_i)]$	AM-4	$-\frac{19}{720}T^5 \frac{d^4}{dt^4} f[\hat{t}_i, x(\hat{t}_i)]$
AB-5	$\frac{475}{1440}T^6 \frac{d^5}{dt^5} f[\hat{t}_i, x(\hat{t}_i)]$	AM-5	$-\frac{27}{1440}T^6 \frac{d^5}{dt^5} f[\hat{t}_i, x(\hat{t}_i)]$

EXAMPLE 6.7

The dynamics of a tumor growth is described by the first-order differential equation as follows (Braun 1978):

$$\frac{d}{dt} V(t) = \lambda e^{-\alpha t} V(t) \quad (6.195)$$

- a. Find the difference equations for approximate tumor growth $V_A(i)$, $i = 1, 2, 3, \dots$ using AB-2 and AM-3 integrators.

- b. Find the analytical solution $V(t)$ to Equation 6.195.

The model parameters are $\lambda = 0.2$ new cells per cell per week and $\alpha = 0.02$ per week. A tumor initially contains one thousand cells.

- c. Compare results from the exact solution and approximate solutions with a step size of $T = 0.25$ week. Plot the approximate and exact solutions on the same graph.

- a. Combining the derivative function

$$f_A(i) = f[t_i, V_A(i)] = \lambda e^{-\alpha t_i} V_A(i) \quad (6.196)$$

with the AB-2 integrator of Equation 6.180, that is,

$$V_A(i+1) = V_A(i) + \frac{T}{2} [3f_A(i) - f_A(i-1)]$$

yields the second-order difference equation

$$V_A(i+1) = V_A(i) + \frac{T}{2} [3\lambda e^{-\alpha t_i} V_A(i) - \lambda e^{-\alpha t_{i-1}} V_A(i-1)] \quad (6.197)$$

$$= \left(1 + \frac{3}{2} \lambda T e^{-\alpha i T}\right) V_A(i) - \frac{1}{2} \lambda T e^{-\alpha (i-1) T} V_A(i-1), \quad i = 1, 2, 3, \dots \quad (6.198)$$

Repeating the steps for the AM-2 integrator, Equation 6.191 yields the implicit form of the difference equation, that is,

$$V_A(i+1) = \frac{5}{12} \lambda T e^{-\alpha (i+1) T} V_A(i+1) + \left(1 + \frac{8}{2} \lambda T e^{-\alpha i T}\right) V_A(i) - \lambda T e^{-\alpha (i-1) T} V_A(i) \quad (6.199)$$

Solving for $V_A(i+1)$ in Equation 6.199 produces the explicit form,

$$V_A(i+1) = \left[\frac{1 + (2/3)\lambda T e^{-\alpha i T}}{1 - (5/12)\lambda T e^{-\alpha (i+1) T}} \right] V_A(i) - \left[\frac{(1/12)\lambda T e^{-\alpha (i-1) T}}{1 - (5/12)\lambda T e^{-\alpha (i+1) T}} \right] V_A(i-1), \quad i = 1, 2, \dots \quad (6.200)$$

Note that the discrete-time system models, Equations 6.198 and 6.200, are time-varying due to the appearance of the discrete-time variable "i" in the coefficients of $V_A(i)$ and $V_A(i-1)$. This is expected since the continuous-time model, Equation 6.195 is time-varying as a result of the $e^{-\alpha t}$ term in the coefficient of $V(t)$.

- b. The exact solution is obtained by separating the differential equation, Equation 6.195, and integrating from $t = 0$, $V = V_0$ where V_0 is the initial volume of cells.

$$\int_{v_0}^v \frac{dV}{V} = \int_0^t \lambda e^{-\alpha t} dt \quad (6.201)$$

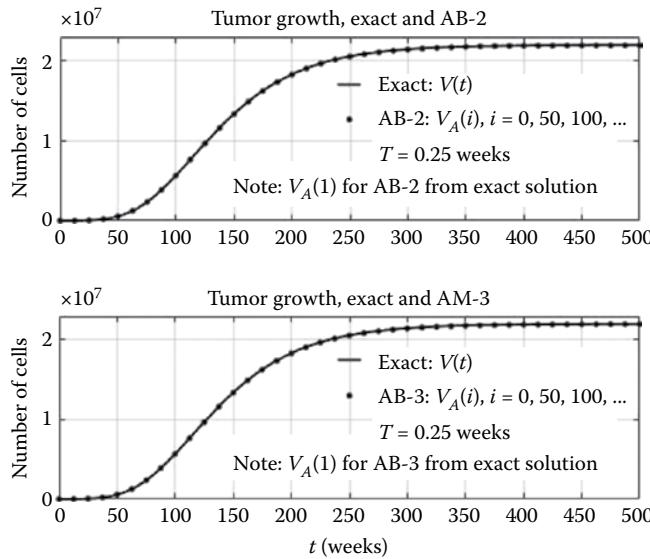


FIGURE 6.17 Tumor growth—exact solution, AB-2 and AM-3 integrators.

$$\Rightarrow V(t) = V_0 e^{(\lambda/\alpha)(1-e^{-\alpha t})} \quad (6.202)$$

- c. The AB-2 and AM-3 integrators require a single integration step using a one-step method to provide a starting value for $V_A(1)$. Ordinarily, an RK-2 one-step integrator would be used for the first step with the AB-2 and AM-3 multistep integrators. In lieu of that, we shall use the exact solution to generate $V_A(1)$ and leave the use of one-step methods to start the solution process as an exercise.

From the initial condition and Equation 6.202, the starting values for the AB-2 and AM-3 integrators are $V_A(0) = 1000$, $V_A(1) = 1051.14$.

Plots of the exact solution for tumor growth and every 50th discrete-time output of the AB-2 and AM-3 integrators are shown in Figure 6.17. Based on comparison with the exact solution of the continuous-time model, both integrators appear to predict cell growth exceptionally well (see “Ch6_Ex6_7.m”).

The limiting value of tumor size $V(\infty)$ occurs when the growth rate $(1/V)(dV/dt)$ approaches zero. This limit cannot be obtained from the continuous-time model by setting the derivative to zero as in the case of logistic growth (see Section 1.5). However, it is possible to compute $V(\infty)$ as the limiting value of the exact solution, that is,

$$V(\infty) = \lim_{t \rightarrow \infty} V_0 e^{(\lambda/\alpha)(1-e^{-\alpha t})} = V_0 e^{\lambda/\alpha} \quad (6.203)$$

Substituting the value of $V_A(0)$ for V_0 along with the given values for λ and α gives $V(\infty) = 1000 e^{0.2/0.002} = 2.203 \times 10^7$ in agreement with the plots in Figure 6.17.

6.4.3 PREDICTOR-CORRECTOR METHODS

Generally speaking, implicit methods are more accurate than explicit methods of the same order. In all but the simplest cases, the solution requires an iterative root-solving scheme, which can wreak havoc on the computational efficiency of the implicit integrator. Fortunately, a solution to the problem exists, although with a slight trade-off in the number of required derivative function evaluations.

The alternative approach is to employ an explicit method to predict the new state followed by an implicit method using the predicted state on the right-hand side of the equation. This eliminates the primary obstacle of implicit methods, namely, a nonlinear algebraic equation with the unknown updated state on both sides. The combination of explicit and implicit numerical integration is called a predictor–corrector method.

If this sounds familiar, it is because we have already implemented a simple predictor–corrector method in Section 3.6, namely, the improved Euler or Heun’s method. In that case, the predictor is the first-order explicit Euler integrator, and the corrector is the second-order trapezoidal integrator. The common practice is to combine explicit Adams–Bashforth and implicit Adams–Moulton integrators of the same order. Integration formulas for several of these predictor–corrector combinations are

$$\text{AB-2 predictor: } \hat{x}_A(i+1) = x_A(i) + \frac{T}{2} \{3f[t_i, x_A(i)] - f[t_{i-1}, x_A(i-1)]\} \quad (6.204)$$

$$\text{AM-2 corrector: } x_A(i+1) = x_A(i) + \frac{T}{2} [\hat{f}_A(i+1) + f_A(i)] \quad (6.205)$$

where $\hat{f}_A(i+1) = f[t_{i+1}, \hat{x}_A(i+1)]$ is the derivative based on the predicted state $\hat{x}_A(i+1)$.

$$\text{AB-3 predictor: } \hat{x}_A(i+1) = x_A(i) + \frac{T}{12} [23f_A(i) - 16f_A(i-1) + 5f_A(i-2)] \quad (6.206)$$

$$\text{AM-3 corrector: } x_A(i+1) = x_A(i) + \frac{T}{12} [5\hat{f}_A(i+1) + 8f_A(i) - f_A(i-1)] \quad (6.207)$$

$$\text{AB-4 predictor: } \hat{x}_A(i+1) = x_A(i) + \frac{T}{24} [55f_A(i) - 59f_A(i-1) + 37f_A(i-2) - 9f_A(i-3)] \quad (6.208)$$

$$\text{AB-4 corrector: } x_A(i+1) = x_A(i) + \frac{T}{24} [9\hat{f}_A(i+1) + 19f_A(i) + 5f_A(i-1) + f_A(i-2)] \quad (6.209)$$

It should be noted that some authors refer to the implicit numerical integrators in Equations 6.191 through 6.194 as Adams integrators and the predictor–corrector formulas in Equations 6.204 through 6.209 as Adams–Moulton integration formulas.

In certain applications, it may be desirable to execute several iterations of the corrector equation before advancing to the next integration step. In other words, corrected values are continually inserted on the right-hand side of the corrector equation until some threshold or tolerance is attained, resulting in improved estimates of the new state. In general, it is inadvisable to execute the corrector equation more than once or twice due to the additional derivative function calculations required. When the corrector equation is implemented only once, predictor–corrector integration formulas are examples of a two-pass (one for the predictor and one for the corrector) approach to updating the discrete-time state. There are no implicit equations to solve.

When the order of the predictor and corrector is the same, the combined predictor–corrector integration formula is also of that order. Furthermore, the truncation errors (local and global) are the same as those of the more accurate implicit corrector (see Table 6.9). Combining same order predictor and corrector makes it possible to estimate the local truncation error after each step (Ralston and Wilf 1965) based on the predicted and corrected states with virtually no computational overhead.

This permits the step size to be changed in an adaptive fashion. Of course, repeatedly changing the step size with a multistep integration method is counterproductive.

The stability of numerical integration methods refers to the sequence of numerical values computed for the discrete-time states when simulating a stable continuous-time system. We shall learn in [Chapter 8](#) that explicit multistep methods exhibit poorer stability characteristics compared with implicit methods. Suffice it to say for now that the higher-order AB multistep integrators are prone to instability. This is mitigated to some extent by the choice of step size. However, reducing the step size to combat the problem adversely impacts computational efficiency reflected in the total number of derivative function evaluations required to simulate the system.

EXAMPLE 6.8

A manufacturer of high-end luxury automobiles has determined that the monthly demand for its cars follows an inverse price relationship, that is,

$$d(p) = a \left(\frac{1}{p} \right), \quad p > 0 \quad (6.210)$$

where

- p is the base price of a single vehicle
- d is the monthly demand
- a is a constant

The number of vehicles produced by the manufacturer is based on the fluctuating price. Suppose the monthly supply of vehicles (up to some limit) is related to price by

$$s(p) = bp^{1/2}, \quad p > 0 \quad (6.211)$$

where

- s is the monthly production
- b is another constant

Furthermore, assume the actual price is governed by supply and demand according to

$$\frac{dp}{dt} = K[d(p) - s(p)], \quad p > 0 \quad (6.212)$$

$$= K \left[a \left(\frac{1}{p} \right) - bp^{1/2} \right] \quad (6.213)$$

where K is also a constant.

Several months ago when the price was \$200,000, 16 cars were sold. The car maker would produce 25 vehicles per month if the vehicle price were \$250,000. The current price is \$180,000.

The numerical value of K is \$2000 per vehicle.

- a. Use an AB-4/AM-4 predictor–corrector with step size $T = 0.5$ month to find the response of the price. Generate the required starting values from an RK-4 integrator.
- b. Simulate the response in part (a) using RK-4 with step size sufficiently small to approximate the exact response. Graph the simulated and “exact” response.
- a. The MATLAB file to compute the RK-4 starting values and implement AB-4/AM-4 predictor–corrector integration is “*Ch6_Ex6_8.m*.” Using the classic RK-4 integrator with

TABLE 6.10

$p_A(i)$ from AB-4/AM-4 Integration Using RK-4 Starting Values with $T = 0.5$ Months and Exact Solution $p(t)$ Approximated by RK-4 with $T = 0.01$ Months

i	t_i , Months	$p_A(i)$, \$	$p(t_i)$, \$	i	t_i , Months	$p_A(i)$, \$	$p(t_i)$, \$
0	0	180,000.00	180,000.00	16	8	161,086.76	161,086.85
2	1	174,122.06	174,122.02	18	9	160,748.08	160,748.17
4	2	169,897.30	169,897.25	20	10	160,514.70	160,514.77
6	3	166,897.64	166,897.62	22	11	160,354.00	160,354.06
8	4	164,787.43	164,787.46	24	12	160,243.42	160,243.47
10	5	163,313.01	163,313.07	26	13	160,167.36	160,167.39
12	6	162,287.89	162,287.97	28	14	160,115.05	160,115.08
14	7	161,577.64	161,577.73	30	15	160,079.08	160,079.10

step size $T = 0.5$ month, the following values were obtained to start the AB-4/AM-4 predictor–corrector:

$$p_A(0) = \$180,000, \quad p_A(1) = 176,823.92$$

$$p_A(2) = \$174,122.06, \quad p_A(3) = \$171,832.02$$

- b. Basing the exact solution $p(t)$ on RK-4 with $T = 0.01$ months produced the results in [Table 6.10](#) and plotted in [Figure 6.18](#). Values from the simulated response $p_A(i)$ are also tabulated in [Table 6.10](#) and plotted in [Figure 6.18](#). According to the graphs, the transient period for the price to reach equilibrium is approximately 15 months. The equilibrium point is easily obtained from Equation 6.213, that is,

$$0 = K \left[a \left(\frac{1}{p(\infty)} \right) - bp^{1/2}(\infty) \right] \quad (6.214)$$

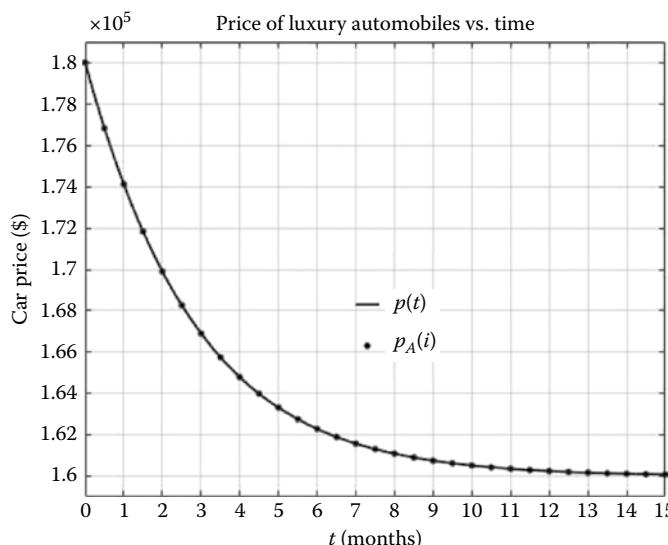


FIGURE 6.18 Price response $p_A(i)$ from AB-4/AM-4 ($T = 0.25$ months) and “Exact” $p(t)$ based on RK-4 ($T = 0.01$ months).

$$\Rightarrow p(\infty) = \left(\frac{a}{b} \right)^{2/3} = \left(\frac{3,200,000}{0.05} \right)^{2/3} = \$160,000 \quad (6.215)$$

in agreement with the value shown in Figure 6.18.

EXERCISES

- 6.15 Rework Example 6.7 using RK-2 to find the starting value $V_A(1)$ for the AB-2 and AM-3 integrators, respectively. Comment on the results.
- 6.16 Rework Example 6.7 with step size $T = 2$ weeks using an RK-4 method to find the starting values $V_A(1)$, $V_A(2)$, and $V_A(3)$ for the AB-4 and AM-4 integrators. Comment on the results.
- 6.17 Show that the equilibrium price in Example 6.8 is stable by choosing initial prices slightly less and slightly greater than $p(\infty)$ and observing the transient price responses. Use a suitable numerical integrator to obtain the transient response.
- 6.18 An unforced continuous-time system is described by the first-order differential equation $(t + 1) \frac{dx}{dt} + x = 0$. An AB-2 numerical integrator with step size T is used to simulate the response of the system with initial condition $x(0) = 1$.
- The difference equation for updating the discrete-time state $x_A(n)$ is

$$x_A(n+1) = \alpha_0 x_A(n) + \alpha_1 x_A(n-1), \quad n = 1, 2, 3, \dots$$

Express α_0 and α_1 in terms of the step size T and discrete-time variable n .

- Use an RK-2 integrator with step size $T = 0.1$ s to find $x_A(1)$, the starting value needed for the AB-2 integration.
 - Use the AB-2 integrator to find $x_A(2)$.
 - Compare the approximate values $x_A(2)$, $x_A(3)$, ..., $x_A(10)$ with the exact values $x(0.2)$, $x(0.3)$, ..., $x(1)$. Note that the exact solution is given by $x(t) = 1/(t + 1)$.
- 6.19 A double integrator is shown in Figure E6.19. Initial conditions are $x(0) = y(0) = 0$.

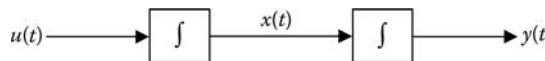


FIGURE E6.19

- Find the discrete-time system approximation (difference equation) of the first integrator using explicit Euler integration with step size T . Denote the input as $u(n)$ and the output as $x_A(n)$.
- The input is a unit step $u(t) = 1$, $t \geq 0$. Find $x_A(1)$, $x_A(2)$, and $x_A(3)$. Leave your answers in terms of T .
- Find the general solution for $x_A(n)$.
- Show that the local truncation error $(\varepsilon_T)n = x_A(n) - x(nT) = 0$, $n = 0, 1, 2, 3, \dots$. Comment on the result, that is, explain why the discrete-time output $x_A(n)$ is identical with the continuous-time output at the end of each integration step.
- Find the discrete-time system approximation (difference equation) of the second integrator using explicit Euler integration with step size T . Denote the input as $x_A(n)$ and the output as $y_A(n)$.
- Find $y_A(1)$, $y_A(2)$, $y_A(3)$, $y_A(4)$, and $y_A(5)$. Leave your answers in terms of T .
- The general solution for the output $y_A(n)$ is

$$y_A(n) = (an^2 + bn + c)T^2, \quad n = 0, 1, 2, 3, \dots$$

Find the numerical values of the constants a , b , and c .

- h. Find the differential equation relating the output $y(t)$ and input $u(t)$.
- i. Find the local truncation error $(\varepsilon_T)_n = y_A(n) - y(nT)$.

6.5 STIFF SYSTEMS

Linear time-invariant models of dynamic systems are termed “stiff” when the time constants, or more specifically the characteristic roots (eigenvalues of the system matrix A), vary significantly in magnitude. For nonlinear system models, the concept applies to the characteristic roots of a linearized model that represents the dynamics of the nonlinear system in some operating region. Linearization of nonlinear systems is discussed in [Chapter 7](#).

Systems tend to be stiff for a number of reasons. Mechanical systems composed of stiff and soft components exhibit resonant frequencies that differ greatly in magnitude. The natural response of certain electrical networks contains spikes, which die out rapidly in comparison to terms with far slower dynamics. Control system components such as controllers, actuators, and sensors oftentimes respond much quicker than the plant or process being controlled.

[Figure 6.19a and b](#) contain s -plane pole plots corresponding to stiff systems, and [Figure 6.19c](#) is a pole plot of a fast system, but not stiff. The stiffness can be quantified by the ratio of the largest (in magnitude) to the smallest characteristic root.

Stiff systems impose requirements on numerical integrators, in particular explicit methods, which can result in exceedingly small integration steps to assure the result is a stable solution. Numerical stability is considered in some detail in [Chapter 8](#). For the present time, we can think of numerical stability as a property of numerical integration, which implies that a stable discrete-time system will result whenever the continuous-time system model is stable.

Suppose the fast pole in [Figure 6.19a](#), the pole furthest from the imaginary axis, is designated s_1 and the slower poles are s_2 and s_3 , $s_4 = -\zeta\omega_n, \pm j\omega_d$. The natural response consists of a linear combination of the real modes $e^{s_1 t}$, $e^{s_2 t}$ and the oscillatory modes $e^{-\zeta\omega_n t} \sin \omega_d t$, $e^{-\zeta\omega_n t} \cos \omega_d t$. That is,

$$x_{nat}(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t} + e^{-\zeta\omega_n t} [A_1 \sin \omega_d t + A_2 \cos \omega_d t] \quad (6.216)$$

The transient response of the system with poles in [Figure 6.19a](#) is of the same form as the natural response in Equation 6.216. Due to the inherent stiffness of the system, the time constant $\tau_1 = -1/s_1$ is considerably shorter than either $\tau_2 = -1/s_2$ or $\tau = 1/\zeta\omega_n$, the effective time constant of the damped oscillations. Hence, the fast component $c_1 e^{s_1 t}$ vanishes well before the remaining terms. However, the numerical stability of fixed-step explicit integrators is controlled by the fast mode, requiring the use of a far smaller integration step than would be necessary in the absence of the fast characteristic root s_1 .

Integration formulas have been developed specifically for stiff systems. References by Gear (1971) and Hartley (1994) contain excellent descriptions of specific “stiff” integrators. MATLAB

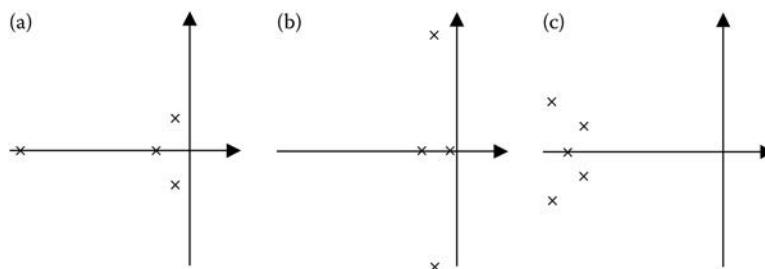


FIGURE 6.19 s -Plane location of characteristic roots for stiff system (a), (b), and nonstiff system (c).

and Simulink offer a choice of one-step and multistep integrators designed for efficient simulation of stiff systems.

6.5.1 STIFFNESS PROPERTY IN FIRST-ORDER SYSTEM

Before we illustrate an example of a stiff system, it should be mentioned that the “stiffness” property can be present in a forced system with only a single state variable, that is, a system with a single characteristic root or eigenvalue modeled by a linear first-order differential equation. The basic requirement is merely the existence of two or more terms in the response with markedly different time constants. Consider the simple forced mechanical system shown in [Figure 6.20](#).

Assuming that the mass M is negligible leads to the continuous-time model,

$$B \frac{d}{dt} x(t) + Kx(t) = F(t) \quad (6.217)$$

The state derivative function is

$$f(x) = \frac{d}{dt} x(t) = \frac{1}{\tau} \left[\frac{1}{K} F - x \right] \quad (6.218)$$

where $\tau = B/K$ is the first-order system time constant.

Suppose the forcing function $F(t)$ is an ideal step input whose amplitude is numerically equal to the spring constant K , that is, $F(t) = K$, $t \geq 0$. Because a step input is physically impossible, it is approximated by $\hat{F}(t)$

$$\hat{F}(t) = K(1 - e^{-t/\tau_F}), \quad t \geq 0 \quad (6.219)$$

where τ_F is the time constant of the exponential rise. From the system’s perspective, $\hat{F}(t)$ will look like a step input provided its rise time is several orders of magnitude less than τ , the system time constant.

Analytical solutions for the state $x(t)$ based on the ideal step input $F(t)$ of magnitude K and the approximation in Equation 6.219 are easily obtained by the use of Laplace transforms. Laplace transforming Equation 6.217 and solving for $X(s)$ give

$$X(s) = \frac{1}{\tau s + 1} \left[\frac{1}{K} F(s) \right] \quad (6.220)$$

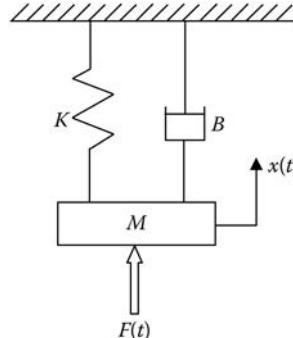


FIGURE 6.20 Simple mechanical system.

The Laplace transform of the state response to an ideal step input of magnitude K is therefore

$$X(s) = \frac{1}{\tau s + 1} \left(\frac{1}{K} \cdot \frac{K}{s} \right) = \frac{1}{\tau s + 1} \left(\frac{1}{s} \right) \quad (6.221)$$

When the input $\hat{F}(t)$ is used, $\hat{F}(s)$ replaces $F(s)$ in Equation 6.220, making the Laplace transform of the state response, denoted. $\hat{x}(t)$, equal to

$$\hat{X}(s) = \frac{1}{\tau s + 1} \left[\frac{1}{K} \cdot K \left(\frac{1}{s} - \frac{\tau_F}{\tau_F s + 1} \right) \right] = \frac{1}{\tau s + 1} \left[\frac{1}{s(\tau_F s + 1)} \right] \quad (6.222)$$

Inverse Laplace transformation of Equations 6.221 and 6.222 gives

$$x(t) = 1 - e^{-t/\tau}, \quad t \geq 0 \quad (6.223)$$

$$\hat{x}(t) = 1 - \frac{1}{\tau - \tau_F} \left(\tau e^{-t/\tau} - \tau_F e^{-t/\tau_F} \right), \quad t \geq 0 \quad (\tau \gg \tau_F) \quad (6.224)$$

Note that the response $\hat{x}(t)$ consists of a fast and a slow component, that is,

$$\hat{x}(t) = \hat{x}_F(t) + \hat{x}_S(t) \quad (6.225)$$

where

$$\hat{x}_F(t) = \frac{\tau_F}{\tau - \tau_F} e^{-t/\tau_F} \quad (6.226)$$

and

$$\hat{x}_S(t) = 1 - \frac{\tau}{\tau - \tau_F} e^{-t/\tau} \quad (6.227)$$

Simulation of the first-order system response to $\hat{F}(t)$ poses problems not previously encountered. To illustrate, consider the case when $B = 1$ and $K = 10$. The system time constant $\tau = B/K = 0.1$ s. Suppose the fast component time constant τ_F in Equation 6.219 is chosen two orders of magnitude less than the system time constant, that is, $\tau_F = \tau/100 = 0.001$ s.

Dividing $\hat{F}(t)$ by K produces the exponential rise approximation to a unit step input shown in the upper left corner of [Figure 6.21](#). The ideal unit step input and unit step response in Equation 6.223 are shown in the lower left quadrant of [Figure 6.21](#).

RK-4 integration was used to generate the simulated responses shown on the right side of [Figure 6.21](#). In the top right quadrant, the integration time step T was chosen to be an order of magnitude less than τ_F , that is, $T = \tau_F/10 = 0.0001$ s, to guarantee accuracy of the simulation. Every 150th point of the simulated response is plotted. The unit step response $x(t)$ and the simulated step response are nearly identical at $t_i = iT$, $i = 0, 1, 2, \dots$

The integration step size $T = 0.0001$ s is a great deal smaller than would seem necessary for RK-4 integration of a first-order system with time constant $\tau = 0.1$ s. Since the fast component. $\hat{x}_F(t)$ decays in $5\tau_F = 5 \times 0.001 = 0.005$ s, an adaptive procedure can be employed, which increases the step size after the transient period of the fast component has elapsed.

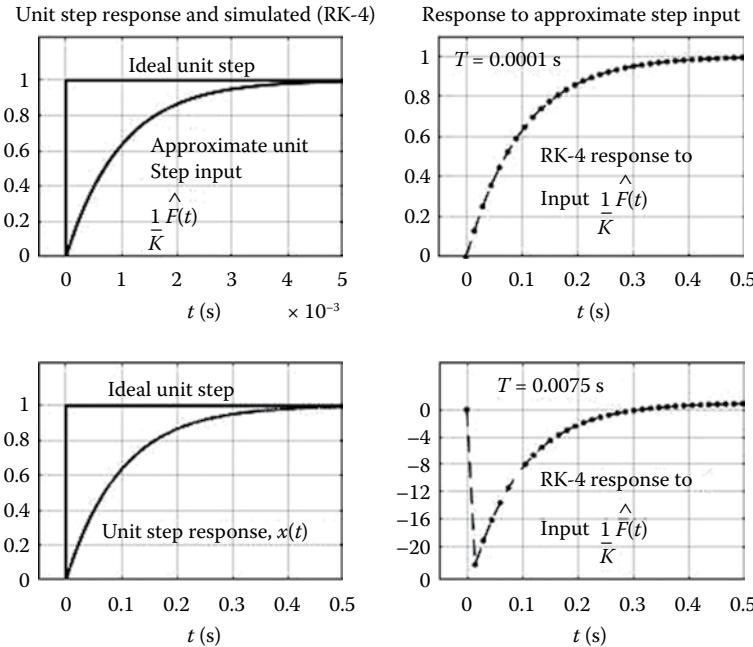


FIGURE 6.21 Unit step response and simulated (RK-4) response to input $(1/K) \hat{F}(t)$.

What do you suppose would happen if we tried a fixed-step RK-4 integrator with T , an order of magnitude smaller than the system time constant, that is, $T = \tau/10 = 0.01$ s? To answer that question, the simulation was rerun using RK-4 with T a little less than 0.01 s, namely, $T = 0.0075$ s. The simulated response (every other point) is shown in the lower right quadrant of Figure 6.21. It bears no resemblance to either $x(t)$ or $\hat{x}(t)$.

Despite the gross inaccuracy, the numerical integrator is nonetheless stable as evidenced by the limiting value approaching the correct steady-state value of unity. Further increases in T will eventually result in an unstable response of the discrete-time system. The integration step size is therefore limited by the fast time constant τ_F .

This example illustrates how a first-order system appears to be stiff, despite the fact there is only a single state. The fast input component ($\tau_F = 0.001$ s) in conjunction with the slower system natural mode ($\tau = 0.1$ s) is responsible for this happening.

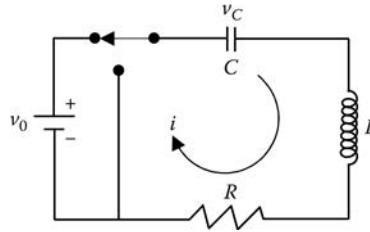
6.5.2 STIFF SECOND-ORDER SYSTEM

A second-order system is stiff if it contains a “fast” and a “slow” natural mode. Consequently, for a second-order system to be inherently stiff, it must be overdamped. The second-order circuit shown in Figure 6.22 is stiff provided the circuit parameters produce a pair of real characteristic roots several orders of magnitude apart.

Compared with fixed-step-size numerical integrators, stiff integrators increase the step size after the fast transients decay to zero, reducing execution time significantly. The following example illustrates the use of one of Simulink’s stiff integrators.

EXAMPLE 6.9

In the circuit shown in Figure 6.22, after the capacitor has fully charged to the battery voltage v_0 , the switch disconnects the battery at $t = 0$, and the capacitor discharges its stored energy to the RLC circuit. The current $i(t)$ satisfies the differential equation

**FIGURE 6.22** A second-order RLC circuit.

$$\begin{aligned} L \frac{d^2i}{dt^2} + R \frac{di}{dt} + \frac{1}{C} i &= 0 \\ v_c(0) = v_0, \quad i(0) &= 0 \end{aligned} \quad (6.228)$$

- Represent the circuit in state variable form where $x_1 = i$ and $x_2 = di/dt$.
- Show that the system is stiff when the circuit parameter values are $R = 25 \Omega$, $L = 20 \text{ mH}$, $C = 200 \text{ mF}$, and $v_0 = 12 \text{ V}$.
- Simulate the transient response using a fixed-step RK-2 integrator, and determine the largest step size T , which yields a stable and accurate solution.
- Use one of the stiff numerical integrators available in Simulink to simulate the transient response.
- Find the analytical solution for the transient response, and compare the results of parts (c) and (d) with the exact solution.

a. Derivation of the state equations is straightforward.

$$\dot{x}_1 = \frac{di}{dt} = x_2 \quad (6.229)$$

$$\dot{x}_2 = \ddot{x}_1 = \frac{d^2i}{dt^2} \quad (6.230)$$

$$= \frac{1}{L} \left[-\frac{1}{C} i - R \frac{di}{dt} \right] \quad (6.231)$$

$$= -\frac{1}{LC} x_1 - \frac{R}{L} x_2 \quad (6.232)$$

- The characteristic equation is $|sI - A| = 0$ where A is the system matrix in the state representation $\dot{\underline{x}} = A\underline{x}$. Thus,

$$|sI - A| = \left| s \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{pmatrix} \right| = 0 \quad (6.233)$$

$$s^2 + \frac{R}{L}s + \frac{1}{LC} = 0 \quad (6.234)$$

The characteristic roots are

$$s_{1,2} = \frac{-R/L \pm \sqrt{(R/L)^2 - 4(1/LC)}}{2} \quad (6.235)$$

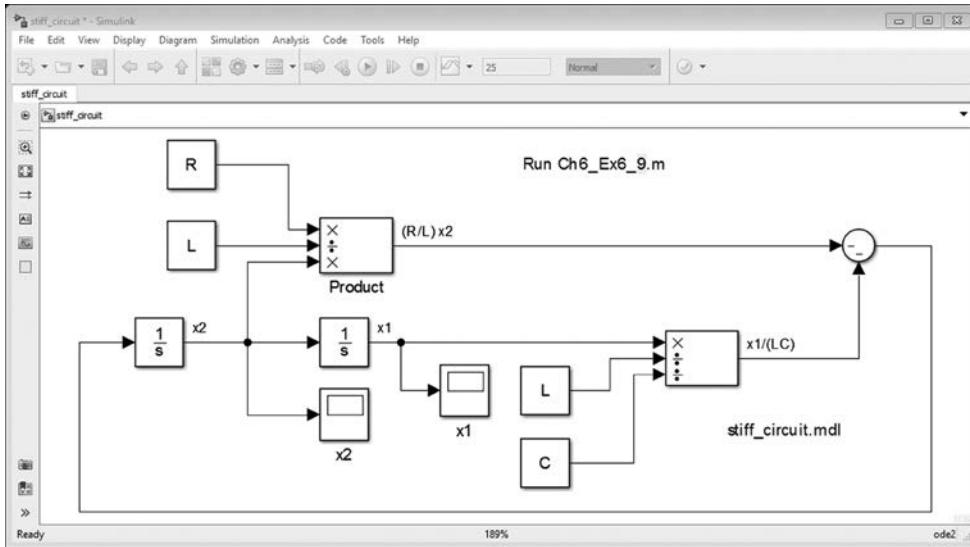


FIGURE 6.23 Simulink model for RLC circuit.

Substituting the given values for R , L , and C in Equation 6.235 yields a stiff system with characteristic roots $s_1 = -1249.8$ rad/s and $s_2 = -0.2$ rad/s.

c. The Simulink model for the system is shown in Figure 6.23.

The natural modes are $e^{s_1 t} = e^{-1249.8t}$ and $e^{s_2 t} = e^{-0.2t}$. Using Simulink's RK-2 integrator with different step sizes eventually produces a stable and accurate simulation with $T = 0.0015$ s. The discrete-time state $x_{1,A}(i)$ is plotted in the upper left graph of Figure 6.24. It requires 16,663 steps to simulate the transient response, lasting approximately 25 s. The first 41 points $x_{1,A}(i)$, $i = 0, 1, 2, \dots, 40$ are shown in Figure 6.24 and every 500th point thereafter.

Increasing the step size T from 0.0015 s to 0.0016 s with RK-2 produces the graph of $x_{1,A}(i)$ in the lower left corner of Figure 6.24. Every 500th point is plotted. While the

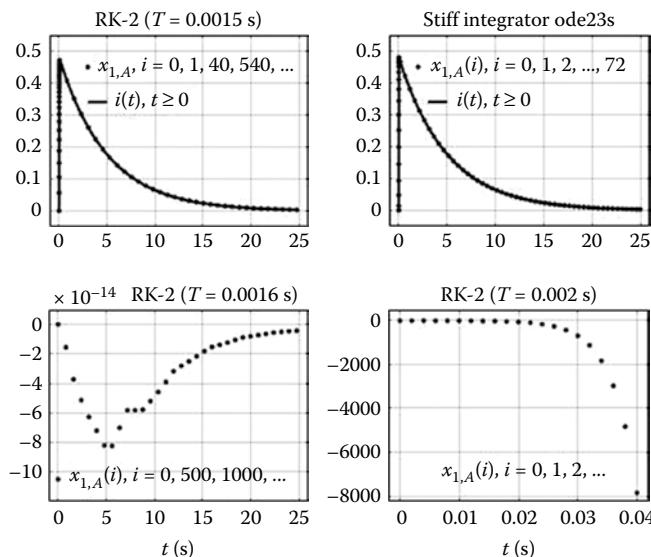


FIGURE 6.24 RK-2 integration with step sizes $T = 0.0015$, 0.0016 , 0.002 s, stiff integrator “ode23s” and exact solution for current $i(t)$.

response is stable, that is, $\lim_{i \rightarrow \infty} x_{1,A}(i) = 0$, it is clearly inaccurate. The graph in the lower right quadrant contains the first 0.04 s of the discrete-time state $x_{1,A}(i)$ when the RK-2 integration step size is 0.002 s. In this case, the discrete-time system is unstable with the simulated response becoming increasingly more negative (approaching $-\infty$) as time increases.

- d. Choosing the “ode23s” stiff integrator produces the response shown in the upper right corner of Figure 6.24. It is similar in appearance to the graph obtained with RK-2 integration and step size $T = 0.0015$ s; however, the entire simulation required a total of 72 steps. The improvement in efficiency compared to the RK-2 integrator is dramatic, that is, an average step size of $25/72 = 0.3472$ s compared to 0.0015 s.
- e. The exact solution for $i(t)$ is obtained by Laplace transformation of Equation 6.228, that is,

$$L\left[s^2I(s) - si(0) - \frac{di}{dt}(0)\right] + R[sI(s) - i(0)] + \frac{1}{C}I(s) = 0 \quad (6.236)$$

$$I(s) = \frac{di}{dt}(0) \left[\frac{1}{s^2 + (R/L)s + 1/LC} \right] \quad (6.237)$$

Replacing the initial derivative (di/dt) (0) by $v_c(0)/L$ gives

$$i(t) = \frac{v_c(0)}{L} \left[\frac{1}{s_1 - s_2} \right] (e^{s_1 t} - e^{s_2 t}) \quad (6.238)$$

where s_1 and s_2 are the characteristic roots found in Equation 6.235. Using the values for $v_c(0) = v_0$, L and the characteristic roots s_1 and s_2 , the exact solution for $i(t)$ is

$$i(t) = -0.4802(e^{-1249.8t} - e^{-0.2t}) \quad (6.239)$$

$$= -0.4802(e^{-t/0.0008} - e^{-t/5}) \quad (6.240)$$

The exact solution for $i(t)$ is plotted on the graphs with the RK-2. ($T = 0.0015$ s) and “ode23s” responses. Both are in excellent agreement with the exact solution. Note the initial spike in $i(t)$ from zero to approximately 0.48 amp. This results from the rapid decay of the fast mode $e^{-1249.8t}$ in the first $5 \times 0.0008 = 0.004$ s. After 0.004 s have elapsed, the response is essentially the slow component $0.4802e^{-0.2t}$, which lasts for approximately $5 \times 5 = 25$ s.

Stiff integrators are designed to take smaller steps while the fast component of the transient response is decaying and then accelerate after the fast component has vanished. Figure 6.25 illustrates how the integrator “ode23s” creeps along for the first 20 or so steps and then ramps up for the last 52 integration steps. Indeed, after the first 21 steps, the simulation has progressed to 0.03675 s with an average step size of 0.00175 s. The average step size over the final 51 steps is 0.4894 s.

6.5.3 APPROXIMATING STIFF SYSTEMS WITH LOWER-ORDER NONSTIFF SYSTEM MODELS

Stiff systems typically consist of components or subsystems that operate at significantly different speeds. For example, consider the control system shown in Figure 6.26 comprising a proportional controller, a second-order system, and a first-order sensor in the feedback loop. An additive disturbance or load component combines with the second-order system output to produce the complete output signal $y(t)$.

The output $Y(s)$ is expressed in terms of two transfer functions $G_R(s)$ and $G_D(s)$

$$Y(s) = G_R(s)R(s) + G_D(s)D(s) \quad (6.241)$$

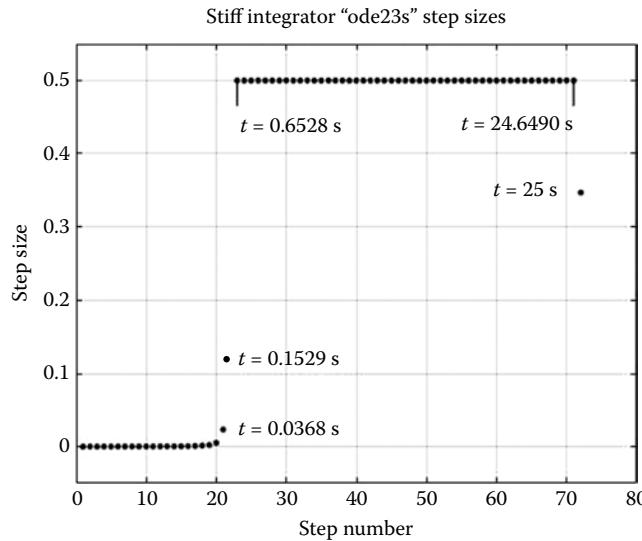


FIGURE 6.25 Step size vs. step number for “ode23s” integrator in Example 6.9.

where

$$G_R(s) = \frac{Y(s)}{R(s)} \Big|_{D(s)=0} = \frac{K_C K (\tau s + 1)}{\tau s^3 + (1 + 2\zeta\omega_n\tau)s^2 + (2\zeta\omega_n + \omega_n^2\tau)s + \omega_n^2 + K_C K K_S} \quad (6.242)$$

and

$$G_D(s) = \frac{Y(s)}{D(s)} \Big|_{R(s)=0} = \frac{K_L [\tau s^3 + (1 + 2\zeta\omega_n\tau)s^2 + (2\zeta\omega_n + \omega_n^2\tau)s + \omega_n^2]}{\tau s^3 + (1 + 2\zeta\omega_n\tau)s^2 + (2\zeta\omega_n + \omega_n^2\tau)s + \omega_n^2 + K_C K K_S} \quad (6.243)$$

The sensor dynamics are considerably faster than those of the second-order plant, a common situation in control systems. Suppose the numerical values of the system parameters are $K_C = 2$, $K = 5$, $\zeta = 0.7$, $\omega_n = 1.5$ rad/s, $K_S = 0.75$, $\tau = 0.00125$ s, and $K_L = 3$. The characteristic polynomial of the third-order system is

$$\Delta(s) = \tau s^3 + (1 + 2\zeta\omega_n\tau)s^2 + (2\zeta\omega_n + \omega_n^2\tau)s + \omega_n^2 K_C K K_S \quad (6.244)$$

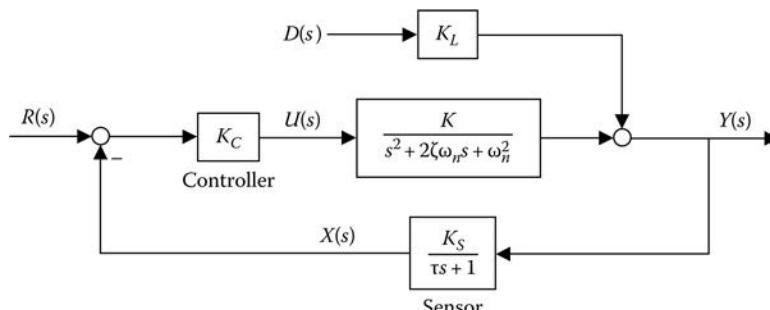


FIGURE 6.26 A stiff system with fast and slow components.

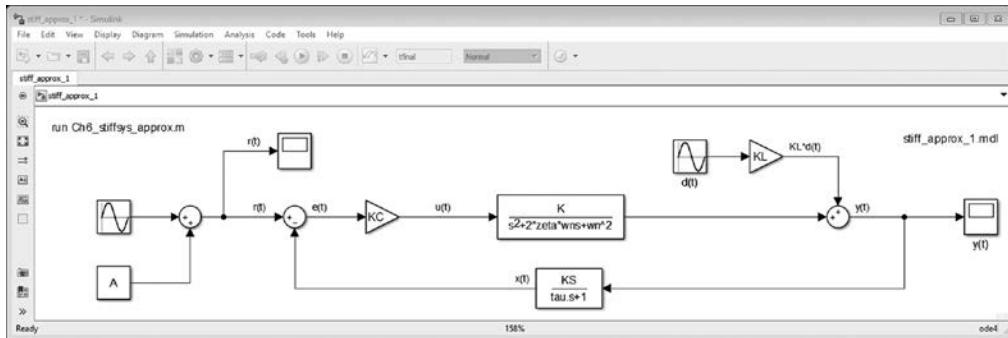


FIGURE 6.27 Simulink diagram for simulating stiff control system dynamics.

Substituting the given values of the system parameters into Equation 6.244 and using the MATLAB function “roots” to find the characteristic roots (poles) of the closed-loop control system result in $p_1 = -800.01$, $p_{2,3} = -1.0453 \pm j2.9423$. The stiffness ratio is

$$\text{stiffness} = \frac{|p_1|}{|p_2|} = \frac{|-800.01|}{|-1.0453 + j2.9423|} = 256.21 \quad (6.245)$$

indicating a moderately stiff system. A Simulink diagram of the system is shown in Figure 6.27. Both reference input and disturbance inputs are accounted for.

The simulated response to a unit step input $r(t) = 1$, $t \geq 0$ is to be obtained using RK-4 integration. Analytical methods exist to compute the largest value of step size T , which results in a stable simulation; however, they are deferred until Chapter 8. Trial and error with different values of T produced the responses shown in Figure 6.28.

The correct step response is shown on top, whereas the one on the bottom is the result of numerical instability of the RK-4 integrator at the larger step size of $T = 0.0035$ s. The results are typical of what happens when a numerical integrator becomes unstable, that is, the simulated results may be quite accurate and suddenly become useless as the integration step size is increased by a slight

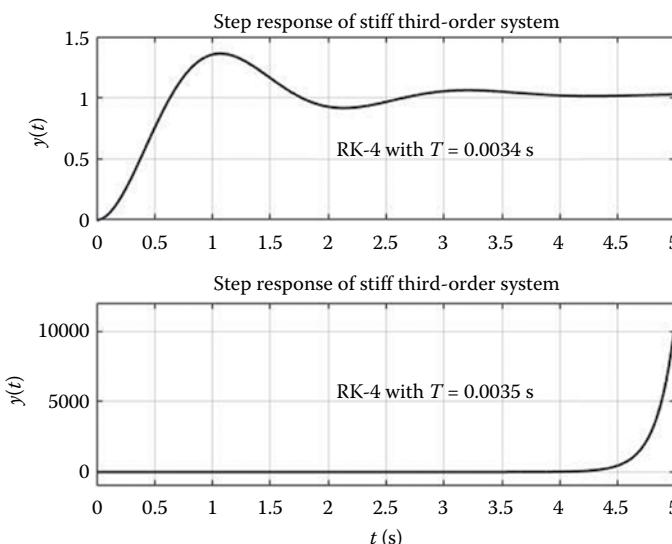


FIGURE 6.28 Stable and unstable simulated responses using RK-4 integration.

amount. Try modifying the Simulink model “*stiff_approx_1.mdl*” to allow a disturbance step input or simply make one of the initial conditions nonzero and look at the natural response. In either case, a step size of $T = 0.0034$ s produces a stable output and $T = 0.0035$ s does not.

The stiffness is attributable to the disparity in the time constant of the sensor and the effective time constant of the second-order system. The question that naturally arises is “What happens if the sensor dynamics are ignored, that is, the sensor responds instantaneously to its inputs?” The characteristic polynomial in Equation 6.244 becomes second order when the sensor time constant τ is set to zero. The control system is underdamped with a pair of complex poles, $-1.50 \pm j2.9407$, nearly identical to the complex poles of the third-order control system with sensor time constant included. The system is no longer stiff and a larger value of T can be used for RK-4 simulation.

Step responses of the original third-order control system and the reduced second-order system are generated in the MATLAB script file “*Ch6_stifffsys_approx.m*,” which calls the Simulink model “*stiff_approx_2.mdl*” shown in Figure 6.29. Both systems are simulated concurrently using RK-4 integration with step size $T = 0.001$ s.

The plant output $y(t)$ and sensor output $x(t)$ for the third-order control system with the sensor dynamics included and second-order control system with sensor approximated as a pure gain are shown in Figure 6.30. There is no noticeable difference in $y(t)$ or $x(t)$ for the second- and third-order systems.

The second-order system was simulated to determine how large the step size could be without concern for numerical instability of the RK-4 integrator. The reader should verify that step sizes up to approximately $T = 0.2$ s produce accurate (and therefore stable) results. This represents a sizable reduction in execution time, a speedup of roughly $0.2/0.0034 \approx 59$ times. Chapter 8 includes a discussion on how to find the limiting value of T precisely.

Consider the load transfer function $G_D(s)$ in Equation 6.243 for the case when $\tau = 0.00125$ s and when $\tau = 0$. Putting $G_D(s)$ in pole-zero form,

$$G_D(s) = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (6.246)$$

$$= \left(\frac{b_3}{a_3} \right) \frac{(s + z_1)(s + z_2)(s + z_3)}{(s + p_1)(s + p_2)(s + p_3)} \quad (6.247)$$

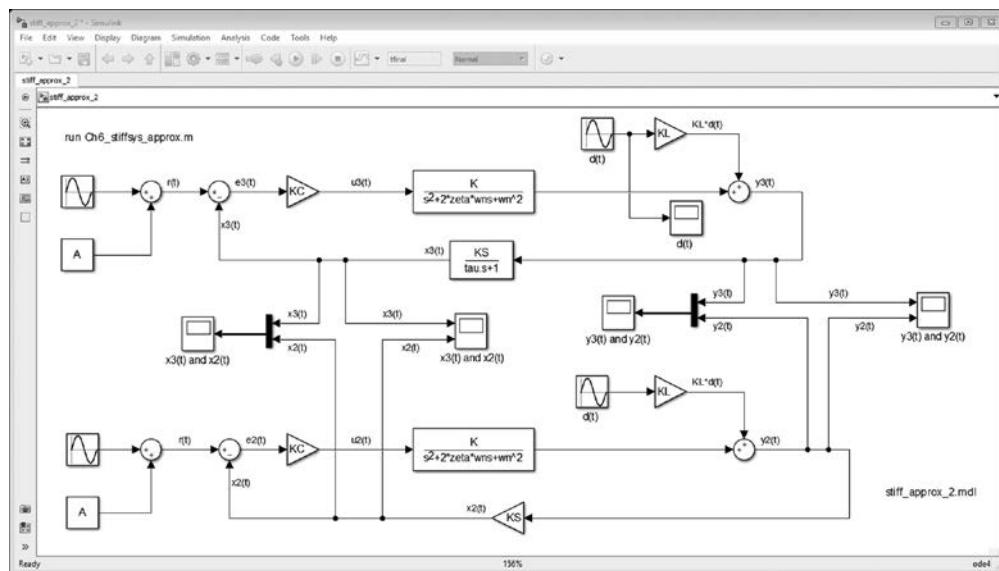


FIGURE 6.29 Simulink diagram for third-order and second-order control systems.

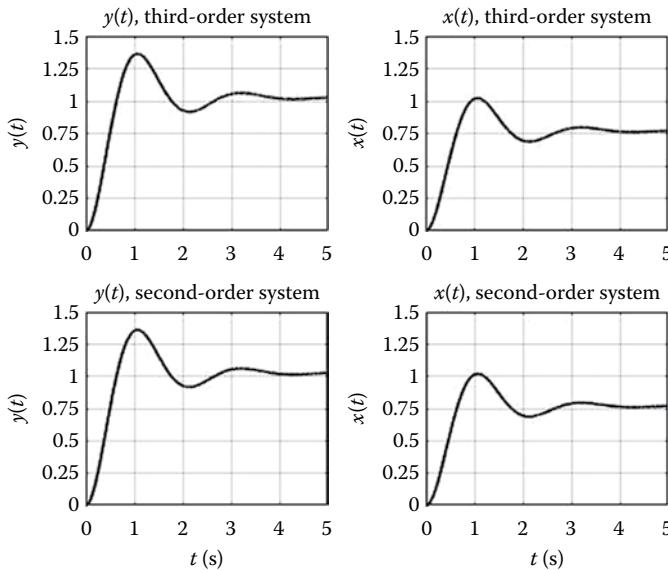


FIGURE 6.30 Step response of stiff and nonstiff control system models.

From M-file “*Ch6_stifffsys_approx.m*,” the results are $\tau = 0.00125$ s:

$b_0 = 6.75$	$a_o = 9.75$	$z_1 = -800$	$p_1 = -800.0094$
$b_1 = 6.3084$	$a_1 = 2.1028$	$z_2 = -1.05 + j1.0712$	$p_2 = -1.0453 + j2.9423$
$b_2 = 3.0079$	$a_2 = 1.0026$	$z_3 = -1.05 - j1.0712$	$p_3 = -1.0453 - j2.9423$
$b_3 = 0.0039$	$a_3 = 0.0013$		

$$G_D(s) = \frac{3(s+800)(s^2 + 2.1s + 2.25)}{(s+800.0094)(s^2 + 2.0906s + 9.7499)} \quad (6.248)$$

$\tau = 0$:

$b_0 = 6.75$	$a_0 = 9.75$	$z_1 = -1.05 + j1.0712$	$p_1 = -1.05 + j2.9407$
$b_i = 6.3$	$a_i = 2.1$	$z_2 = -1.05 - j1.0712$	$p_2 = -1.05 - j2.9407$
$b_2 = 3$	$a_2 = 1$		
$b_3 = 0$	$a_3 = 0$		

$$G_D(s) = \frac{3(s^2 + 2.1s + 2.25)}{s^2 + 2.1s + 9.75} \quad (6.249)$$

Cancelling the real pole and real zero in Equation 6.248 results in a nonstiff second-order system, which accurately represents the dynamics of the stiff third-order system.

Cancelling factors from the numerator and denominator in a transfer function when the pole-zero plot indicates that a pole and zero are close to each other is valid under most conditions. In fact, one of the goals of control system design based on “pole placement” is to mitigate or eliminate entirely the effect of undesirable modes in the open-loop system natural response. A controller with a combination zero and pole is inserted in the loop with the zero located near the undesirable open-loop pole.

Another example of approximating a stiff system model with a lower-order dynamics model is now given. In this case, the order of the approximate system is reduced by ignoring a fast mode and retaining the slower dominant mode as opposed to canceling nearly equivalent numerator and denominator factors.

EXAMPLE 6.10

An armature-controlled DC motor with a load inertia mounted on its shaft is shown in Figure 6.31. The inputs are the armature voltage $e_0(t)$ and the load torque $T_L(t)$. The outputs are the motor torque $T(t)$ and angular speed of the motor $\omega(t)$. Dependent variables (in addition to the outputs) are the armature current $i(t)$ and back emf of the motor $v_b(t)$. R and L are the electrical resistance and inductance of the armature circuit while B and J are the viscous damping coefficient and load inertia. K_b and K_T are the back emf and torque constants of the motor.

The following equations govern the dynamics of this electromechanical system:

$$e_0(t) = Ri(t) + L \frac{di}{dt} + v_b(t) \quad (6.250)$$

$$v_b(t) = K_b\omega(t) \quad (6.251)$$

$$T(t) = K_T i(t) \quad (6.252)$$

$$J \frac{d\omega}{dt} + B\omega(t) = T(t) + T_L(t) \quad (6.253)$$

- Draw a block diagram of the system and find the transfer functions $I(s)/E_0(s)$ and $\Omega(s)/E_0(s)$ where $E_0(s) = \mathcal{L}\{e_0(t)\}$, $I(s) = \mathcal{L}\{i(t)\}$, and $\Omega(s) = \mathcal{L}\{\omega(t)\}$.
- Find the steady-state gain (from armature voltage to angular speed), natural frequency, and damping ratio of the motor as a function of the motor parameters.
- Find expressions for the motor time constants in terms of the motor parameters.
- The motor constants and load inertia are

$$R = 0.2 \Omega, \quad L = 0.1 \text{ mH}, \quad K_T = 8 \times 10^{-3} \text{ ft lb}_f/\text{A}$$

$$K_b = 0.05 \frac{\text{V}}{\text{rad/s}}, \quad B = 0.01 \frac{\text{ft lb}_f}{\text{rad/s}}, \quad J = 4.5 \times 10^{-3} \frac{\text{ft lb}_f}{\text{rad/s}^2}$$

Compute the second-order system parameters, characteristic roots, time constants, and stiffness ratio.

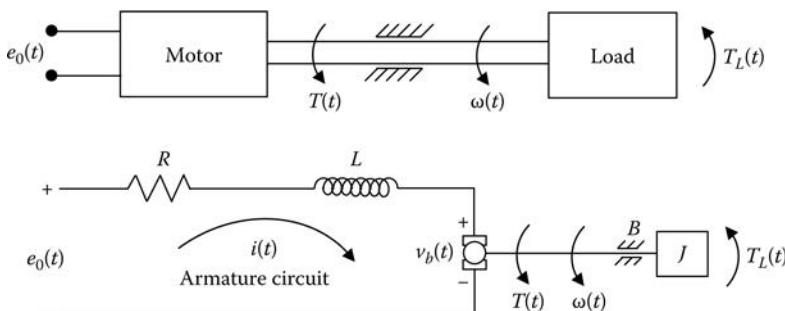


FIGURE 6.31 Armature-controlled DC motor and load.

TABLE 6.11
Motor Inductance and Load Torque
Frequency Values

$L(\text{mH})$	ω_L $2\pi \text{ rad/s}$	ω_L 200 rad/s
0		
1		
0.1		

- e. Find expressions for the time constants when the armature inductance is assumed to be negligible. Find the reduced order transfer functions $I(s)/E_0(s)$ and $\Omega(s)/E_0(s)$ when $L \approx 0$.
- f. Simulate the response $\omega(t)$, $t \geq 0$ of the first- and second-order models to a unit step input in armature voltage using Simulink's Euler integrator. Compare the results and comment on the step size required to achieve a stable response in each case.
- g. Use one of Simulink's stiff integrators to obtain the step response of the DC motor second-order system model. Compare the number of steps and execution time required for the stiff integrator and the RK-1 Euler integrator with step size $T = 0.0005 \text{ s}$.
- h. Compare the frequency response function $G_\Omega(j\omega) = \Omega(j\omega)/E_0(j\omega)$ when $L = 0.1$ and 0 mH . Comment on the results.
- i. Compare the outputs $i(t)$, $t \geq 0$ and $\omega(t)$, $t \geq 0$ in response to a load torque $T_L(t) = \sin \omega_L t$, $t \geq 0$ for the following cases shown in Table 6.11.

- a. Laplace transforming Equations 6.250 through 6.253 with initial conditions zero provides the basis for constructing the block diagram shown in Figure 6.32.

$$G_\Omega(s) = \left. \frac{\Omega(s)}{E_0(s)} \right|_{T_L(s)=0} = \frac{(1/(Ls + R))K_T(1/Js + B)}{1 + K_b(1/(Ls + R))K_T(1/Js + B)} \quad (6.254)$$

$$= \frac{K_T}{(Ls + R)(Js + B) + K_bK_T} \quad (6.255)$$

$$G_I(s) = \left. \frac{I(s)}{E_0(s)} \right|_{T_L(s)=0} = \frac{\Omega(s)}{E_0(s)} / \frac{\Omega(s)}{I(s)} \quad (6.256)$$

$$= \frac{K_T / [(Ls + R)(Js + B) + K_bK_T]}{K_T(Js + B)} \quad (6.257)$$

$$= \frac{(Js + B)}{(Ls + R)(Js + B) + K_bK_T} \quad (6.258)$$

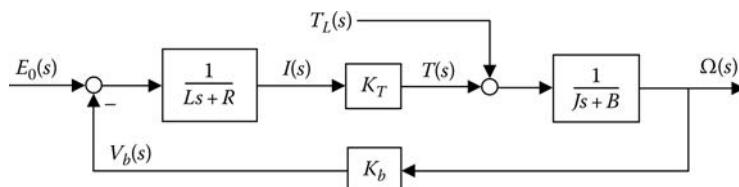


FIGURE 6.32 Block diagram of armature-controlled DC motor.

- b. Dividing $G_\Omega(s)$ in Equation 6.255 by JL and equating the result to the standard form of a second-order system,

$$G_\Omega(s) = \frac{K_T/JL}{[s + (R/L)][s + (B/J)] + K_bK_T/JL} = \frac{K_m\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (6.259)$$

Solving for the steady-state gain K_m , the natural frequency ω_n , and the damping ratio ζ in terms of the motor parameters results in

$$K_m = \frac{K_T}{BR + K_bK_T} \quad (6.260)$$

$$\omega_n = \left(\frac{BR + K_bK_T}{JL} \right)^{1/2} \quad (6.261)$$

$$\zeta = \frac{BL + JR}{2[JL(BR + K_bK_T)]^{1/2}} \quad (6.262)$$

- c. It is possible to show that the motor is overdamped ($\zeta > 1$), and, therefore, the transfer function in Equation 6.259 is expressible as

$$G_\Omega(s) = \frac{K_m\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{K_m\omega_n^2\tau_1\tau_2}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (6.263)$$

The denominator of $G_\Omega(s)$ is the characteristic polynomial $\Delta(s)$ whose roots are

$$s_1, s_2 = -\zeta\omega_n \pm \sqrt{\zeta^2 - 1}\omega_n \quad (6.264)$$

The motor time constants in Equation 6.263 are related to the characteristic roots according to $\tau_1 = -1/s_1$, $\tau_2 = -1/s_2$. Substituting Equations 6.261 and 6.262 into Equation 6.264 produces an expression for the characteristic roots,

$$s_1, s_2 = -\frac{1}{2JL}[(BL + JR) \pm \{(BL + JR)^2 - 4JL(BR + K_bK_T)\}^{1/2}] \quad (6.265)$$

Taking the negative reciprocals of s_1 and s_2 gives

$$\tau_1, \tau_2 = \frac{2JL}{(BL + JR) \pm \{(BL + JR)^2 - 4JL(BR + K_bK_T)\}^{1/2}} \quad (6.266)$$

- d. The second-order system parameters are computed using Equations 6.260 through 6.262. The results are $K_m = 3.33$ rad/s/V, $\omega_n = 73.03$ rad/s, and $\zeta = 13.71$. The characteristic polynomial of the second-order system model is

$$\Delta(s) = (Ls + R)(Js + B) + K_bK_T \quad (6.267)$$

$$= LJs^2 + (LB + RJ)s + RB + K_bK_T \quad (6.268)$$

Substituting the numerical values of the motor constants gives

$$\Delta(s) = 4.5 \times 10^{-7} s^2 + 9.0 \times 10^{-4} s + 2.4 \times 10^{-3} \quad (6.269)$$

The characteristic roots s_1 and s_2 can be found directly from Equation 6.265 or by solving for the roots of $\Delta(s)$ in Equation 6.269. The result is $s_1 = -1999.6$ rad/s and $s_2 = -2.67$ rad/s. The motor time constants are $\tau_1 = -1/s_1 = 0.0005$ s and $\tau_2 = -1/s_2 = 0.375$ s. The stiffness ratio is $s_1/s_2 = 749.7$.

- e. Ignoring terms involving L in the denominator of Equation 6.266 gives

$$\tau_1 \approx \left[\frac{2JL}{(BL + JR) \pm \{(BL + JR)^2 - 4JL(BR + K_b K_T)\}^{1/2}} \right]_{L \approx 0} = \frac{L}{R} \quad (6.270)$$

$$\tau_2 \approx \lim_{L \rightarrow 0} \left[\frac{2JL}{(BL + JR) - \{(BL + JR)^2 - 4JL(BR + K_b K_T)\}^{1/2}} \right] \quad (6.271)$$

Application of L'Hospital's rule in Equation 6.271 results in

$$\tau_2 \approx \frac{JR}{BR + K_b K_T} \quad (6.272)$$

Ignoring the effect of armature inductance, that is, assuming $L = 0$ in Equations 6.255 and 6.258, yields a first-order model of the motor with transfer functions

$$\frac{\Omega(s)}{E_0(s)} = \frac{K_T}{JR s + RB + K_b K_T} \quad (6.273)$$

$$\frac{J(s)}{E_0(s)} = \frac{Js + B}{JR s + RB + K_b K_T} \quad (6.274)$$

Hence, the motor can be modeled as a first-order component

$$\frac{\Omega(s)}{E_0(s)} = \frac{K_m}{\tau_m s + 1} \quad (6.275)$$

with time constant $\tau_m = \tau_2 = 0.375$ s and $K_m = 3.3\bar{3}$ rad/s/V.

- f. The Simulink diagram for the step responses of the first- and second-order system models using Euler integration is shown in [Figure 6.33](#).

The simulated responses of the motor to a unit step input in armature voltage occurring at $t = 0.25$ s are shown in [Figure 6.34](#). Euler integration at $T = 0.001$ s is stable for both cases, $L = 0.1$ and 0 mH. Note that both responses approach the predicted steady-state value $\omega_{ss} = K_m \times 1 = 3.3\bar{3}$ rad/s/V $\times 1V = 3.33\bar{3}$ rad/s in roughly $5 \times \tau_m = 5 \times 0.375 = 1.875$ s after the unit step is applied.

[Figure 6.35](#) shows the simulated response of the second-order system model ($L = 0.1$ mH) with Euler integration for step sizes of $T = 0.001001$ and 0.001002 s. The first plot indicates the onset of numerical instability, while the second shows clear instability at the larger step size. By trial and error, the upper limit for stable Euler integration of the stiff system model, the second-order system with $L = 0.1$ mH, is approximately $T = 0.001$ s.

The first-order system model obtained by ignoring the fast pole at $s_1 = -1999.6$ rad/s leaving only the dominant pole at $s_2 = -2.7$ rad/s can be simulated with Euler integration using a far greater integration step. [Figure 6.36](#) shows what to expect with step sizes of

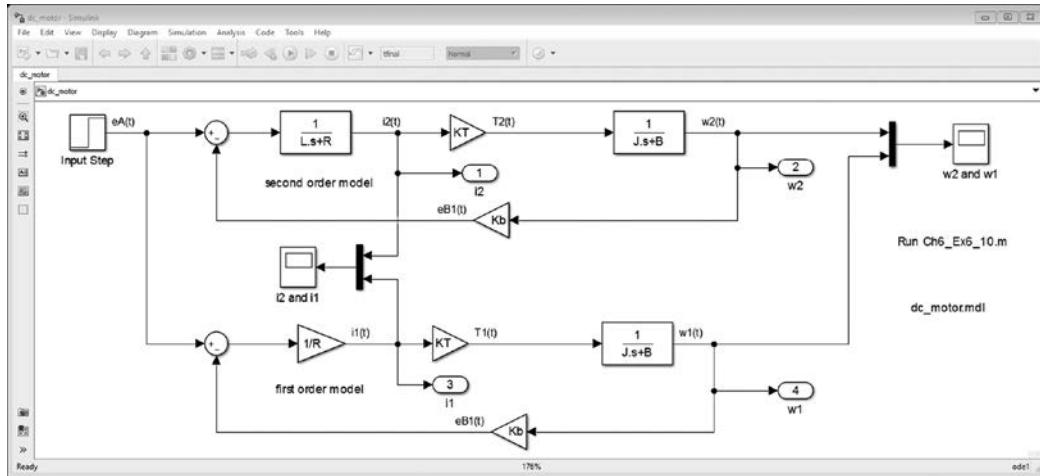


FIGURE 6.33 Simulink diagram for step responses of first- and second-order models.

$T = 0.1, 0.25, 0.75$, and 1 s , respectively. The lowest value of T results in a step response nearly identical to the analytical solution (not shown). The result is still quite acceptable for $T = 0.25\text{ s}$. The integrator appears to be marginally stable (and grossly inaccurate) when T is equal to 0.75 s . The response in the lower right is clearly unstable.

- g. The Simulink model in Figure 6.33 was called from M-file “Ch6_Ex6_10.m” with the “ode1” and “ode15s” integrators selected to simulate the motor angular speed and current. Simulated outputs of the second-order system are plotted in Figure 6.37. “ode1” is Euler and “ode15s” is one of the stiff integrators available in MATLAB.

The y-labels are written as $\omega(t)$ and $i(t)$ even though the plots are actually of the discrete-time (simulated) system outputs. The armature voltage $e_0(t)$ was applied at $t = 0.25\text{ s}$, and the simulation ran for $0.25 \pm 5\tau_m = 0.25 + 5(0.375) = 2.125\text{ s}$. The analytical solutions for $\omega(t)$ and $i(t)$ are considered in Exercise 6.24.

Euler simulation required $(0.25 + 5\tau_m)/T = 4250$ integration steps. The stiff integrator needed only 83 steps to produce comparably accurate results. The execution times for

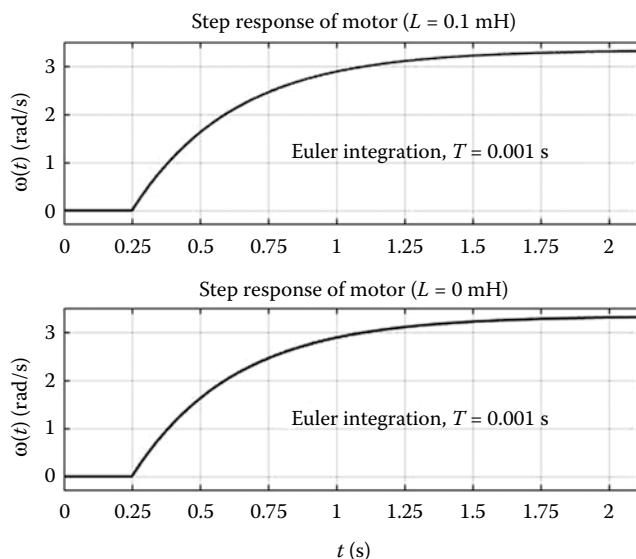


FIGURE 6.34 Unit step responses of first- and second-order system models.

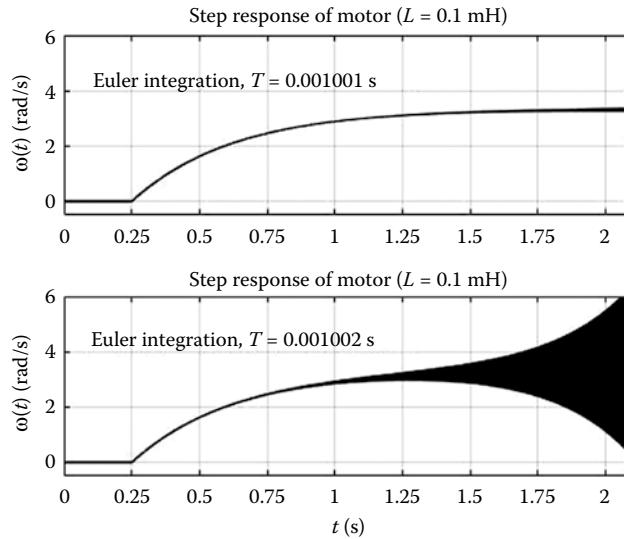


FIGURE 6.35 Unstable second-order model step responses.

each were obtained using the MATLAB command “`cputime`,” which returns the CPU time used by MATLAB from the time it is first loaded. Execution times for the Euler and stiff integrator were 63 and 47 ms, respectively.

- h. The frequency response functions $G_\Omega(j\omega)$ for the first-order model ($L = 0$) and second-order model ($L = 0.1 \text{ mH}$) are shown in Figure 6.38. The magnitude function $|G_\Omega(j\omega)|$ for $L = 0.1$ and 0 mH is nearly identical up to 1000 rad/s well beyond the cutoff frequency or bandwidth of the motor. The DC gain $G_\Omega(j0)$ is the same as the motor gain $K_m = 3.33 \text{ rad/s/V}$ (10.46 db). At $\omega = 2000 \text{ rad/s}$, the magnitudes are 0.0031 rad/s/V (-50.05 db) with $L = 0.1 \text{ mH}$ and 0.0044 rad/s/V (-47.04 db) with $L = 0$.

Figure 6.38 suggests that the dynamic response of the motor to changes in armature voltage be accurately predicted by the first-order (nonstiff) model.

- i. The six cases in Table 6.11 were simulated using the Simulink model “`dc_motor_2.mdl`” shown in Figure 6.39. “`Ch6_Ex6_10.m`” calls “`dc_motor_2.mdl`” twice, once with

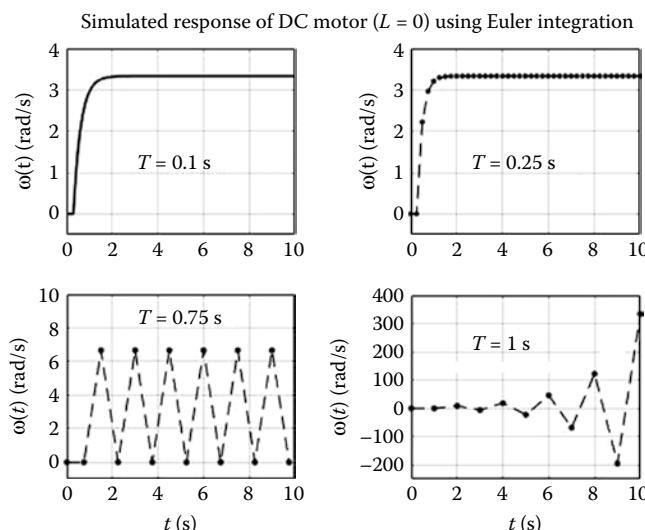


FIGURE 6.36 Simulated response using Euler integration with four different step sizes.

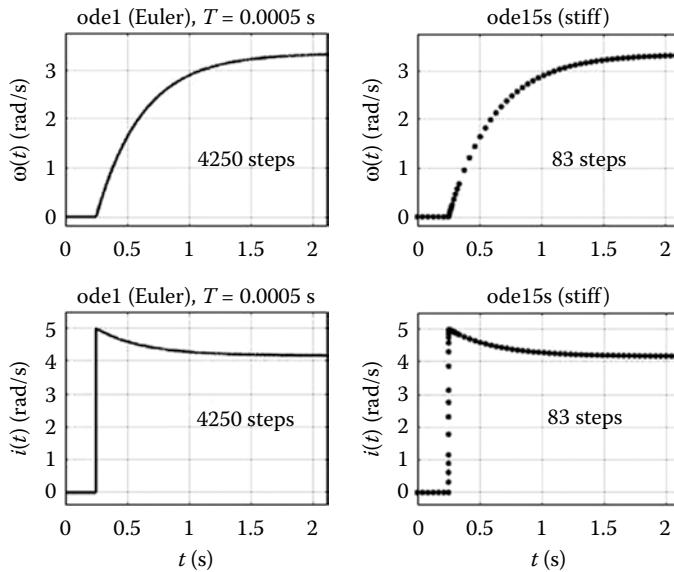


FIGURE 6.37 Simulated DC motor step response using Euler and stiff integrator.

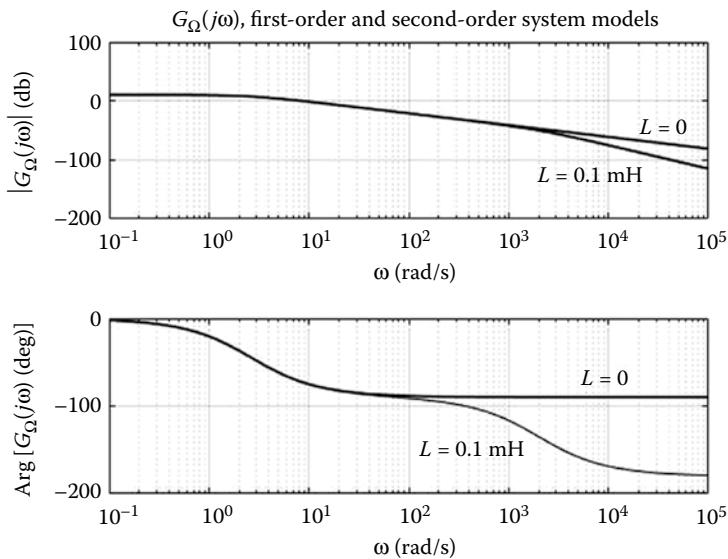


FIGURE 6.38 Frequency response function $G_\Omega(j\omega)$ for $L = 0$ and 0.1 mH .

$\omega_L = 2\pi \text{ rad/s}$ and the second time with $\omega_L = 2000 \text{ rad/s}$. The armature voltage $e_0(t)$ is zero for both calls. RK-4 integration with step size 0.0001 s was specified in "Ch6_Ex6_10.m."

The motor current and angular speeds for $L = 0, 0.1$, and 1 mH are indistinguishable from each other when the load torque frequency is $2\pi \text{ rad/s}$ (see Figure 6.40). Figure 6.41 shows angular speed and current of the motor when the load torque frequency $\omega_L = 200 \text{ rad/s}$. The angular speeds are nearly identical; however, there is a noticeable difference in current when $L = 1 \text{ mH}$. Hence, for an accurate simulation of motor current for the case when $L = 1 \text{ mH}$ and the load torque frequency is 200 rad/s (or greater), the stiff second-order system model is required.

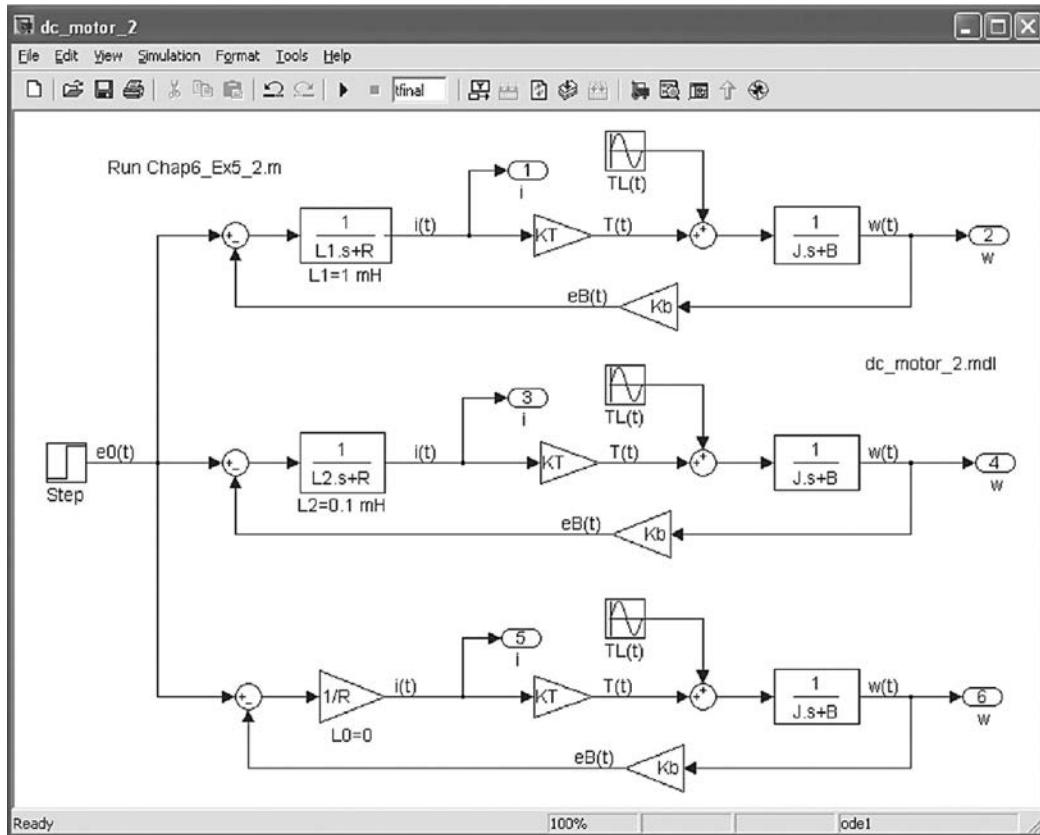


FIGURE 6.39 Simulink diagram for $i(t)$ and $\omega(t)$ with $L = 0, 0.1$, and 1 mH .

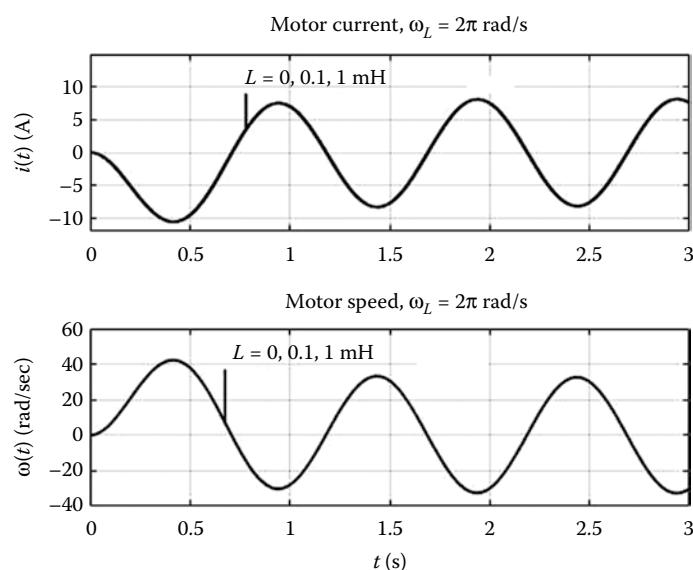


FIGURE 6.40 Motor current and speed for $L = 0, 0.1, 1 \text{ mH}$, $\omega_L = 2\pi \text{ rad/s}$.

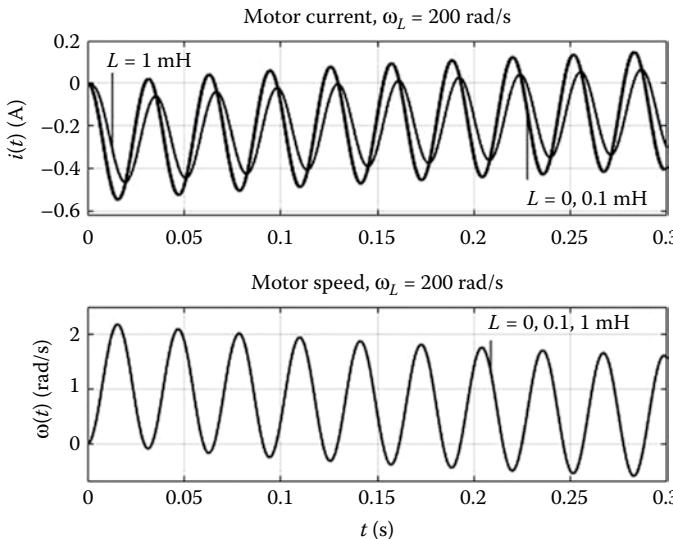


FIGURE 6.41 Motor current and speed for $L = 0, 0.1, 1 \text{ mH}$, $\omega_L = 200 \text{ rad/s}$.

EXERCISES

6.20 In Example 6.9,

- Find the largest integration time step T , which yields stable and accurate approximations of the current $i(t)$ using RK-1, RK-3, and RK-4 integrators.
- Find the analytical solution for the current $i(t)$.
- Simulate the transient response of the circuit using the remaining stiff integrators available with Simulink and compare the number of integration steps required for each one. Calculate

$$|\bar{e}| = \frac{1}{N} \sum_{k=1,2,\dots,N} |i(t_k) - x_{1,A}(t_k)|$$

where t_k , $k = 1, 2, \dots, N$ are the discrete times used by the stiff integration method to approximate the exact solution $i(t_k)$.

6.21 Figure E6.21 shows a thermal second-order system with input $u(t)$ and output $y(t)$. The temperature output is converted by a transducer, modeled as a first-order lag, to an electronic signal $v(t)$.

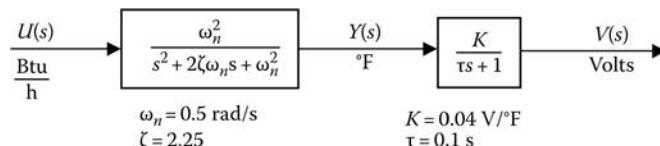
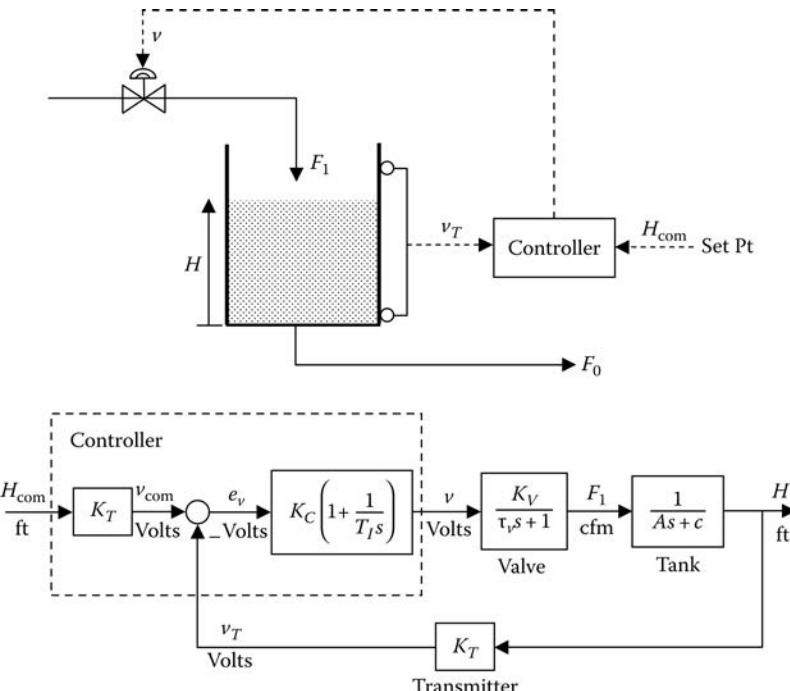


FIGURE E6.21

- Find the exact solution for the unit step response of $v(t)$.
- Find the stiffness ratio relating the ratio of the largest to the smallest (in magnitude) characteristic root of the system. Is the system stiff?
- Simulate the unit step response with a fixed-step RK integrator. What is the largest integration step size that can be used to obtain a stable solution?
- Repeat part (c) using one of Simulink's stiff integrators, and compare the number of steps used by the RK and stiff integrator.

- e. Compare the frequency response function $V(j\omega)/U(j\omega)$ with and without the sensor dynamics by generating a Bode plot for each on the same graph. Comment on the results.
- 6.22 The liquid level in the tank shown in [Figure E6.22](#) is regulated by controlling the flow in F_1 using an electronically actuated control valve. A level transmitter provides a voltage signal v_T to the controller. The set point level H_{com} is converted to a voltage v_{com} inside the controller. The actuating signal $e_v = v_{\text{com}} - v_T$ is input to the controller that outputs the voltage signal v that determines the valve opening. The valve dynamics are described by a gain K_V and time constant τ_v as shown in the block diagram of the control system. The outflow from the tank F_0 is assumed to be proportional to the level, that is, $F_0 = cH$.

**FIGURE E6.22**

- a. The characteristic equation of the closed-loop control system is

$$1 + K_T K_C \left(1 + \frac{1}{T_ls}\right) \left(\frac{K_V}{\tau_v s + 1}\right) \left(\frac{1}{As + c}\right) = 0$$

The numerical values of the system parameters are

$$\begin{aligned} K_T &= 0.25 \text{ V/ft}, K_C = 2, K_I = 10 \text{ min}, K_V = 4 \text{ cfm/V}, \tau_v = 0.01 \text{ min}, \\ A &= 100 \text{ ft}^2, c = 3 \text{ cfm/ft}. \end{aligned}$$

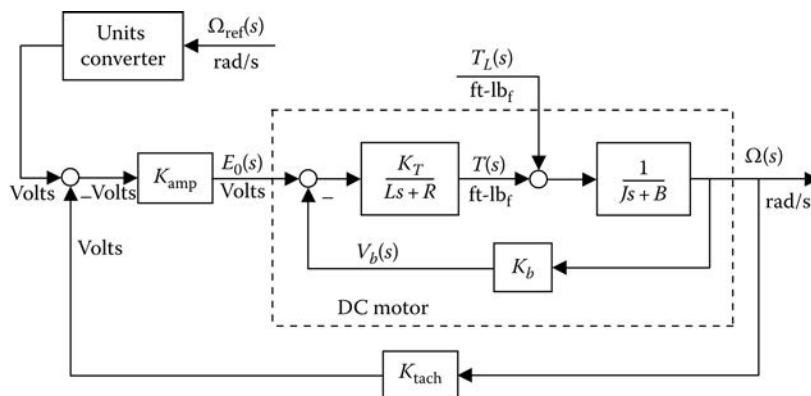
- b. Find the characteristic roots and the stiffness ratio.
- c. The system is initially at steady state with the tank empty. The set point input is a step function $H_{\text{com}}(t) = 3 \text{ ft}$, $t > 0$. The step response $H(t)$, $0 < t < 180 \text{ min}$ is simulated using Simulink's fixed-step integrators "ode1" through "ode4." Use trial and error to estimate the integration step size T (to eight places after the decimal point), resulting in a marginally stable simulated response. Enter the values in the second column of the following table.

- d. Obtain plots of $H(t)$ and $F_1(t)$ with each integrator when the step size is one half the limiting values found in part (b). Enter the number of integration steps used to simulate the tank level response in the third column of the following table.
- e. Obtain plots of $H(t)$ and $F_1(t)$ using Simulink's stiff integrator "ode23s."
- f. Find the number of integration steps in part (d) and enter the value in [Table E6.23](#).

TABLE E6.23

Integrator	T (Marginally Stable Response)	Number of Steps (Step Size $T/2$)
Ode1		
Ode2		
Ode3		
Ode4		
Ode23s	n/a	

- 6.23 Consider a third-order system with transfer function in Equation 6.248 and second-order system approximation with transfer function in Equation 6.249. Denote the transfer functions by $G_3(s)$ and $G_2(s)$, respectively. Suppose the input to both systems is $u(t) = 100e^{at}$, $t \geq 0$.
- a. Simulate the responses of each system and plot them on the same graph for the following cases:
- (i) $a = 0$
 - (ii) $a = -100$
 - (iii) $a = -800$
 - (iv) $a = -800.0094$
 - (v) $a = -5000$
- 6.24 Find analytical solutions for $\omega(t)$ and $i(t)$ in response to a step input $e_0(t)$ in Example 6.10. Compare the exact solutions with the simulated results obtained using "ode1" and "ode15s" integrators.
- 6.25 For the DC motor in Example 6.10 with armature voltage zero,
- a. Find the transfer functions
- $$G_\Omega(s)|_{E_0(s)=0} = \frac{\Omega(s)}{T_L(s)}\Big|_{E_0(s)=0}, \quad G_I(s)|_{E_0(s)=0} = \frac{I(s)}{T_L(s)}\Big|_{E_0(s)=0}$$
- b. Draw Bode plots for $G_\Omega(j\omega)|_{E_0(s)=0}$ and $G_I(j\omega)|_{E_0(s)=0}$ for $L = 0, 0.1, 1$ mH.
- c. Are the motor current and speed profiles in [Figures 6.40](#) and [6.41](#) consistent with the results in part (b)?
- 6.26 An angular speed control system is shown in [Figure E6.26a](#):

**FIGURE E6.26A**

The motor constants and load inertia are

$$R = 1 \Omega, L = 0.1 \text{ mH}, K_T = 0.8 \text{ ft lb}_f/\text{A}, K_b = 0.05 \text{ V}/\text{rad/s},$$

$$B = 0.01 \text{ ft lb}_f/\text{rad/s}, J = 0.045 \text{ ft lb}_f/\text{rad/s}^2.$$

The tachometer gain in the feedback path is $K_{\text{tach}} = 0.0475 \text{ V}/\text{rad/s}$ and the amplifier gain $K_{\text{amp}} = 50$. A units converter is inserted before the first summer to convert the reference input from rad/s to volts. The gain of the units converter is the same as K_{tach} .

- Find the stiffness of the DC motor.
- Find the stiffness of the closed-loop control system.
- Prepare a Simulink diagram for simulating the control system. The reference input and load torque profiles are shown in [Figure E6.26b](#).

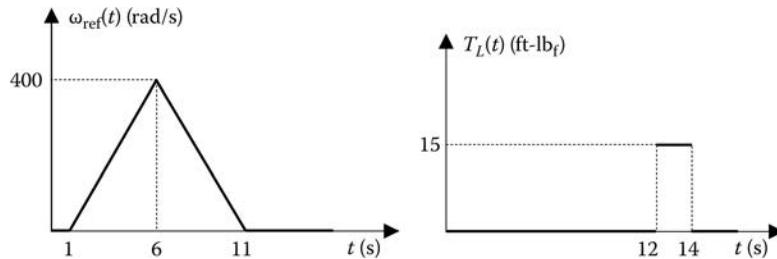


FIGURE E6.26B

- Use trial and error to find the maximum step size for stable integration of the model using RK-1 through RK-4 integration.
- Simulate the control system using RK-1 through RK-4 integration with step sizes equal to one half the values found in part (d). Repeat using the stiff integrators "ode15s," "ode23s," "ode23t," and "ode23tb." Compare the execution times and number of integration steps required with each.

6.27 The block diagram of a control system is shown in [Figure E6.27](#).

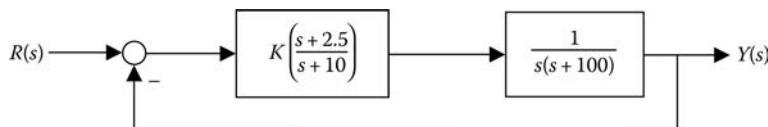


FIGURE E6.27

- Find the closed-loop system transfer function $Y(s)/R(s)$.
- Find the closed-loop system poles (characteristic roots) and the stiffness ratio for controller gains of $K = 1, 100, 1000$.
- Find the analytical solutions for the unit step responses when $K = 1, 100, 1000$.
- Select any order RK integrator and find the step size for each value of K where the integrator is on the verge of becoming unstable.
- Simulate the step responses using the selected RK integrator with a step size of one half the value found in part (d) for each value of K .
- Plot the analytical and simulated step responses on the same graphs.
- Approximate the stiff closed-loop system dynamics when $K = 100$ with a second-order transfer function obtained by ignoring the fast pole of the third-order closed-loop transfer function. Introduce a gain in the numerator of the second-order transfer function that makes the DC gain of the second- and third-order system transfer functions identical. Compare the third-order

system analytical and simulated step responses to the second-order system analytical and simulated step responses. Compare the step sizes, number of integration steps, and execution times used to simulate the original system and the reduced order system approximation.

6.6 LUMPED PARAMETER APPROXIMATION OF DISTRIBUTED PARAMETER SYSTEMS

Dynamic systems involving variables that exhibit both spatial and temporal variations are modeled by partial differential equations and referred to as distributed parameter systems. The introductory section in [Chapter 1](#) cited the example of a room temperature $T(x, y, z, t)$ that varies as a function of the point coordinates (x, y, z) as well as time t . Analytical solutions of partial differential equation models subject to various boundary conditions are rare in all but the simplest of examples. Numerical solutions are based on a partitioning of the entire volume and surface areas within the system into meshes comprising finite-sized triangular elements with interior and exterior nodes at the vertices. Difference equations, sometimes numbering in the hundreds of thousands depending on the size and shape of the finite elements, are written for the dependent variable(s) at a subset of the nodes. Accurate approximations to the continuous solutions of the partial differential equation models are possible using this “finite element analysis” approach. Examples include the temperature distribution and heat flows from irregular-shaped cooling surfaces, structural analysis, fluid dynamics, and so forth.

In dynamic systems with regular-shaped geometries, a continuously varying spatial parameter can be discretized into a finite number of values associated with discrete geometric regions. For example, consider a long, thin cylindrical rod with perfect insulation along its length and top face like the one shown in [Figure 6.42](#).

Suppose one end of the rod is immersed in a liquid bath of constant temperature \bar{T} . Assuming negligible heat flow in the x and y directions, temperature gradients exist solely in the longitudinal direction, that is, along the z -axis of the cylinder. The temperature is described by $T(t, z)$. The initial temperature distribution $T_0(z)$ is known as well.

Derivation of the equation governing the cylinder’s temperature $T(t, z)$ is straightforward (Miller 1975). The result is the partial differential equation

$$\frac{\partial}{\partial t} T(t, z) - \alpha \frac{\partial^2}{\partial z^2} T(t, z) = 0 \quad (6.276)$$

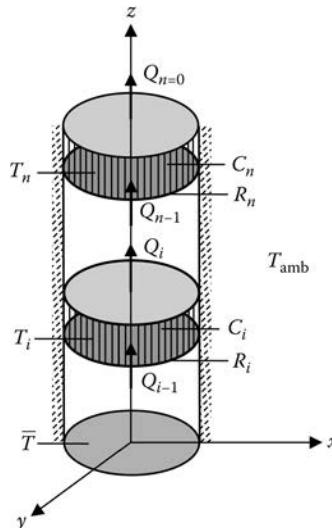


FIGURE 6.42 Lumped parameter depiction of rod with discrete thermal capacitances.

subject to initial condition $T(0,z) = T_0(z)$, $0 \leq z \leq L$ along with the boundary conditions $T(t,0) = \bar{T}$, $t \geq 0$ and $(\partial/\partial z)T(t,z)|_{z=L} = 0$, $t \geq 0$. L is the length of the cylinder, and α is a parameter related to the physical and thermal properties of the cylinder material.

A lumped parameter model consisting of coupled ordinary differential equations is obtained by dividing the cylinder into n equal segments of length $\Delta z = L/n$ (see Figure 6.42). Each segment has, associated with it, a thermal capacitance C_i and is assigned a node temperature T_i . Energy balances for each segment relate the net heat flow to the accumulation of thermal energy, that is,

$$C_i \frac{d}{dt} T_i(t) = Q_{i-1} - Q_i, \quad i = 1, 2, 3, \dots, n \quad (6.277)$$

Heat flows across the boundaries of each segment along the z -axis by conduction. Fourier's law of heat conduction states that the conductive heat flow per unit area is negatively proportional to the temperature gradient in the direction of flow. The heat flow from the constant temperature source at the bottom to the first segment with temperature T_1 is

$$Q_0 = -kA \left(\frac{T_1 - \bar{T}}{\Delta z / 2} \right) = \frac{\bar{T} - T_1}{R_l} \quad (6.278)$$

The term in parenthesis is the temperature gradient, and k is the thermal conductivity of the material. R_l represents the thermal resistance at the lower boundary and is computed from

$$R_l = \frac{\Delta z}{2kA} \quad (6.279)$$

The internal heat flows are described by

$$Q_i = -kA \left(\frac{T_{i+1} - T_i}{\Delta z} \right) = \frac{T_i - T_{i+1}}{R_{i+1}}, \quad i = 1, 2, \dots, n-1 \quad (6.280)$$

where

$$R_{i+1} = \frac{\Delta z}{2kA}, \quad i = 1, 2, \dots, n-1 \quad (6.281)$$

Heat flow between the top segment and its surroundings is zero as a result of assuming that the top face is perfectly insulated. Consequently,

$$Q_n = 0 \quad (6.282)$$

The cylindrical rod with $n = 5$ segments is illustrated in Figure 6.43.

Combining Equations 6.277, 6.278, and 6.280 through 6.282 leads to the linear system of differential equations

$$\frac{d}{dt} \begin{bmatrix} T_1(t) \\ T_2(t) \\ T_3(t) \\ T_4(t) \\ T_5(t) \end{bmatrix} = A \begin{bmatrix} T_1(t) \\ T_2(t) \\ T_3(t) \\ T_4(t) \\ T_5(t) \end{bmatrix} + B\bar{T} \quad (6.283)$$

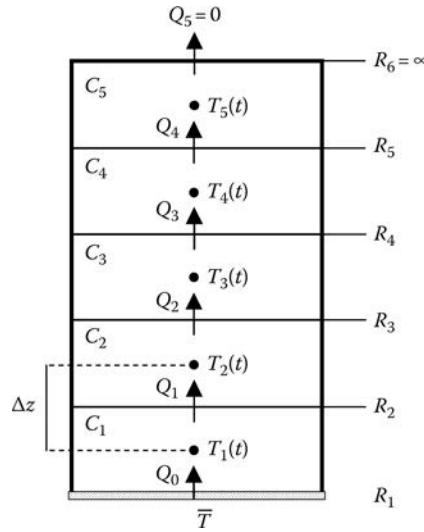


FIGURE 6.43 Cylinder with five distinct temperature nodes.

The coefficient matrix A and input matrix B are given by

$$A = \begin{bmatrix} -\left(\frac{1}{R_1} + \frac{1}{R_2}\right)\frac{1}{C_1} & \frac{1}{R_2 C_1} & 0 & 0 & 0 \\ \frac{1}{R_2 C_2} & -\left(\frac{1}{R_2} + \frac{1}{R_3}\right)\frac{1}{C_2} & \frac{1}{R_3 C_1} & 0 & 0 \\ 0 & \frac{1}{R_3 C_3} & -\left(\frac{1}{R_3} + \frac{1}{R_4}\right)\frac{1}{C_3} & \frac{1}{R_4 C_3} & 0 \\ 0 & 0 & \frac{1}{R_4 C_4} & -\left(\frac{1}{R_4} + \frac{1}{R_5}\right)\frac{1}{C_4} & \frac{1}{R_5 C_5} \\ 0 & 0 & 0 & \frac{1}{R_5 C_5} & -\frac{1}{R_5 C_5} \end{bmatrix} \quad (6.284)$$

$$B = \begin{bmatrix} \frac{1}{R_1 C_1} & 0 & 0 & 0 & 0 \end{bmatrix}^T \quad (6.285)$$

EXAMPLE 6.11

The temperature of a 10 ft long, 2 ft diameter copper cylinder is initially 75°F throughout its entire length. One of its edges is placed in contact with a surface maintained at a constant temperature of 200°F. The cylinder is thermally insulated from its surroundings except for the edge surface in contact with the 200°F temperature. Assume heat flows in the longitudinal direction only.

The physical properties of copper are:

Thermal conductivity: $k = 224 \text{ Btu/h}^\circ\text{F/ft}$

Specific heat: $c = 2.93 \text{ Btu}^\circ\text{F/slug}$

Mass density: $\rho = 17.3 \text{ slug/ft}^3$

Partition the cylinder into five equal-sized sections and

- Find the matrices A and B in the state equation $\dot{\underline{T}}(t) = A\underline{T}(t) + B\bar{T}$ where $\underline{T}(t) = [T_1(t) T_2(t) T_3(t) T_4(t) T_5(t)]^T$ is the state vector.
- Find the steady-state node temperatures.
- Simulate and plot the temperature responses of each section long enough for the transient response to die out.
- Plot the temperature profile along the bar at $t = 0, 2.5, 5, 10, 20, 30$ h.
- The volume of each section is

$$V_i = A_i \Delta z = \pi \left(\frac{D}{2}\right)^2 \Delta z = \pi \left(\frac{2}{2}\right)^2 \left(\frac{10}{5}\right) = 2\pi \text{ ft}^3, \quad i = 1, 2, \dots, 5 \quad (6.286)$$

The thermal capacitance of each section is

$$C_i = c_i \rho V_i = 2.93 \frac{\text{Btu}}{\text{°F} - \text{slug}} \times 17.3 \frac{\text{slug}}{\text{ft}^3} \times 2\pi \text{ ft}^3 = 318.49 \frac{\text{Btu}}{\text{°F}}, \quad i = 1, 2, \dots, 5 \quad (6.287)$$

and the thermal resistances at the interfaces of each section are

$$R_i = \frac{\Delta z_i}{2kA_i} = \frac{2\text{ft}}{2 \times 224 \text{ (Btu/h/°F ft)} \times \pi \text{ ft}^2} = 0.0014 \frac{\text{°F}}{\text{Btu/h}} \quad (6.288)$$

$$R_i = \frac{\Delta z_i}{kA_i} = 2 \times R_i = 0.0028 \frac{\text{°F}}{\text{Btu/h}}, \quad i = 2, 3, 4, 5 \quad (6.289)$$

Substituting the values for R_i and C_i into Equations 6.284 and 6.285 gives (see M-file "Ch6_Ex6_1.m")

$$A = \begin{bmatrix} -3.3143 & 1.1048 & 0 & 0 & 0 \\ 1.1048 & -2.2096 & 1.1048 & 0 & 0 \\ 0 & 1.1048 & -2.2096 & 1.1048 & 0 \\ 0 & 0 & 1.1048 & -2.2096 & 1.1048 \\ 0 & 0 & 0 & 1.1048 & -1.1048 \end{bmatrix}, \quad B = \begin{bmatrix} 2.2096 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- The steady-state state vector \underline{T}_{ss} is obtained from Equation 6.283 with the left-hand side equal to the zero vector. The result is

$$\underline{T}_{ss} = -A^{-1}B\bar{T} \quad (6.290)$$

$$= - \begin{bmatrix} -3.3143 & 1.1048 & 0 & 0 & 0 \\ 1.1048 & -2.2096 & 1.1048 & 0 & 0 \\ 0 & 1.1048 & -2.2096 & 1.1048 & 0 \\ 0 & 0 & 1.1048 & -2.2096 & 1.1048 \\ 0 & 0 & 0 & 1.1048 & -1.1048 \end{bmatrix}^{-1} \begin{bmatrix} 2.2096 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 200 \\ 200 \\ 200 \\ 200 \\ 200 \end{bmatrix}$$

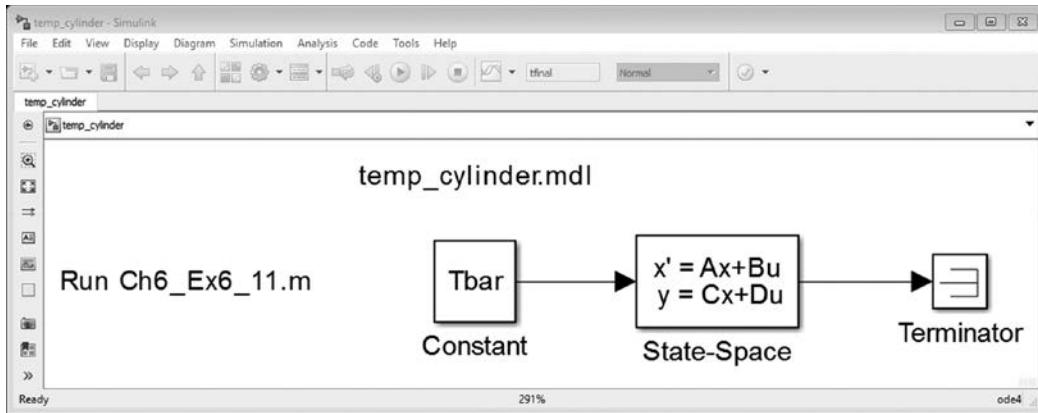


FIGURE 6.44 Simulink diagram for simulation of lumped parameter system model.

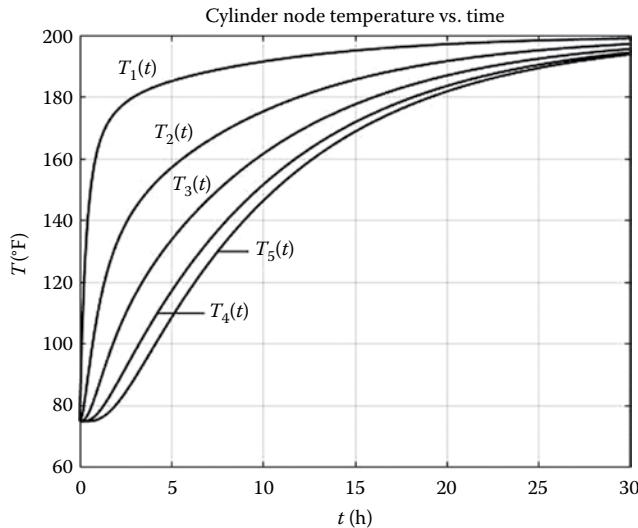


FIGURE 6.45 Time histories of cylinder node temperatures.

- c. A "Constant" block and the "state-space" block in Simulink are all that are needed to simulate the response of the lumped parameter system model. The output matrix C is chosen to be the 5×5 identity matrix, forcing the output vector to be identical to the state vector. The direct transmission matrix D is a 5×1 column vector of all zeros. The Simulink diagram is shown in Figure 6.44.

The "Workspace I/O" tab in the "Simulation Parameters" dialog box must have "Time" and "states" checked. The Simulink model file "temp_cylinder.mdl" is called from within "Ch6_Ex6_1.m." RK-4 integration with step size $T = 0.01$ h was used to generate the node temperature responses $T_1(t)$, $T_2(t)$, ..., $T_5(t)$ shown in Figure 6.45.

- d. The temperature profiles are approximated by linearly interpolating the node temperatures at the required times (see Figure 6.46).

6.6.1 NONLINEAR DISTRIBUTED PARAMETER SYSTEM

The next example illustrating the approximation of a distributed parameter system with a lumped parameter model is that of a coffee pot used for brewing coffee. In the coffee pot shown in Figure

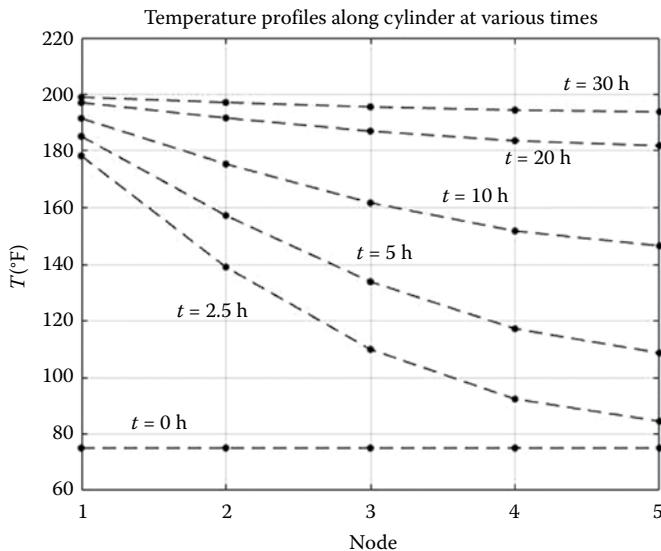


FIGURE 6.46 Temperature profiles along cylinder at $t = 0, 2.5, 5, 10, 20, 30 \text{ h}$.

6.47, liquid rises up through the riser, is distributed uniformly over the bed of coffee grounds, passes through the bed taking up coffee extract, and falls back to the bottom of the pot.

The following notation is used in the partial differential equation model, which governs the concentration of coffee in the liquid as it passes through the layer of coffee grounds, and the ordinary differential, which describes the concentration of coffee in the well-mixed reservoir at the bottom of the pot.

Notation:

A : cross-sectional area of the contact bed, ft^2

L : height of the contact bed, ft

H_L : holdup of liquid per unit height of contact bed, $\text{lb water}/\text{ft}$

H_r : holdup of liquid in reservoir of pot, lb water

a : mass transfer area per unit of volume of bed, ft^2/ft^3

k_m : mass transfer rate coefficient, $\text{lb/s coffee}/(\text{ft}^2 \times (\text{lb coffee}/\text{lb water}))$

c_s : saturated concentration of coffee, $\text{lb coffee}/\text{lb water}$

z : independent spatial variable measured from bottom to top of contact bed, ft

t : independent time variable, min

$w(t)$: circulation of liquid, $\text{lb water}/\text{s}$

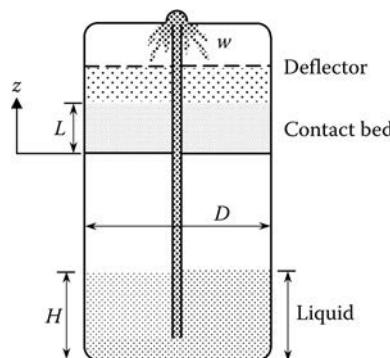


FIGURE 6.47 Coffee pot with liquid circulation.

$E_0(z, t)$: fraction of coffee not yet extracted at height z and time t

$c(z, t)$: concentration of coffee in liquid at height z and time t , lb coffee/lb water

$c_R(t)$: concentration of coffee in reservoir, lb coffee/lb water

Assuming no coffee concentration gradients in the radial direction of the contact bed and a well-mixed reservoir leads to a mathematical model based on conservation of coffee extract in the contact bed and reservoir (Huntsinger, personal notes)

$$\text{Contact bed: } -w \frac{\partial}{\partial z} c(z, t) + E_0(z, t) A a k_m [c_s - c(z, t)] = H_L \frac{\partial}{\partial z} c(z, t) \quad (6.291)$$

$$\text{subject to: } c(0, t) = c_R(t), \quad c(z, 0) = 0 \quad (6.292)$$

$$\text{Reservoir: } H_L \frac{d}{dt} c_R(t) + w c_R(t) = w c(L, t), \quad c_R(0) = 0 \quad (6.293)$$

The lumped parameter model of the coffee pot is developed in a similar manner to the way it was obtained for the temperature distribution along the cylindrical rod. That is, the contact bed is divided into a number of discrete layers with homogeneous properties throughout. The situation is illustrated in [Figure 6.48](#) for the case of three sections with uniform liquid concentrations $c_1(t)$, $c_2(t)$, and $c_3(t)$. The liquid concentration in the reservoir is $c_R(t)$.

Equations expressing the conservation of coffee extract in each homogeneous section are

$$\begin{aligned} \text{Section 1: } & w(t) c_R(t) - w(t) c_1(t) + (\Delta z) A a K_m [c_s - c_1(t)] E_{0,1}(t) \\ &= H_L \Delta z \frac{d}{dt} c_1(t) \end{aligned} \quad (6.294)$$

$$\begin{aligned} \text{Section 2: } & w(t) c_1(t) - w(t) c_2(t) + (\Delta z) A a K_m [c_s - c_2(t)] E_{0,2}(t) \\ &= H_L \Delta z \frac{d}{dt} c_2(t) \end{aligned} \quad (6.295)$$

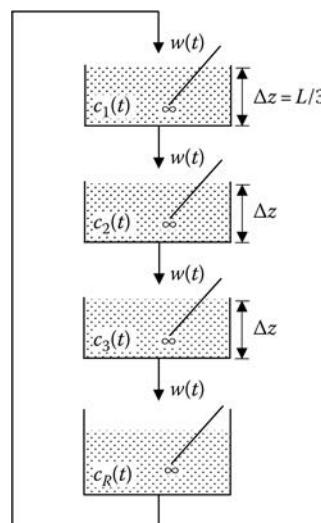


FIGURE 6.48 Lumped parameter view of coffee pot.

$$\begin{aligned} \text{Section 3: } & w(t)c_2(t) - w(t)c_3(t) + (\Delta z)AaK_m[c_s - c_3(t)]E_{0,2}(t) \\ & = H_L\Delta z \frac{d}{dt}c_3(t) \end{aligned} \quad (6.296)$$

The third term in Equations 6.294 through 6.296 accounts for the mass transfer of coffee extracted from the coffee grounds to the liquid. $E_{0,i}(t)$, $i = 1, 2, 3$ represents the fraction of coffee not yet extracted from section “ i ” after time “ t .” The equation for $E_{0,i}(t)$ is

$$E_{0,i}(t) = \frac{B_0 A \Delta z - K_{TE_i}(t)}{B_0 A \Delta z}, \quad i = 1, 2, 3 \quad (6.297)$$

where

B_0 is the total coffee per volume of bed for fresh grounds

$K_{TE_i}(t)$ is the total coffee extracted from section i in time “ t ” obtained from

$$K_{TE_i}(t) = \int_0^t E_{0,i}(\Delta z) A a K_m [c_s - c_i(\tau)] d\tau \quad (6.298)$$

The final equation of the lumped parameter model is the mass balance on the coffee in and out of the reservoir.

$$\text{Reservoir: } w(t)c_3(t) - w(t)c_R(t) = H_t \frac{d}{dt}c_R(t) \quad (6.299)$$

A careful check of all terms in Equations 6.294 through 6.296 and 6.299 will reveal the units to be lb coffee/s.

The circulation of coffee is described by

$$w(t) = \begin{cases} \left(\frac{\bar{w}}{t_1} \right) t, & 0 \leq t < t_1 \\ \bar{w}, & t \geq t_1 \end{cases} \quad (6.300)$$

The model equations are represented in the Simulink model file “coffee.mdl” shown in [Figure 6.49](#). Numerical values of the system parameters are given in “Ch6_coffee.m” and listed as follows:

$$D = 6 \text{ in.}, H = 5 \text{ in. of water}, L = 2.5 \text{ in. of coffee}$$

$$a = 3000 \text{ ft}^2 \text{ of bed}/\text{ft}^3 \text{ of bed}, \quad k_m = 0.00003 \frac{\text{lb/s coffee}}{\text{ft}^2 \times (\text{lb coffee/lb water})}$$

$$B_0 = 3 \text{ lb coffee}/\text{ft}^3 \text{ of bed}, \quad c_s = 0.2 \text{ lb coffee/lb water}$$

$$t_1 = 60 \text{ s}, \quad \bar{w} = 0.05 \text{ lb water/s}$$

Initial conditions: $c_1(0) = c_2(0) = c_3(0) = c_R(0) = 0$ lb of coffee/lb of water

Coffee concentration in the three sections and reservoir are shown in [Figure 6.50](#).

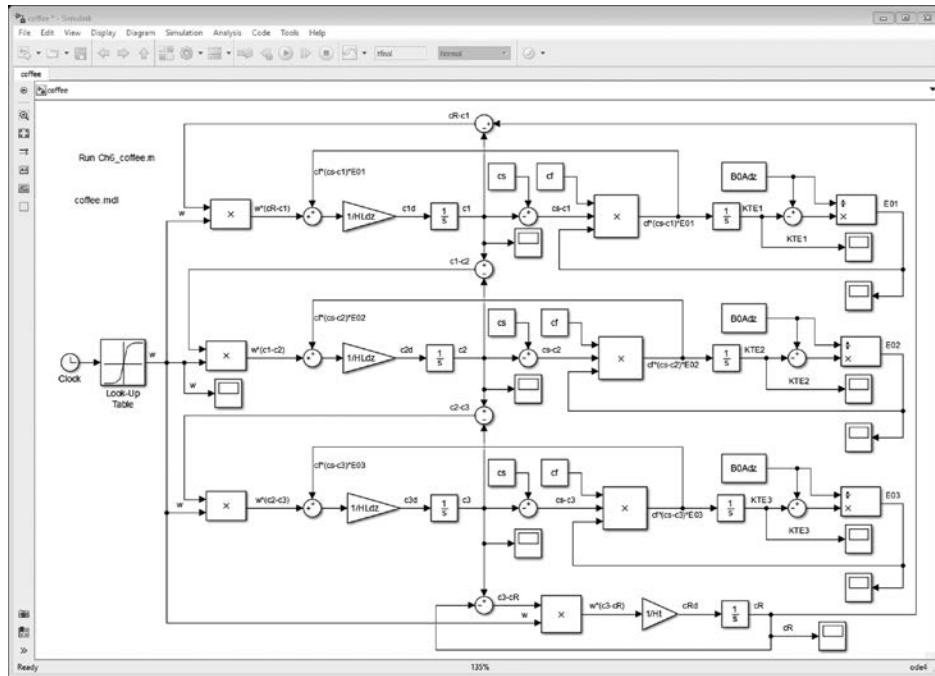


FIGURE 6.49 Simulink model for simulating coffee pot.

The transient period is approximately 10 min (600 s). The steady-state concentration attained in each section and in the reservoir is slightly greater than 0.02 lb coffee/lb of water, well below the saturation limit of $c_s = 0.2$ lb coffee/lb water.

There is no analytical method for determining $(c_1)_{ss}$, $(c_2)_{ss}$, $(c_3)_{ss}$, and $(c_R)_{ss}$ from the model Equations 6.294 through 6.296 and 6.299 when all the coffee has been extracted from the coffee grounds, that is, $E_{0,1}(\infty) = E_{0,2}(\infty) = E_{0,3}(\infty) = 0$. When this occurs, the steady-state concentrations are an identical amount that depends on the quantity of coffee grounds initially placed in the coffee pot.

Figure 6.51 shows the amount of coffee extracted (in oz) from each section and the overall amount as a function of time. The initial amount of coffee in each section (0.6545 oz) and the total

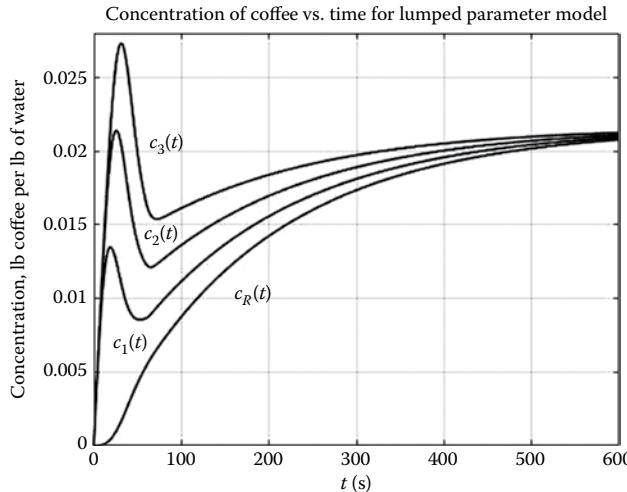


FIGURE 6.50 Concentration of coffee in lumped sections and reservoir.

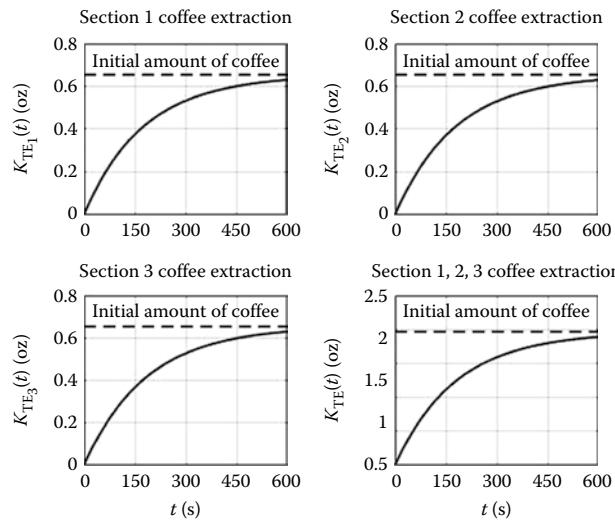


FIGURE 6.51 Coffee extraction from each section and combined.

(1.9635 oz) are calculated from the initial volumes of coffee extracted in each section and B_0 , the coffee density in lb coffee/cu ft of bed. After 10 min, the total amount of coffee extracted from sections 1, 2, and 3 is 1.8892 oz.

There is sufficient water for nearly ten 8 oz cups of coffee. Can you verify this? Figures 6.50 and 6.51 are plotted in M-file “*Ch6_coffee.m*.”

EXERCISES

- 6.28 Rework Example 6.11 for the case where the top surface of the cylinder is no longer insulated. Instead, the top surface is maintained at 0°F.
- 6.29 Rework Example 6.11 using $n = 10$ and 20 segments, and compare the results with those shown in Figures 6.45 and 6.46.
- 6.30 Rework Example 6.11 for the case where the diameter of the cylinder is 1 ft instead of 2 ft. Compare the results to those shown in Figures 6.45 and 6.46.
- 6.31 Rework Example 6.11 for the case where the bottom face of the cylinder receives a constant supply of heat in the amount of 25,000 Btu/h and the top surface is maintained at 75°F, the same as the initial temperature of the cylinder.

6.7 SYSTEMS WITH DISCONTINUITIES

Mathematical models of dynamic systems sometimes exhibit discontinuities. Internal and external forces in mechanical systems and energy sources in electrical and thermal systems can change instantaneously as a result of infinitesimal displacements in the state of these systems. Distinct regions exist in the state space where the system model is represented by different sets of algebraic and differential equations. The situation is illustrated in Figure 6.52 for the case of a discontinuous second-order system with state variables $x_1 \geq 0$, $x_2 \geq 0$ and 3 distinct regions S_1 , S_2 , and S_3 .

For a second-order system without discontinuities, a suitable mathematical model assumes the form of a system of first-order differential equations

$$\left. \begin{aligned} \frac{dx_1}{dt} &= f_1(t, x_1, x_2) \\ \frac{dx_2}{dt} &= f_2(t, x_1, x_2) \end{aligned} \right\} \quad (6.301)$$

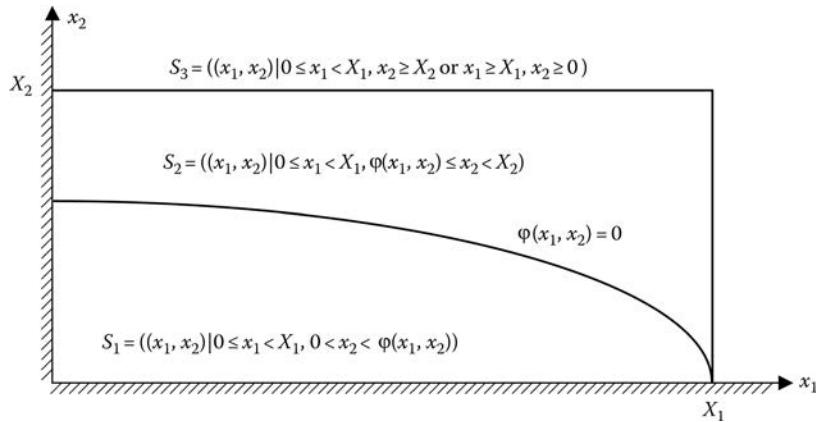


FIGURE 6.52 Discontinuous system with three distinct regions in state space.

For the second-order system with discontinuities like the one shown in Figure 6.52,

$$\left. \begin{array}{l} \frac{dx_1}{dt} = f_{11}(t, x_1, x_2), \quad (x_1, x_2) \in S_1 \\ \frac{dx_1}{dt} = f_{12}(t, x_1, x_2), \quad (x_1, x_2) \in S_2 \\ \frac{dx_1}{dt} = f_{13}(t, x_1, x_2), \quad (x_1, x_2) \in S_3 \end{array} \right\} \quad (6.302)$$

$$\left. \begin{array}{l} \frac{dx_2}{dt} = f_{21}(t, x_1, x_2), \quad (x_1, x_2) \in S_1 \\ \frac{dx_2}{dt} = f_{22}(t, x_1, x_2), \quad (x_1, x_2) \in S_2 \\ \frac{dx_2}{dt} = f_{23}(t, x_1, x_2), \quad (x_1, x_2) \in S_3 \end{array} \right\} \quad (6.303)$$

In the general case of an n th-order system with m regions S_1, S_2, \dots, S_m , we have

$$\frac{dx_i}{dt} = f_{ij}(t, x_1, x_2, \dots, x_n), \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (6.304)$$

where the m regions are defined by a set of discontinuity functions $\phi_k(x_1, x_2, \dots, x_n)$ such that a discontinuity occurs when one of the functions $\phi_k = 0$.

Simulation of a dynamic system modeled as in Equation 6.304 is not as straightforward as the systems previously encountered. The complication arises from the requirement of knowing which region the state resides in to assure numerical integration of the appropriate equations. With fixed-step as well as variable-step integration methods, the state (x_1, x_2, \dots, x_n) and the set of discontinuity functions ϕ_k are available only at discrete points in time corresponding to the end point of each integration step. The presence of a discontinuity (or several discontinuities) at an interior point of the step is sensed by a change in sign of one (or more) of the discontinuity functions.

Several approaches to the problem are possible. The simplest is to merely assume the discontinuity (or discontinuities) occurs at the end of the step in which it is detected. The appropriate model

equations are numerically integrated, starting from the beginning of the next step. The shortcoming of this approach is apparent, namely, the creation of a cumulative error resulting from integration of the incorrect equations over a portion of the interval in which the discontinuity occurs. The error is minimized by choosing excessively small integration steps when using fixed-step integrators, not a very satisfactory solution, even impossible for certain applications.

The second approach is applicable for variable-step integration methods, which adjust the step size based on estimation of the local truncation error. Instead of waiting for the end of an integration step to check for the occurrence of a discontinuity, the discontinuity functions ϕ_k are evaluated after each derivative function evaluation within the interval. A change of sign in any ϕ_k triggers a switch in one of the derivative functions, eventually producing an artificially large estimate of the truncation error. The result is a self-correcting reduction in the current integration step leading up to the time of the discontinuity and slightly beyond.

The next approach is similar to the first in that the discontinuity functions ϕ_k are evaluated only at the end of each fixed-size integration step. When one or more discontinuities are found to have occurred in the current interval, some form of interpolation or possibly root finding is employed to locate their time(s) of occurrence to a prescribed accuracy. Once the time of occurrence is determined, the integration is repeated over the subinterval ending at the time of the first (earliest) discontinuity. Subsequent integrations proceed to the end of the fixed-size integration step using the state equations appropriate to the corresponding region in state space.

The last approach is best illustrated by a simple example. [Figure 6.53](#) shows a pendulum swinging from a frictionless hinge with angular displacement confined to a single plane of motion. The bob at the end of the pendulum is immersed in a viscous fluid during a portion of its travel. The pendulum rod is assumed to be of negligible mass as is the drag force on the bob when exposed to air.

The bob is subject to a gravitational force W at all times along with a drag force F_D and buoyant force F_B acting on it while it is submerged. The forces are shown in [Figure 6.54](#) for both cases.

The pendulum dynamics are modeled by the differential equation

$$J \frac{d^2\theta}{dt^2} = \begin{cases} (-W + F_B)R \sin \theta - F_D R, & -\theta_L \leq \theta \leq \theta_L \\ -WR \sin \theta, & |\theta| > \theta_L \end{cases} \quad (6.305)$$

Expressions for the constant buoyant force and assumed linear drag force are

$$F_B = \gamma V = E \left(\frac{4}{3} \pi r^3 \right) \quad (6.306)$$

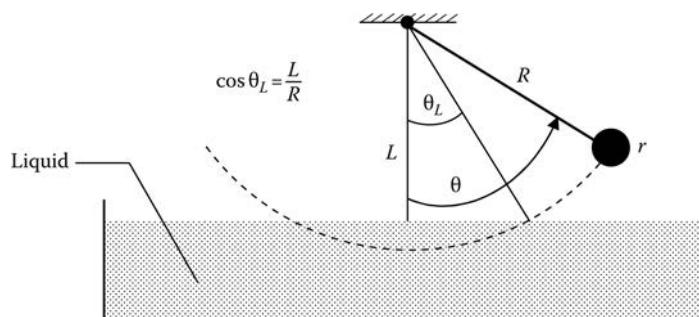


FIGURE 6.53 Pendulum traveling through air and liquid.

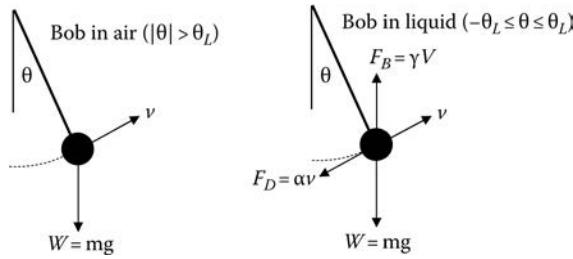


FIGURE 6.54 Diagram showing external forces acting on bob.

$$F_D = \alpha v = \alpha R \frac{d\theta}{dt} \quad (6.307)$$

where

γ is the specific weight of the liquid

V is the volume of the bob

α is the drag coefficient

Combining Equations 6.305 through 6.307 gives

$$J \frac{d^2\theta}{dt^2} = \begin{cases} \left(-mg + \frac{4}{3}\gamma\pi r^3 \right) R \sin \theta - \alpha R^2 \frac{d\theta}{dt}, & -\theta_L \leq \theta \leq \theta_L \\ -mg R \sin E, & |\theta| > \theta_L \end{cases} \quad (6.308)$$

Introducing state variables $x_1(t) = \theta(t)$ and $x_2(t) = \dot{\theta}(t)$ results in

$$\dot{x}_1 = \dot{\theta} = x_2, \quad \ddot{x}_2 = \ddot{\theta} = \begin{cases} \frac{1}{J} \left[\left(-mg + \frac{4}{3}\gamma\pi r^3 \right) R \sin x_1 - \alpha R^2 x_2 \right], & -\theta_L \leq x_1 \leq \theta_L \\ \frac{1}{J} (-mg R \sin x_1), & |x_1| > \theta_L \end{cases} \quad (6.309)$$

Defining regions S_1 and S_2 in the state space according to

$$S_1 = \{(x_1, x_2), -\theta_L \leq x_1 \leq \theta_L\} \quad \text{and} \quad S_2 = \{(x_1, x_2), |x_1| > \theta_L\} \quad (6.310)$$

and using the notation in Equation 6.304, the state derivative functions become

$$f_{11}(x_1, x_2) = x_2, \quad (x_1, x_2) \in S_1 \quad (6.311)$$

$$f_{12}(x_1, x_2) = x_2, \quad (x_1, x_2) \in S_2 \quad (6.312)$$

$$f_{21}(x_1, x_2) = \frac{1}{J} \left[\left(-mg + \frac{4}{3}\gamma\pi r^3 \right) R \sin x_1 - \alpha R^2 x_2 \right], \quad (x_1, x_2) \in S_1 \quad (6.313)$$

$$f_{22}(x_1, x_2) = \frac{1}{J} (-mg R \sin x_1), \quad (x_1, x_2) \in S_2 \quad (6.314)$$

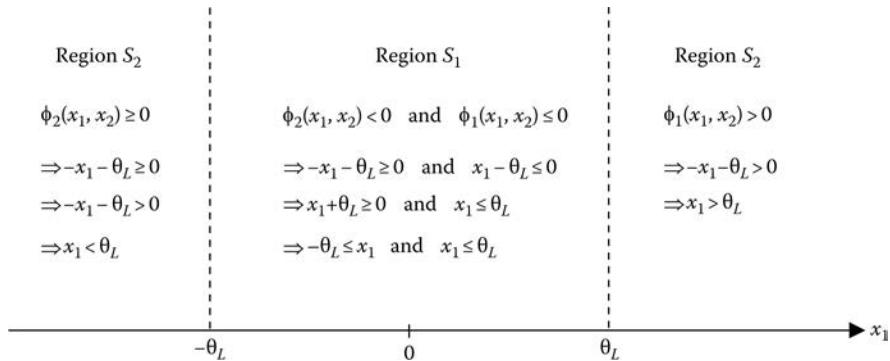


FIGURE 6.55 Definition of regions S_1 and S_2 in terms of discontinuity functions.

The discontinuity functions are

$$\phi_1(x_1, x_2) = x_1 - \theta_L \quad (6.315)$$

$$\phi_2(x_1, x_2) = -x_1 - \theta_L \quad (6.316)$$

Note that $\phi_1(x_1, x_2) = 0 \Rightarrow x_1 = \theta_L$ and $\phi_2(x_1, x_2) = 0 \Rightarrow x_1 = -\theta_L$. Hence, when either discontinuity function is zero, the pendulum is transitioning from region S_1 to S_2 or vice versa. Figure 6.55 shows the state vector (x_1, x_2) is inside region S_1 when the discontinuity functions satisfy the inequalities

$$\phi_1(x_1, x_2) \leq 0 \text{ and } \phi_2(x_1, x_2) < 0 \quad (6.317)$$

Conversely, the state vector (x_1, x_2) is in region S_2 whenever

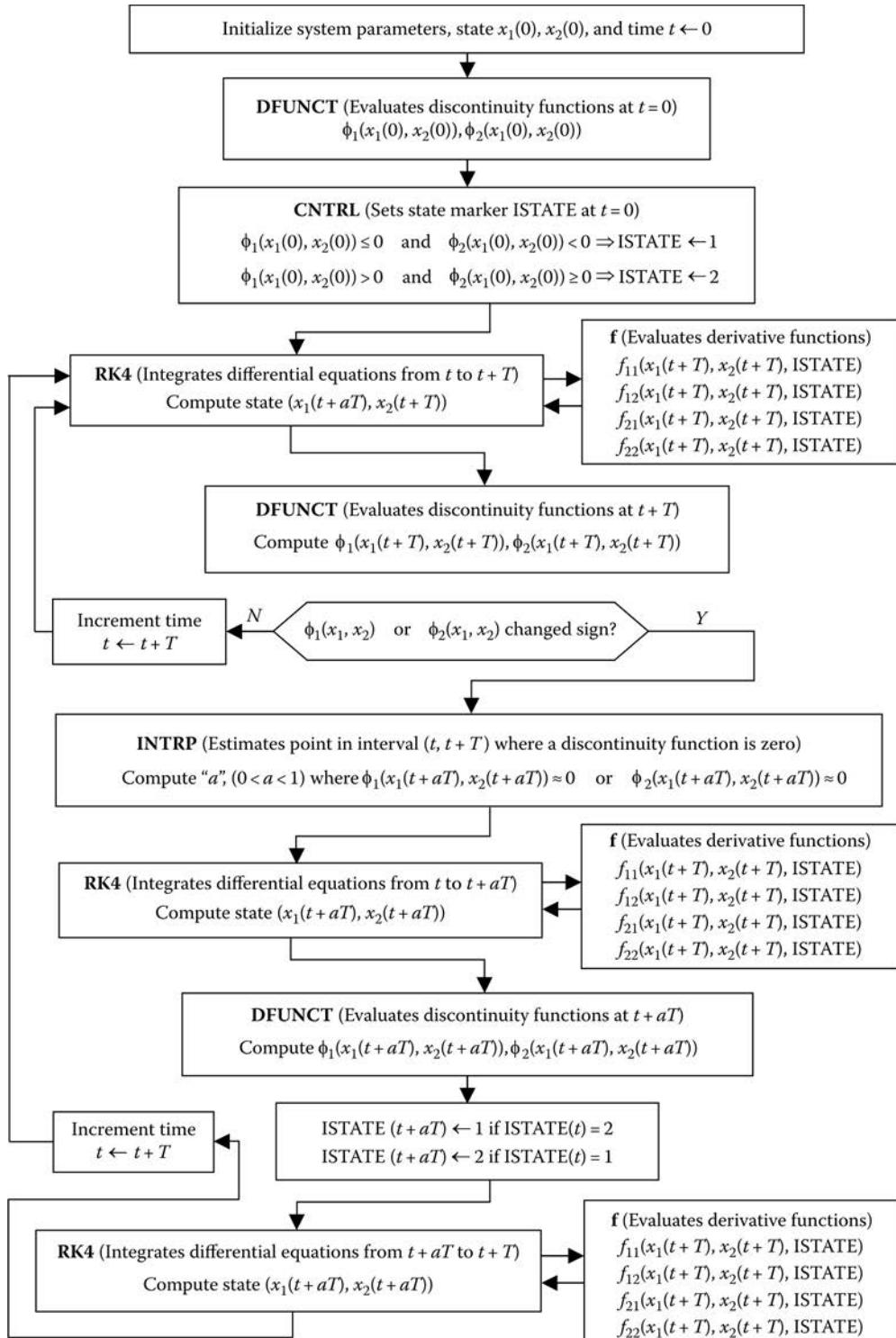
$$\phi_1(x_1, x_2) > 0 \text{ or } \phi_2(x_1, x_2) \geq 0 \quad (6.318)$$

A flow chart is shown in Figure 6.56 for simulating the pendulum dynamics. MATLAB routines called by the main program “Ch6_discont.m” are listed followed by a brief explanation of their function.

```
function [phi_1, phi_2] = DFUNCT (x1,x2)
% Evaluates discontinuity functions given state components
% Inputs: x1,x2 - components of state
% Outputs: phi1,phi2 - discontinuity functions at (x1,x2)
global thetaL
phi_1 = x1-thetaL;
phi_2 = -x1-thetaL;
```

MATLAB function “DFUNCT.m” receives the coordinates (x_1, x_2) of the state vector and returns values of the two discontinuity functions $\phi_1(x_1, x_2)$ and $\phi_2(x_1, x_2)$.

```
function ISTATE = CNTRL (phi_1,phi_2)
% Determines whether state vector is in Region S1 or S2
% S1 - pendulum bob in liquid, i.e. |x1| <= theta_L
% S2 - pendulum bob in air, i.e. |x1| > theta_L
% Inputs: phi_1, phi_2 - discontinuity functions
% Output: ISTATE - marker indicating if state is in Region S1 or S2
```

**FIGURE 6.56** Flow chart for simulation of pendulum dynamics.

```

if phi_1<=0 & phi_2<0
    ISTATE = 1; % state is in Region S1
else
    ISTATE = 2; % state is in Region S2
end

```

“CNTRL.m” accepts the values of the discontinuity functions $\phi_1(x_1, x_2)$ and $\phi_2(x_1, x_2)$ and checks which of the mutually exclusive conditions in Equation 6.317 or 6.318 are true. The marker “ISTATE” is set accordingly.

```

function [x1_new, x2_new] = RK4 (T,x1_old,x2_old, ISTATE)
% RK-4 numerical integrator for updating state
% Inputs: T - integration step size
% x1_old,x2_old - starting values of state components
% ISTATE - marker indicating if state is in Region S1 or S2
% Outputs: x1_new,x2_new - updated state vector
global g R m J gamma r alpha
[k11 k12] = f (x1_old, x2_old, ISTATE);
x1_half=x1_old + (T/2) * k11;
x2_half=x2_old + (T/2) * k12;
[k21 k22] = f (x1_half, x2_half, ISTATE);
x1_half_hat=x1_old + (T/2) * k21;
x2_half_hat=x2_old + (T/2) * k22;
[k31 k32] = f (x1_half_hat, x2_half_hat, ISTATE);
x1_full_hat=x1_old+T*k31;
x2_full_hat=x2_old+T*k32;
[k41 k42] = f(x1_full_hat, x2_full_hat, ISTATE);
x1_new=x1_old + (T/6) * (k11 + 2 * k21 + 2 * k31 + k41);
x2_new=x2_old + (T/6) * (k12 + 2 * k22 + 2 * k32 + k42);

```

“RK4.m” implements the commonly used fourth-order RK integration algorithm presented in Equations 6.60 through 6.64. In addition to inputs specifying the integration step size and the current state vector, the last input “ISTATE” is passed to the function “f.m” to assure the appropriate state derivative equations are selected, that is, Equations 6.311 and 6.313 or 6.312 and 6.314.

```

function [f1, f2] = f(x1,x2, ISTATE)
% Inputs: x1, x2 - components of state
%         ISTATE - marker indicating if state is in Region S1 or S2
% Outputs: f1, f2 - state derivatives
global g R m J gamma r alpha
f1=x2;
if ISTATE == 1
    f2 = (R/J)*(-m*g + (4/3)*(gamma*pi*r^3))*sin (x1) -alpha*x2;
elseif ISTATE == 2
    f2 = (R/J)*(-m*g*sin(x1));
end

```

“f.m” is called from “RK4.m” four times (once at the start, twice in the middle, and once at the end of the integration interval) in the process of updating the state. It returns the values of the state derivative functions.

```

function a = INTRP (ti, ph_old, ph_new, x1l, x22, k)
% Interpolates to estimate pt ti+aT where one of the discontinuity
% functions is zero. Uses linear interpolation to find intermediate
% pt ti+bT followed by quadratic interpolation based on given two pts
and intermediate pt.

```

```
% Inputs: ti - starting pt of interval to be interpolated
%         ph_old,ph_new - starting and ending value of discontinuity
%         function which changed sign over interval
%         x11, x22 - state vector at start of interval
%         k - index of discontinuity function which changed sign
% Outputs: a - decimal number between 0 and 1 indicating where
%         discontinuity function is estimated to be zero
global T ISTATE
b=ph_old/(ph_old-ph_new); % zero crossing at ti+bT based on linear
interpolation of (ti,ph_old) and (ti+T,ph_new)
t0=ti+b*T;
t1=ti;
t2=ti+T;
y1=ph_old;
y2=ph_new;
[x11 x22] = RK4(b*T, x11, x22, ISTATE);% compute state at ti+bT
[ph1 ph2] = DFUNCT (x11, x22);% compute ph1 and ph2 at ti+bT
if k == 1
    y0 = ph1;
else
    y0 = ph2;
end% if
t = [t1 t0 t2];
y = [y1 y0 y2];
p = polyfit (t, y, 2);% fit quadratic thru (ti,ph_old), (ti+T, ph_new)%
and (ti+bT, y0)
r=roots (p);% roots of quadratic
if r (1) > =ti & r(1)<= ti+T% find root in interval (ti, ti+T)
t_root = r(1);
else
    t_root = r (2);
end % if
a = (t_root-ti)/T; % normalizes "a" to between 0 and 1
```

“INTRP.m” is invoked when a change in sign of either discontinuity function is detected from one end of the integration interval to the other (see [Figure 6.56](#)). Several options are possible when it comes to estimating the point in time where the discontinuity function is zero. One approach is illustrated in [Figure 6.57](#).

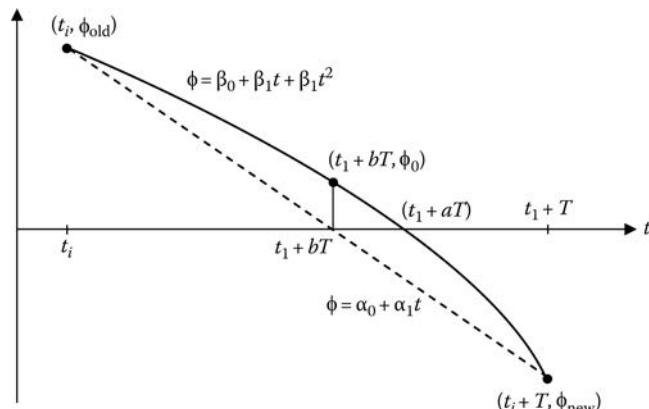


FIGURE 6.57 Quadratic interpolation to locate approximate pt of discontinuity.

The first step is to fit a linear function through the pts (t_i, ϕ_{old}) and $(t_i + T, \phi_{\text{new}})$, where t_i, ϕ_{old} and ϕ_{new} are provided as inputs to “INTRP.m.” The root of the linear function occurs at $t_i + bT$, where

$$b = \frac{\phi_{\text{old}}}{\phi_{\text{old}} - \phi_{\text{new}}} \quad (0 < b < 1) \quad (6.319)$$

The time $t_i + bT$ can be treated as the pt where the discontinuity function is approximately zero. However, an improved estimate is possible if we determine ϕ_0 , the value of the discontinuity function at $t_i + bT$, and generate the quadratic function through all three pts, namely, (t_i, ϕ_{old}) , $(t_i + T, \phi_{\text{new}})$, and $(t_i + bT, \phi_0)$. The root of the quadratic interpolation polynomial that falls between t_i and $t_i + T$ is the desired time $t_i + aT$, $(0 < a < 1)$. “INTRP.m” returns the value of “ a .”

Once the pt $t_i + aT$ is identified, RK-4 integration is repeated for the interval $(t_i, t_i + T)$ by sequentially integrating from t_i to $t_i + aT$ and then from $t_i + aT$ to $t_i + T$. Note that since the state transitions between regions at points where either discontinuity function is zero, the state marker “ISTATE” is switched from 1 to 2 or vice versa in preparation of the RK-4 integration from $t_i + aT$ to $t_i + T$ (see [Figure 6.56](#)).

An alternative to the method described involves the use of an iterative root-solving technique (e.g., Bisection, False Position, and so forth) to locate the pt $t_i + aT$. The number of iterations is controlled by setting a tolerance on the magnitude of the discontinuity function at $t_i + aT$.

A numerical example for the pendulum shown in [Figure 6.53](#) follows. Baseline system parameter values are

Radius of spherical pendulum bob: $r = 2.5$ in

Density of iron pendulum bob: $\gamma_{\text{bob}} = 491.32$ lb/ft³

Length of negligible mass pendulum rod: $R = 3$ ft

Vertical distance from center of rotation to liquid surface: $L = 2.25$ ft

Density of liquid: $\gamma = 62.4$ lb/ft³

Drag coefficient on pendulum bob in liquid: $\alpha = 0.15$ lb/ft/s

6.7.1 PHYSICAL PROPERTIES AND CONSTANT FORCES ACTING ON THE PENDULUM BOB

$$\text{Weight: } W = \gamma_{\text{iron}} V = 491.32 \frac{\text{lb}}{\text{ft}^3} \times \frac{4}{3} \pi r^3 \text{ft}^3 = 491.32 \frac{\text{lb}}{\text{ft}^3} \times \frac{4}{3} \pi \left(\frac{2.5}{12} \text{ft} \right)^3 = 18.61 \text{ lb}$$

$$\text{Mass: } m = \frac{W}{g} = \frac{18.61}{32.17} \text{ slug} = 0.5785 \text{ slug}$$

Moment of inertia about axis of rotation: $J = mR^2 = 0.5785 \text{ slug} \times (3 \text{ ft})^2 = 5.21 \text{ ft lb}_f \text{ s}^2$

$$\text{Buoyant force: } F_B = \gamma V = 62.4 \frac{\text{lb}}{\text{ft}^3} \times \frac{4}{3} \pi r^3 \text{ft}^3 = 62.4 \frac{\text{lb}}{\text{ft}^3} \times \frac{4}{3} \pi \left(\frac{2.5}{12} \text{ft} \right)^3 = 2.36 \text{ lb}$$

Angle of pendulum at initial contact with liquid:

$$\begin{aligned} \theta_L &= \cos^{-1} \left(\frac{L}{R} \right) \cos^{-1} \left(\frac{2.25}{3} \right) \\ &= 0.7227 \text{ rad (41.41}^\circ\text{)} \end{aligned}$$

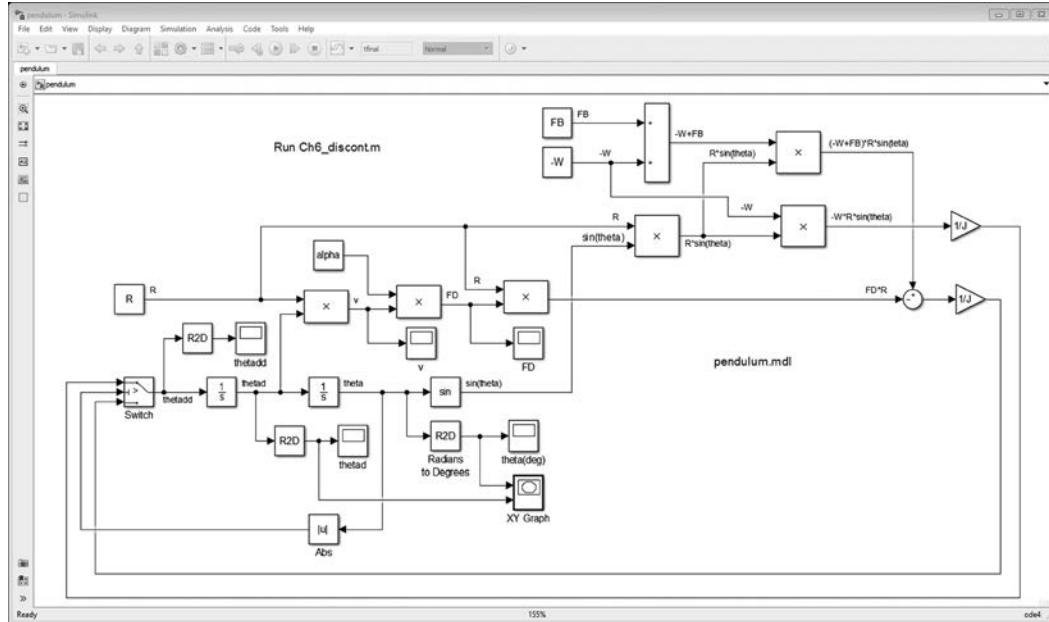


FIGURE 6.58 Simulink diagram for pendulum dynamics.

In addition to the model system parameters, initial conditions must be specified. Choosing $\theta(0) = 75^\circ$, $\dot{\theta}(0) = 0^\circ/\text{s}$, the pendulum dynamics were simulated consistent with the logic outlined in the flow chart shown in Figure 6.56.

A Simulink diagram of the pendulum dynamics using fixed-step RK-4 integration, without searching for the precise time when a discontinuity occurs, is shown in Figure 6.58. A step size of $T = 0.1 \text{ s}$ was used in both cases.

Comparison of the simulation results for the pendulum angle $\theta(t)$ is shown in Figure 6.59. The MATLAB M-file “Ch6_discont.m” was executed with a time step of $T = 0.1 \text{ s}$. A third plot intended to represent the exact solution for $\theta(t)$ is also shown. It was obtained by running the Simulink model

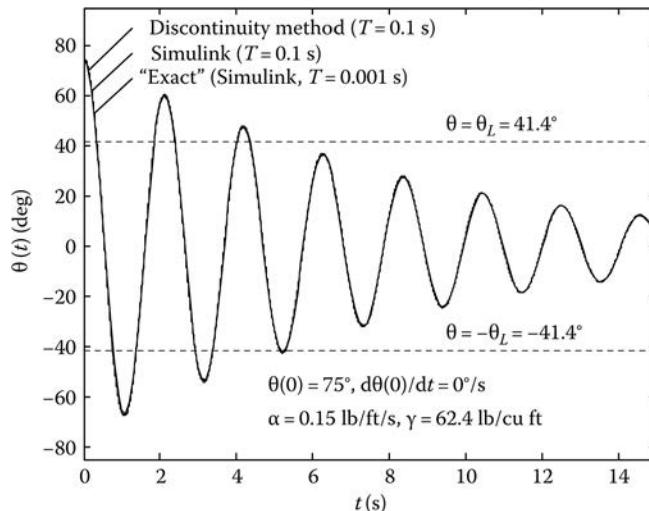


FIGURE 6.59 Simulated results using method for locating discontinuities, Simulink, and approximation to “exact” solution.

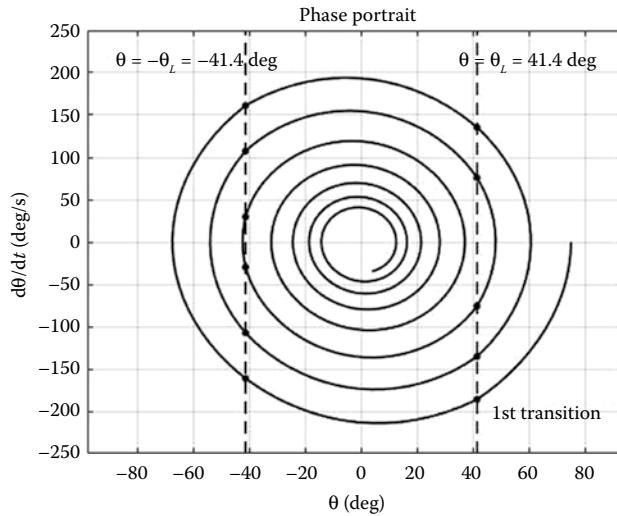


FIGURE 6.60 Plot of state trajectory $x_2(t) = \dot{\theta}(t)$ vs. $x_1(t) = \theta(t)$.

with RK-4 and step size of $T = 0.001$ s. Using a time step of this magnitude negates almost entirely the adverse effect of a discontinuity occurring part way into the integration interval. The three responses are in close agreement resembling that of a lightly damped linear second-order system.

Useful information about the pendulum dynamics can be obtained from inspection of time histories and phase plots of additional system variables. Figure 6.60 is a phase portrait of the state trajectory evolving from the initial point $\theta(0) = 75^\circ$, $\dot{\theta}(0) = 0^\circ/\text{s}$ and lasting for a period of 15 s.

The points along the trajectory where $\theta(t) = \theta_L$ and $\theta(t) = -\theta_L$ indicates a transition from one region to the other, that is, the first marker corresponds to the pendulum entering the liquid for the first time on its way down.

Figure 6.61 includes time histories of $\theta(t)$, $\dot{\theta}(t)$, and $\ddot{\theta}(t)$. In addition, the marker “ISTATE” is shown fluctuating between 1 and 2 corresponding to transitions of the pendulum bob from air to water and vice versa.

The pendulum bob velocity and the drag force exerted by the liquid opposing its motion were captured in the Simulink model scopes and are shown in Figure 6.62.

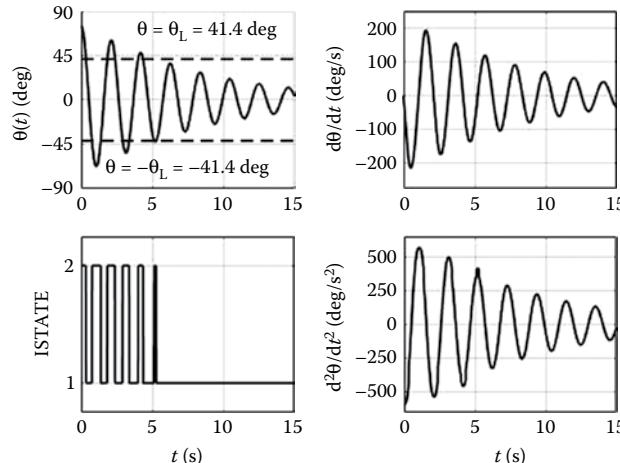


FIGURE 6.61 Time histories of θ , $d\theta/dt$, $d^2\theta/dt^2$ and state marker “ISTATE.”

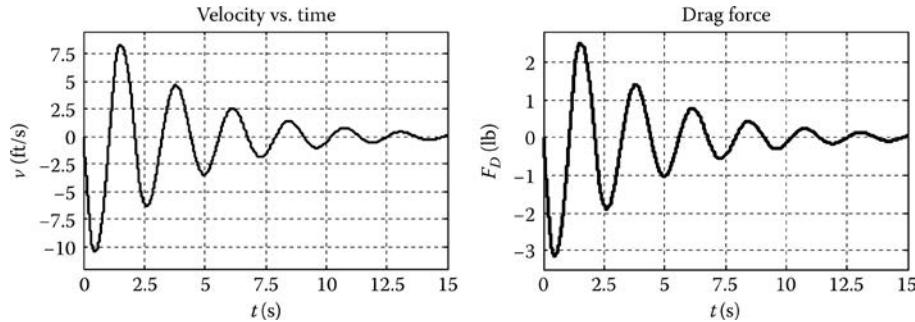


FIGURE 6.62 Velocity and drag force on pendulum bob.

The constant buoyant force of 2.36 lb opposes the motion of the pendulum bob on the way down and does the opposite while the bob is moving upward. The drag force never exceeds 2 lb in magnitude. The pendulum bob weighs 18.6 lb. From Figure 6.62, we notice that it continues to oscillate for a relatively long period of time due to minimal damping forces.

The discontinuous nature of the system is best illustrated by taking a closer look at the angular acceleration. Figure 6.63 shows the step changes that occur as the pendulum bob transitions between the two media. Note that the step changes in angular acceleration are greater at the moments when the pendulum bob is going from air to liquid compared with transitions from liquid to air. Can you explain why this happens? Exercise 6.34 addresses this point in greater detail.

Suppose we increase the damping effect of the liquid by replacing it with a heavier fluid. Instead of water, imagine a liquid with weight density of $\gamma = 150$ lb/ft³ responsible for producing a drag coefficient of $\alpha = 0.3$ lb/ft/s. Further, suppose the pendulum bob is released with an initial angular displacement $\theta(0) = 75^\circ$ and initial velocity of $\dot{\theta}(0) = -90^\circ/\text{s}$.

Figure 6.64 shows a portion of the transient responses obtained from the discontinuity method ($T = 0.1$ s), Simulink with RK-4 ($T = 0.1$ s) and Simulink with RK-4 ($T = 0.001$ s) as the approximation to the exact solution. Values obtained from the method based on locating the points of

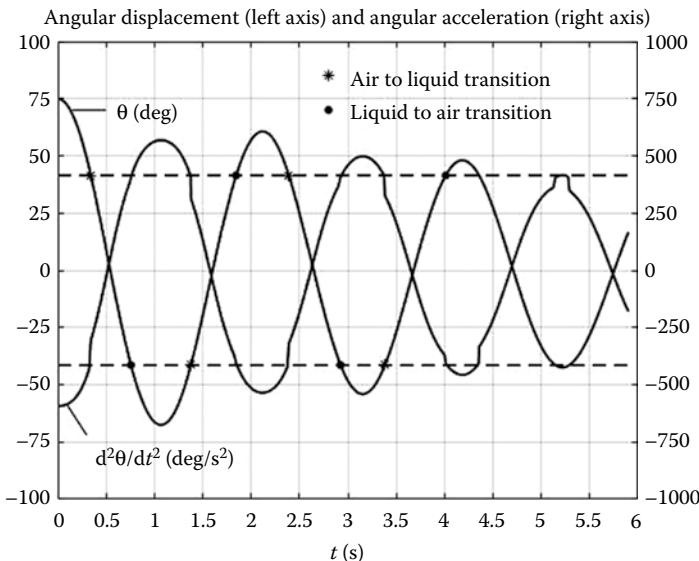


FIGURE 6.63 Angular acceleration showing discontinuities at air/liquid transitions.

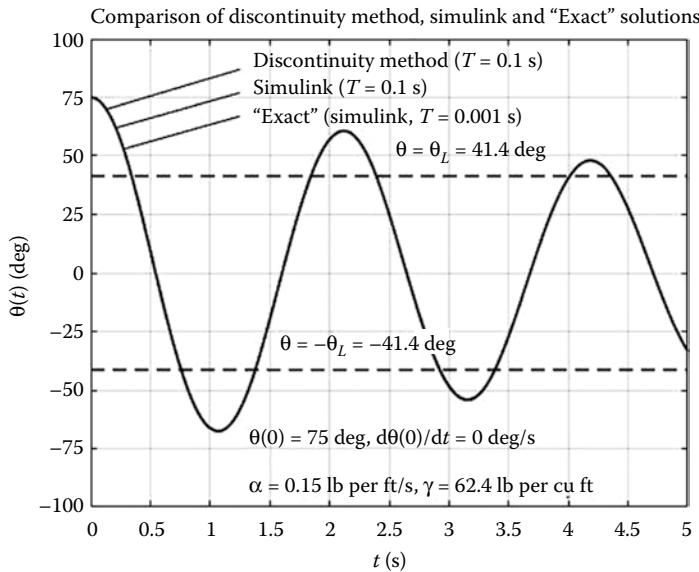


FIGURE 6.64 Comparison of solutions with new initial conditions and parameters.

discontinuity within the integration interval are closer to the “exact” solution than the values obtained from conventional implementation of RK-4 integration.

It is instructive to look at graphs of the discontinuity functions $\phi_1(x_1, x_2)$ and $\phi_2(x_1, x_2)$. Figure 6.65 shows their time histories for the conditions listed in Figure 6.59.

The zero crossings of $\phi_1(x_1, x_2)$ and $\phi_2(x_1, x_2)$ correspond to the transitions of the system between regions S_1 and S_2 . A close-up view of the discontinuity functions is shown in Figure 6.66.

Note how the quadratic interpolation function “INTRP” successfully locates the zero crossings, enabling the RK-4 integrator to stop at the correct point in time within the integration interval, reset the derivative functions, and then continue to integrate for the remainder of the interval as indicated in the flow chart in Figure 6.56.

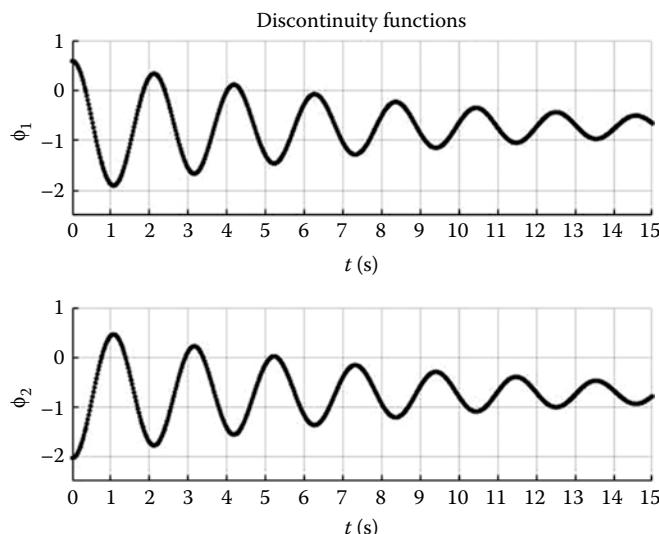


FIGURE 6.65 Plot of discontinuity functions.

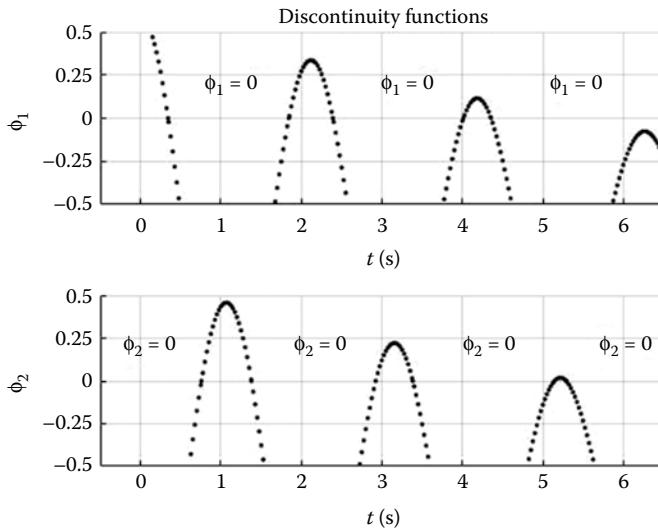


FIGURE 6.66 Close-up view of discontinuity functions $\phi_1(x_1, x_2)$ and $\phi_2(x_1, x_2)$.

EXAMPLE 6.12

Using the baseline conditions for the pendulum except for $\gamma = 150 \text{ lb/ft}^3$ and $\alpha = 0.3 \text{ lb/ft/s}$, determine the largest initial angle of the pendulum rod, so that when it is released with zero initial angular velocity, its fails to emerge from the liquid. Plot the angular rotation of the pendulum as a check.

The problem is to find the initial condition $\theta(0) \geq 0$, which satisfies

$$\min_{\theta(0) \geq 0} \theta(t) = -\theta_L \quad (6.320)$$

A simple search for the required initial condition was performed by varying $\theta(0)$ from 45° to 60° in increments of 0.5° . The results are shown in graphical form in Figure 6.67. The answer appears to be slightly less than 52° .

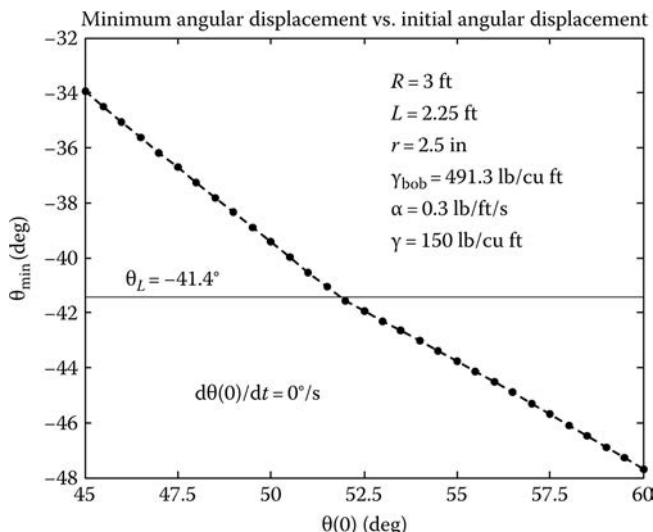


FIGURE 6.67 Results of search for $\theta(0)$ resulting in $\theta_{\min} = -\theta_L$.

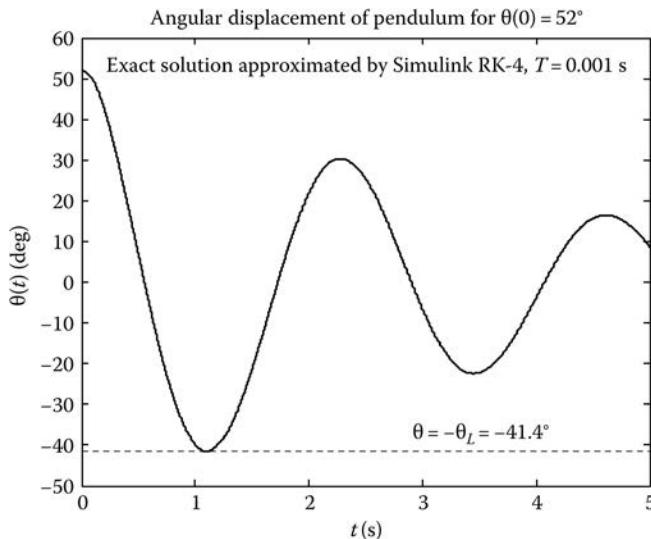


FIGURE 6.68 Simulated pendulum response with initial condition $\theta(0) = 52^\circ$.

The pendulum response with $\theta(0) = 52^\circ$ was generated for the conditions shown in Figure 6.68 using RK-4 integration with step size $T = 0.001$ s. The result is shown in Figure 6.68. As expected, the minimum angular response is approximately $-\theta_L = -41.4^\circ$.

EXERCISES

- 6.32 The pendulum bob in Example 6.12 is released from the vertical position $\theta(0) = \pi$ rad with initial angular velocity $\dot{\theta}_0$. Find $\dot{\theta}_0$ if the bob makes a complete revolution and returns to the vertical position with zero angular velocity.
- 6.33 The pendulum bob shown in Figure E6.33 passes through two different nonmixing liquids. Physical parameter values are

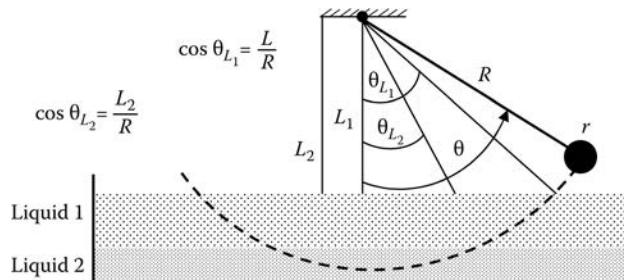


FIGURE E6.33

Radius of spherical pendulum bob: $r = 3$ in

Density of pendulum bob: $\gamma_{\text{bob}} = 250$ lb/ft³

Length of negligible mass pendulum rod: $R = 4$ ft

Vertical distance from center of rotation to liquid 1 surface: $L_1 = 2.5$ ft

Vertical distance from center of rotation to liquid 2 surface: $L_2 = 3.5$ ft

Density of liquid 1: $\gamma_1 = 62.4$ lb/ft³

Density of liquid 2: $\gamma_2 = 175$ lb/ft³

Drag coefficient on pendulum bob in liquid 1: $\alpha_1 = 0.10$ lb/ft/s

Drag coefficient on pendulum bob in liquid 2: $\alpha_2 = 0.65$ lb/ft/s

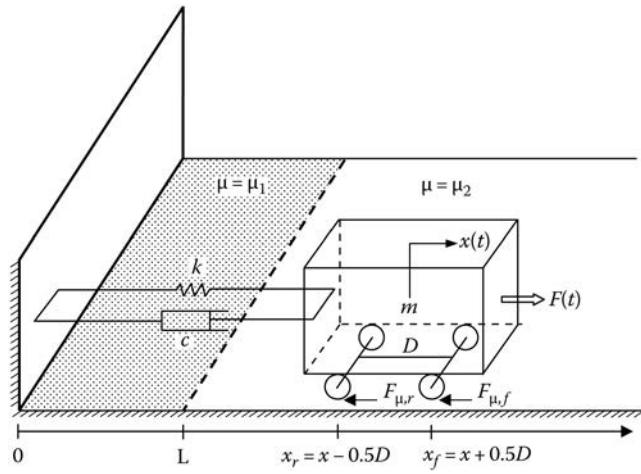
With state vector $(x_1, x_2) = (\theta, \dot{\theta})$, the state equations are

$$\dot{x}_1 = \begin{cases} f_{11}(x_1, x_2), & (x_1, x_2) \in S_1 \\ f_{12}(x_1, x_2), & (x_1, x_2) \in S_2 \\ f_{13}(x_1, x_2), & (x_1, x_2) \in S_3 \end{cases} \quad \dot{x}_2 = \begin{cases} f_{21}(x_1, x_2), & (x_1, x_2) \in S_1 \\ f_{22}(x_1, x_2), & (x_1, x_2) \in S_2 \\ f_{23}(x_1, x_2), & (x_1, x_2) \in S_3 \end{cases}$$

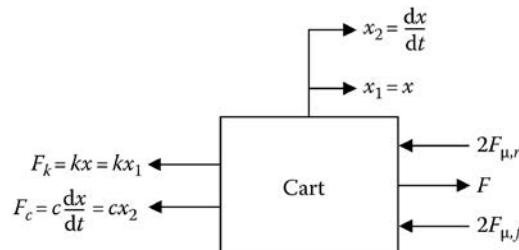
where the regions S_1 , S_2 , and S_3 in state space are described by

- S_1 : $\{(x_1, x_2) | \text{pendulum bob in air}\}$
- S_2 : $\{(x_1, x_2) | \text{pendulum bob in liquid 1}\}$
- S_3 : $\{(x_1, x_2) | \text{pendulum bob in liquid 2}\}$

- a. Find expressions for S_1 , S_2 , and S_3 similar to those in Equation 6.310.
 - b. Find the state derivative functions f_{ij} , $i = 1, 2, j = 1, 2, 3$.
 - c. Find the discontinuity functions $\phi_1(x_1, x_2)$, $\phi_2(x_1, x_2)$, $\phi_3(x_1, x_2)$, and $\phi_4(x_1, x_2)$, where $\phi_i(x_1, x_2) = 0$, $i = 1, 2, 3, 4$ indicates the pendulum bob is passing from region S_i to S_1 , S_2 to S_3 , S_3 to S_2 , and S_2 to S_1 , respectively.
 - d. Use the method outlined in the flow chart in Figure 6.66 to simulate the angular position and angular velocity of the pendulum for initial conditions $\theta(0) = 90^\circ$ and $\dot{\theta}(0) = 0^\circ/\text{s}$. Choose any of the RK integrators with integration step size T selected on the basis of a trade-off between accuracy and computational effort. Plot time histories of $\theta(t)$ and $\dot{\theta}(t)$ as well as a phase portrait similar to the one in Figure 6.60 showing the points where the system transitions between regions.
 - e. Simulate the same conditions in part (d) using Simulink with an excessively small integration step size T in order to obtain an approximation to the exact solution. Compare the results in parts (d) and (e).
- 6.34 According to the graphs in Figure 6.63, the angular acceleration appears to be continuous when the pendulum bob passes from liquid to air for the first time.
- a. Verify this by plotting $d^2\theta/dt^2$ vs. t shortly before to shortly after this occurs.
 - b. For this to happen, the component of the buoyant force F_B in the direction of motion and the drag force F_D must effectively cancel each other out. On the same axes, plot both quantities and compare them at the moment the pendulum bob exits from the liquid for the first time.
- 6.35 Consider the pendulum in Figure 6.53 with physical properties
- Radius of spherical pendulum bob: $r = 2 \text{ in}$
 - Density of pendulum bob: $\gamma_{\text{bob}} = 400 \text{ lb/ft}^3$
 - Length of negligible mass pendulum rod: $R = 5 \text{ ft}$
 - Vertical distance from center of rotation to liquid surface: $L = 3 \text{ ft}$
- The drag coefficient α ($\text{lb}/(\text{ft/s})$) is related to the density of the liquid γ (lb/ft^3) according to the relationship $\alpha = 0.05 + 0.02\gamma$, $50 \leq \gamma \leq 400$.
- The pendulum is released from an almost vertical position $\theta(0) = 179.9^\circ$ with zero angular velocity. Simulate the pendulum dynamics using any suitable method and prepare graphs of
- a. θ_{max} vs. γ ($50 \leq \gamma \leq 400$) where θ_{max} is the total number of degrees the pendulum rotates through on its first swing.
 - b. $t_{\text{settling time}}$ vs. γ ($50 \leq \gamma \leq 400$) where $t_{\text{settling time}}$ is the time in seconds for the transient response to remain within 2% of its steady-state equilibrium value $\theta_{ss} = 0^\circ$.
 - c. θ_{max} vs. γ ($50 \leq \gamma \leq 400$) where $|\dot{\theta}_{\text{max}}|$ is the absolute value of the maximum angular velocity in $^\circ/\text{s}$.
- 6.36 A rolling cart of mass m is connected to a stationary support located at $x = 0$ by a spring with stiffness k and damper with damping constant c as shown in Figure E6.36a:

**FIGURE E6.36A**

The cart is subjected to an external force $F(t)$. The coefficient of rolling friction changes from surface 1 ($\mu = \mu_1$) to surface 2 ($\mu = \mu_2$) at $x = L$. The frictional force at each wheel is $F_\mu = \mu (mg/4)$ where μ is either μ_1 or μ_2 depending on which surface the wheel is in contact with. A diagram of the cart and the forces acting on it is shown in [Figure E6.36b](#). Note that the state definition $x_1 = x$ and $x_2 = \dot{x}$.

**FIGURE E6.36B**

Introduce regions S_1 , S_2 , and S_3 in state space $\{x_1, x_2\}$ according to the cart location, that is,

$$S_1 = \{(x_1, x_2) | x_1 + 0.5D < L\} \text{—cart is located entirely on surface 1}$$

$$S_2 = \{(x_1, x_2) | x_1 - 0.5D < L \text{ and } x_1 + 0.5D > L\} \text{—cart is on surface 1 and surface 2}$$

$$S_3 = \{(x_1, x_2) | x_1 - 0.5D > L\} \text{—cart is located entirely on surface 2}$$

- Find expressions for $f_{ij}(x_i, x_2)$, $i = 1, 2, j = 1, 2, 3$, the i th state derivative \dot{x}_i when the state (x_1, x_2) is located in region S_j .
- Find the discontinuity functions $\phi_1(x_1, x_2)$ and $\phi_2(x_1, x_2)$ where
 $\phi_1(x_1, x_2) = 0 \Rightarrow (x_1, x_2)$ is transitioning between S_1 and S_2
 $\phi_2(x_1, x_2) = 0 \Rightarrow (x_1, x_2)$ is transitioning between S_2 and S_3
- Implement the method outlined in the flow chart of [Figure 6.56](#) using RK-4 integration with integration step size T , based on a trade-off between accuracy and computational effort, to simulate the cart dynamics. Baseline conditions are

$$\mu_1 = 0.4, \mu_2 = 0.05$$

$$m = 30 \text{ slugs}, c = 5 \text{ lb/ft/s}, k = 25 \text{ lb}_f/\text{ft}$$

$$L = 25 \text{ ft}, D = 5 \text{ ft}$$

$$x(0) = L, \dot{x}(0) = 0 \text{ ft/s}$$

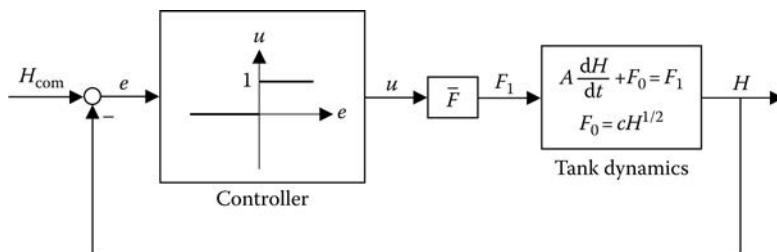
The applied force $F(t)$ is a step input of magnitude $F_0 = 250$ lb. Plot the cart position $x(t)$ vs. time and cart velocity $\dot{x}(t)$ vs. time.

- d. Simulate the cart dynamics with Simulink, and compare the results with those obtained in part (c).
- 6.37 A block diagram for a simple on-off tank level control system is shown in below figure. The flow in to the tank F_1 is either zero or \bar{F} depending on the state of the on-off controller, that is,

$$F_1 = \bar{F}u, \quad u = \begin{cases} 0, & e \leq 0 \\ 1, & e > 0 \end{cases} \quad \text{where } e = H_{\text{com}} - H$$

The tank dynamics is modeled by

$$A \frac{dH}{dt} + F_0 = F_1, \quad F_0 = cH^{1/2}$$



- a. The state derivative is

$$\frac{dH}{dt} = \begin{cases} f_1(H), & H \in S_1 \\ f_2(H), & H \in S_2 \end{cases}$$

Regions S_1 and S_2 are defined such that when H is in region S_1 of state space, the controller is off, and the opposite is true when H is in region S_2 . Find expressions for S_1 and S_2 in terms of the state H .

- b. Find expressions for the state derivate functions $f_1(H)$ and $f_2(H)$.
- c. Find the discontinuity function $\phi(H)$ that specifies which region the state is in based on its sign, that is, $\phi(H) = 0$ implies the state H is transitioning between the two regions.
- d. Use the method that finds the time of the discontinuity to simulate the tank level. Choose any RK integrator with suitable integration step size based on accuracy and computation requirements.

The following conditions apply:

$$A = 20 \text{ ft}^2, c = 0.4 \text{ ft}^3/\text{min}/\text{ft}^{1/2}, \bar{F} = 10 \text{ ft}^3/\text{min}, H(0) = 0 \text{ ft}$$

$$H_{\text{com}} = 15, \quad t \geq 0$$

Run the simulation for a period of time sufficient for the controller to cycle on and off several times and plot time histories of $H(t)$ and $\dot{H}(t)$.

- e. Plot a phase portrait \dot{H} vs. H showing the points where the controller cycles between its two states.

- f. Simulate the system for the same conditions in part (d) with Simulink using RK-4 integration with an excessively small step size in order to approximate the exact solution. Compare the results with those in part (d).

6.8 CASE STUDY: SPREAD OF AN EPIDEMIC

Epidemic models for various fatal and nonfatal diseases in humans and animals have been postulated since the early 1900s (Kermack and McKendrick 1927; Hethcote 1976; Keen and Spain 1992; Brown and Rothery 1993). Modern-day epidemics such as the spread of AIDS have been studied with the help of simulation models (Isham 1988; Perelson 1993; Culshaw and Ruan 2000; Coutinho *et al.* 2001).

The formulation of a mathematical model in the field of epidemiology requires some basic information about disease and how it spreads among a population. To start with, symptoms of the disease may not appear at the time a host is infected, rather an incubation period may be necessary prior to appearance of the symptoms. A host infected with a pathogen may become infectious only after a period of latency. The infectious period is the duration of time during which the host is capable of transmitting the disease to others in the population. The incubation, latent, and infectious periods depend on the pathology of the disease.

For certain diseases, the host may experience an immune period where the infection has run its course, the host has recovered, and cannot be re-infected. However, the individual may still be a carrier and capable of transmitting the disease to susceptible individuals. As a means of preventing or limiting the scope of an epidemic, some infected individuals may be isolated from the population to prevent transmission of the disease to susceptible individuals. If the disease is potentially fatal, a number of infected individuals will die. If a vaccine exists, individuals receiving the vaccine pass from the class of susceptibles to the class of recovered individuals.

Early epidemic models concentrated on the movement of individuals through three stages, namely, (S)usceptible, (I)nfected carrier, and (R)ecovered. The so-called S-I-R models relate the state derivatives dS/dt , dI/dt , and dR/dt to the states S , I , and R using expressions formulated by epidemiologists to describe the interactions between individuals in each group. Inherent in the models are a number of parameters (rate constants) associated with infection, transmission, recovery, mortality, and so forth. Later on, more sophisticated models were developed to account for additional stages. Finally, partial differential equations evolved as modelers attempted to predict both temporal and spatial variations of the populations in each stage during the course of an epidemic.

The following information is postulated to provide a framework for studying the dynamics of an epidemic stemming from the spread of a fatal disease.

- The initial population consists entirely of susceptible individuals, that is, those at risk of contracting the disease.
- The disease is introduced by individuals immigrating from outside the area, a fraction of which are sick.
- A subset of the susceptible individuals contract the disease through contact with sick individuals.
- An outbreak of the disease is recognized after a specified period of time immediately followed by a cessation of immigration.
- After recognizing the existence of a possible epidemic, a segment of the susceptible individuals is inoculated with a vaccine making them immune to the disease.
- Starting at the same time inoculations begin, a portion of those who are sick or become sick later are separated from the general population by quarantine.
- Sick individuals either recover and become immune or die.

Members of the population exist in one of five states.

$x_1(t)$: Number of susceptible people at time t

$x_2(t)$: Number of sick people in population at time t

$x_3(t)$: Number of immune people at time t

$x_4(t)$: Number of deceased people at time t

$x_5(t)$: Number of sick people quarantined from population at time t

Possible transitions between states are illustrated in Figure 6.69. Note that the $m = m(t)$ is the rate of immigration and $n = n(t)$ represents the rate of inoculation of susceptible individuals.

The state vector \underline{x} is $[x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T$. A mathematical model of the system requires knowledge of a vector function $f(t, \underline{x}, m)$, describing the state derivatives. In this example, the system of coupled differential equations $\dot{\underline{x}} = f(t, \underline{x}, m)$ is given by

$$\frac{dx_1}{dt} = f_1(t, \underline{x}, m) = -cx_1x_2 + \alpha m - n \quad (6.321)$$

$$\frac{dx_2}{dt} = f_2(t, \underline{x}, m) = cx_1x_2 - a_{23}x_2 - a_{24}x_2 - a_{25}x_2 + (1 - \alpha)m \quad (6.322)$$

$$\frac{dx_3}{dt} = f_3(t, \underline{x}, m) = a_{23}x_2 + a_{53}x_5 + n \quad (6.323)$$

$$\frac{dx_4}{dt} = f_4(t, \underline{x}, m) = a_{24}x_2 + a_{54}x_5 \quad (6.324)$$

$$\frac{dx_5}{dt} = f_5(t, \underline{x}, m) = \begin{cases} 0, & 0 \leq t \leq t_0 \\ a_{25}x_2 - a_{53}x_5 - a_{54}x_5, & t > t_0 \end{cases} \quad (6.325)$$

The constants a_{23} , a_{24} , a_{25} , a_{53} , a_{54} , c , and α are system parameters, which describe the transitions by individuals from one state to another. For example, the disease spreads by contact between susceptible and sick members of the population, and c is a transmission constant. The constant α is the fraction of immigrants who are susceptible. All terms on the right-hand side of Equations 6.321 through 6.325 are in units of individuals per unit of time, the same as the left-hand-side state derivatives.

The time t_0 in Equation 6.325 is the length of time it takes to recognize the outbreak of a possible epidemic. Quarantining of sick people, cessation of immigration, and inoculation of susceptible individuals begin at $t = t_0$. Immigration and inoculation profiles are shown in Figure 6.70.

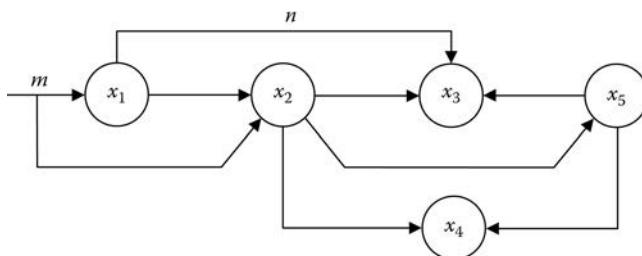


FIGURE 6.69 State transition diagram.

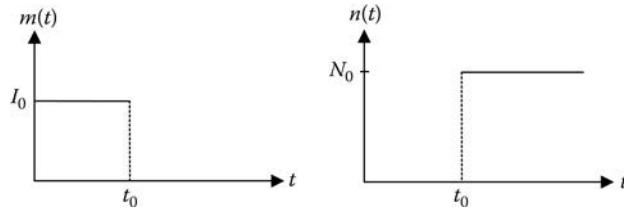


FIGURE 6.70 Immigration and inoculation profiles.

The simple model ignores birth and deaths from other causes and does not account for emigration of individuals. The following values have been arbitrarily selected for conducting a baseline study.

$$a_{23} = 0.1 \text{ per week}, a_{24} = 0.003 \text{ per week}, a_{25} = 0.05 \text{ per week}$$

$$a_{53} = 0.1 \text{ per week}, a_{54} = 0.003 \text{ per week}$$

$$\alpha = 0.9, c = 2.25 \times 10^{-8} \text{ people}^{-1}/\text{week}$$

$$t_0 = 8 \text{ weeks}, I_0 = 2500 \text{ people/week}, N_0 = 0 \text{ inoculations/week}$$

Note that the baseline conditions assume zero inoculations following the recognition of a possible epidemic. A number of interesting simulation studies are possible. First, we will investigate various inoculation policies and their mitigating effect on spreading of the disease in a population initially consisting of 10 million susceptible individuals.

The classic RK-4 numerical integrator introduced in Equations 8.60 through 8.64 was chosen for simulating the system response. After several trial runs with different integration step sizes, $T = 0.1$ weeks were selected. The results of a baseline and additional simulations using inoculation rates of 5000, 10,000, and 15,000 people per week are shown in Figures 6.71 through 6.74. Refer to

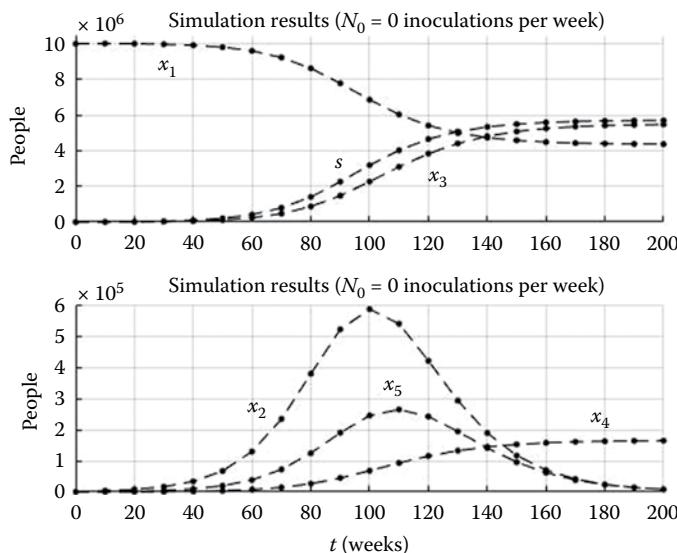


FIGURE 6.71 Epidemic response for baseline conditions ($N_0 = 0$ inoculations/week).

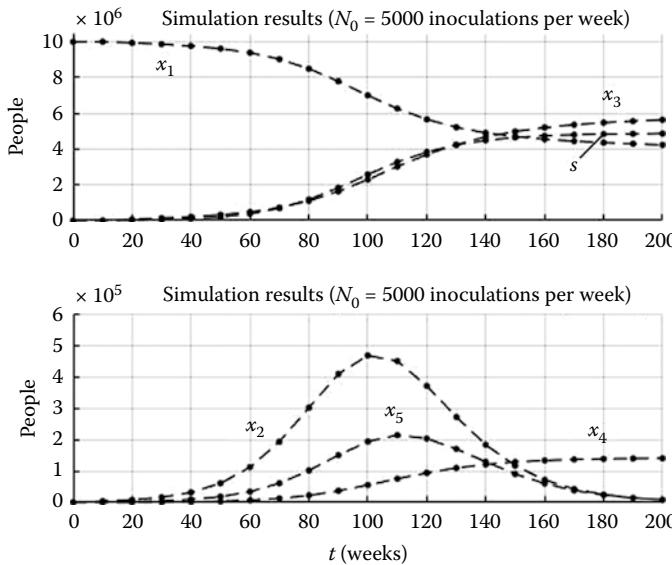


FIGURE 6.72 Epidemic response ($N_0 = 5000$ inoculations/week).

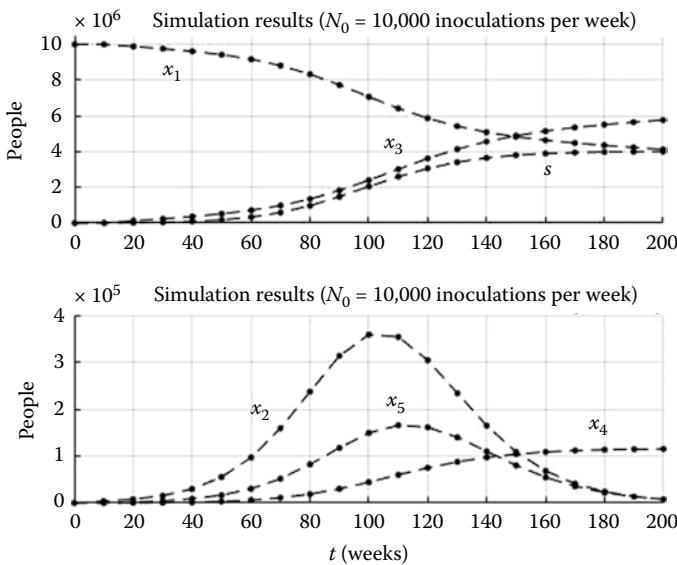


FIGURE 6.73 Epidemic response ($N_0 = 10,000$ inoculations/week).

MATLAB M-file “*Ch6_CaseStudy.m*.” For simplicity, the subscript “A” has been dropped from the notation for the discrete-time signals.

A summary of the results is listed in [Table 6.12](#).

As expected, the highest inoculation level results in the fewest deaths. The third row shows the maximum number of sick people at any time in the 200 week study period. The peak is reduced from 585,834 sick at one time to 268,548 as a result of administering 15,000 vaccinations/week compared with none at all.

The discrete-time state variable $x_2(i)$ represents the number of infected individuals at the discrete times $t_i = iT$, $i = 0, 1, 2, \dots$. The cumulative number of people who have been sick up through time

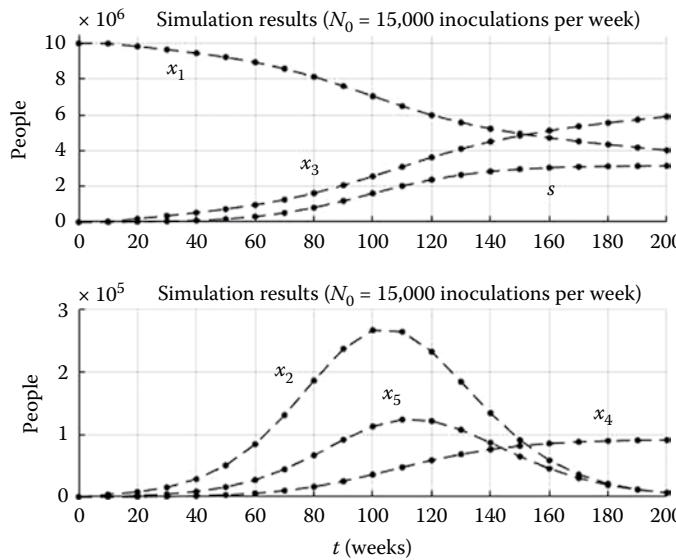


FIGURE 6.74 Epidemic response ($N_0 = 15,000$ inoculations/week).

TABLE 6.12
Summary of Epidemic Simulation Results after 200 Weeks

	$N_0 = 0$	$N_0 = 5000$	$N_0 = 10000$	$N_0 = 15000$
$x_1(200)$	4,368,093	4,233,072	4,125,930	4,016,139
$x_2(200)$	8089	8038	7614	6395
Max x_2	585,834	469,690	362,637	268,548
$x_3(200)$	5,471,549	5,630,535	5,763,273	5,900,147
$x_4(200)$	164,146	140,116	115,298	90,604
$x_5(200)$	8123	8238	7885	6715
$s(200)$	5,700,412	4,867,698	4,007,144	3,149,525

t_i is denoted $s(i)$ (see Figures 6.71 through 6.74 and Table 6.12). It is computed by numerical integration of ds/dt , where (Figures 6.72 and 6.73)

$$\frac{ds}{dt} = (1 - \alpha)m + cx_1x_2 \quad (6.326)$$

Note the difference between ds/dt and dx_2/dt . The former is the rate of change of newly infected individuals, that is, those people entering state x_2 . As a result, $s(t)$ is monotonically increasing. The state derivative dx_2/dt is the overall rate of change of infected people in the nonquarantined population. It is negative when more individuals are leaving state x_2 than entering, which results in $x_2(t)$ decreasing [Figure 6.74].

The same RK-4 integration method and step size were used to numerically integrate the discrete-time signal $(1 - \alpha)m(i) + cx_1(i)x_2(i)$ to generate $s(i)$.

A valuable check on the accuracy of the simulation is possible. Conservation of individuals can be verified at every discrete point in time. In this case, the total number of individuals begins at 10 million and increases at a rate of 2500 per week for 8 weeks. Hence, after approximately 2 months, the total population consists of 10,020,000 people distributed among the five states x_1 , x_2 , x_3 , x_4 , and

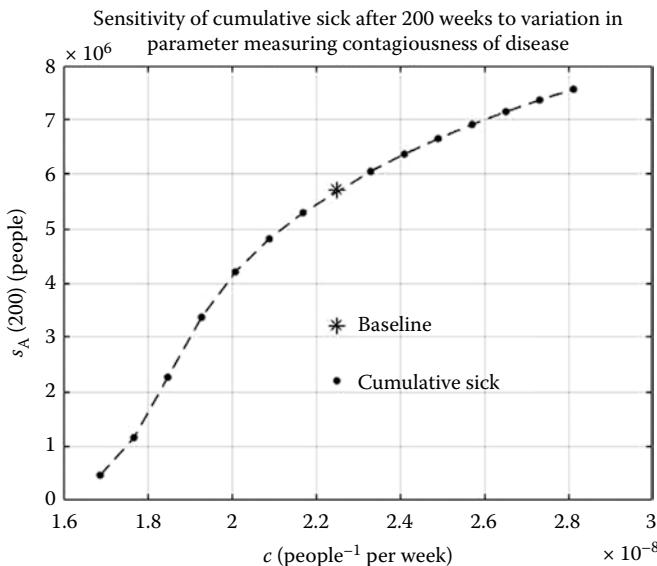


FIGURE 6.75 Sensitivity analysis: $s(200)$ vs. c .

x_5 . Summing $x_1(200)$, $x_2(200)$, $x_3(200)$, $x_4(200)$, and $x_5(200)$ in each column of Table 6.12 will show that all individuals are accounted for. This is crucial in the context of real-world simulations where analytical solutions of the continuous-time model are not available.

Sensitivity analyses with respect to each system parameter at baseline conditions offer insight into the dynamics of the epidemic. To illustrate this, suppose we are interested in relating the number of individuals who contract the disease with the system parameter that measures how contagious the disease is, that is, the transmission coefficient c that appears in Equations 6.321 and 6.322. The parameter was allowed to vary by 25% in both directions from the nominal or baseline value $c = 2.25 \times 10^{-8}$ people $^{-1}$ /week, and the simulation is repeated with the remaining parameters fixed at their baseline values. Figure 6.75 shows that $s(200)$, the predicted number of sick people in the first 200 weeks, increases as the transmission coefficient parameter c increases, as one would expect.

A similar study was conducted to investigate the relationship between the cumulative number of deaths $x_4(200)$ and the transmission coefficient c . The graph in Figure 6.76 shows what can be expected in terms of the number of people dying over the 200 week period as the level of contagiousness varies about the baseline value.

Try running the simulation with the same baseline conditions to ascertain the numerical value of c that results in the epidemic spreading to every member of the population. A number of other studies are suggested in the exercise problems.

EXERCISES

6.38 In the study that looked at the effect of various inoculation rates, prepare graphs of

- a. Cumulative sick vs. inoculation rate
- b. Number of deaths vs. inoculation rate
- c. Peak number of sick vs. inoculation rate

Use the classic RK-4 integrator with baseline values for all parameters (except inoculation rate), and run the simulations for a sufficient period of time to include the transient response. Consider inoculation rates from 0 to 50,000 per week.

Repeat parts (a), (b), and (c) using Simulink with the same numerical integrator.

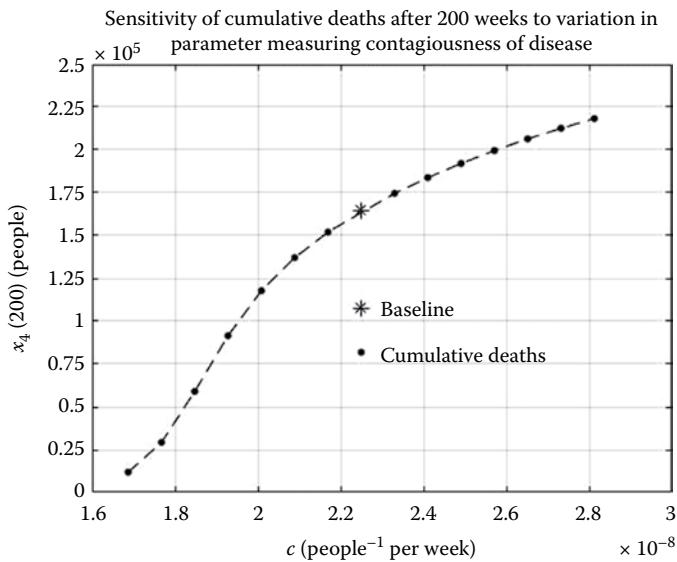


FIGURE 6.76 Sensitivity analysis: $x_4(200)$ vs. c .

- 6.39 In the inoculation study, find the cumulative number of individuals quarantined.

Hint: Let $q(t)$ represent the cumulative number of people quarantined through time t , and write the differential equation for $q(t)$ similar to the procedure for finding $s(t)$.

- 6.40 Investigate the duration of the epidemic transient period as a function of the parameter c . Does the epidemic last longer when the disease is more contagious?



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

7 Simulation Tools

7.1 INTRODUCTION

Mathematical models of dynamic systems are derived with their intended use in mind. For example, systems with fast internal dynamics driven by inputs that change infrequently (relative to the system time constants) reside in steady state the majority of the time. Accordingly, the model consists of a system of coupled, possibly nonlinear, algebraic equations. In this context, a solution (or solutions) defines an equilibrium state (or states) corresponding to fixed values of the system inputs. When one or more inputs change, a stable system transitions from one equilibrium state to another and the dynamics, that is, transient response, is ignored. Solving the steady-state algebraic equations for an equilibrium solution is rarely a straightforward task, particularly when dealing with nonlinear systems. The MATLAB steady-state solver is introduced in Section 7.2. It is designed to locate equilibrium states of a Simulink model.

Tuning a simulation model of a real, continuous-time system is an iterative process like the one shown in [Figure 7.1](#). Assumptions about the structure of the mathematical model, that is, the state derivative vector $f(x, u, p)$ and parameter values p , are tested and refined using observed data acquired from the system. An essential component of the validation process is minimization of an error function $\varepsilon(f, p)$, a measure of the differences between the actual system and simulation model outputs.

A Simulink add-on called Parameter Estimation compares empirical data with data generated by a Simulink model. Using optimization techniques, Simulink Parameter Estimation estimates the parameter and (optionally) initial conditions of states such that a user-selected cost function is minimized. The cost function typically calculates a least-square error between the empirical and model data.

Once the mathematical model has been determined, some type of exploratory study can be performed to determine the “best” (in some sense) values for the controllable system parameters. Optimization theory is a broad area of study with roots in *Operations Research and Applied Mathematics*. It is the foundation for implementation of what are collectively called optimum seeking methods. The MATLAB optimization toolbox provides the simulationist access to a number of algorithms for locating points in the model’s parameter space where the system performs at optimum or near optimum levels. Examples are presented in Section 7.3 of using optimization for both parameter identification and system optimization involving Simulink models.

Another Simulink add-on, Response Optimization, is a tool that helps you tune design parameters in Simulink models by optimizing time-based signals to meet user-defined constraints. It supports continuous-time, discrete-time, and multirate models accounting for model uncertainty by conducting Monte Carlo simulations. Simulink Response Optimization can be used to tune multiinput/multioutput and adaptive controllers in nonlinear systems and optimize physical parameters to minimize power consumption, reduce range of motion, and tune filter coefficients.

An equilibrium state is sometimes required to serve as the initial state for a simulation investigation of the system’s dynamic response. After locating an equilibrium point of a nonlinear system model, the system’s response to dynamically changing inputs can be approximated by linearizing the equations about the equilibrium point. Accuracy of the linearized model depends in part on the magnitude of the state vector’s excursions from the equilibrium state. In general, if the changing input vector remains in close proximity to its equilibrium level, the state vector will do the same. Regulatory control systems are a good example of an application where linearization of process models has proven beneficial in both the design and analysis of the system. Section 7.4 illustrates the capabilities of MATLAB to linearize nonlinear models created with Simulink.

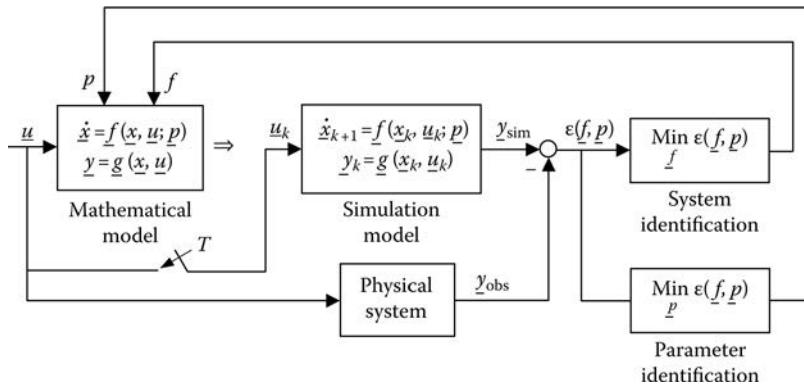


FIGURE 7.1 Iterative procedure for simulation model validation.

7.2 STEADY-STATE SOLVER

Unlike linear system models, it is possible for nonlinear systems to possess any number of equilibrium points, that is, points in state space where the state derivatives are all zero. Furthermore, there is no uniform approach guaranteed to determine the number or location of the equilibrium points.

Knowledge of a nonlinear system's equilibrium points is important for several reasons. The first relates to stability. Once an equilibrium point is located, stability can be determined by linearizing the model's state equations in the neighborhood of the equilibrium point. Second, the behavior of forced nonlinear systems is often approximated by "small signal" linearized models. The characteristic dynamics (time constants, poles, critical frequencies, eigenvalues, and so forth.) of the linearized system depend on the location of the equilibrium point. Linearization is discussed in a later section.

Consider a nonlinear state model

$$\dot{x} = \underline{f}(x, u) \quad (7.1)$$

where

x and u are the state and input vectors, respectively

$\underline{f}(x, u)$ is a vector of functions defining the state derivatives

Equilibrium points x_e corresponding to a constant input vector u_e are solutions to the nonlinear system of algebraic equations

$$\underline{f}(x_e, u_e) = 0 \quad (7.2)$$

Some type of numerical method for finding the solutions $(x_e)_1, (x_e)_2, \dots$, given the input u_e , is needed. Nonlinear autonomous systems described by

$$\dot{x} = \underline{f}(x) \quad (7.3)$$

may also possess a finite (or infinite) number of equilibrium points that satisfy

$$\underline{f}(x_e) = 0 \quad (7.4)$$

To illustrate the point, we focus on a nonlinear system model from the field of ecology. A predator-prey model for the population of fish (prey) and sharks (predator) in the ocean is (Haberman 1997)

$$\frac{dF}{dt} = F(a - bF - cS) \quad (7.5)$$

$$\frac{dS}{dt} = S\left(e - \lambda \frac{S}{F}\right) \quad (7.6)$$

where $F = F(t)$ and $S = S(t)$ are the instantaneous populations (or population densities) of fish and sharks in a fixed geographical area. The system is autonomous since there are no fish or sharks entering or leaving the region according to an external function of time t . Conversely, harvesting of either population according to some predetermined schedule, independent of the levels F and S , would require additional terms with explicit dependence on t resulting in a nonautonomous system model.

The model equations are based on the following observations:

1. The growth rate of fish $(1/F)dF/dt$ is reduced from a constant “ a ” by an amount proportional to the number of fish (which compete for the limited food supply) as well as an amount proportional to the number of sharks (for which the fish are the primary food source). Proportionality constants b and c reflect the level of competition among the fish for their food and the aggressiveness of the sharks.
2. Shark growth rate $(1/S)dS/dt$ is reduced from a constant e by an amount proportional to the ratio of sharks to fish. A higher S/F depletes the fish supply more rapidly.

Equilibrium points (F_e, S_e) satisfy the steady-state algebraic equations resulting from setting the state derivatives to zero. Thus,

$$0 = F_e(a - bF_e - cS_e) \quad (7.7)$$

$$0 = S_e\left(e - \lambda \frac{S_e}{F_e}\right) \quad (7.8)$$

There is more than one equilibrium point (see Exercise 7.2). Our interest is in the nontrivial equilibrium point where neither fish nor sharks vanish. From Equation 7.8 with the term in parenthesis equal to zero,

$$S_e = \frac{e}{\lambda}F_e \quad (7.9)$$

Substituting S_e from Equation 7.9 into Equation 7.7 gives (after simplification)

$$F_e = \frac{a\lambda}{b\lambda + ce} \quad (7.10)$$

Finally, S_e is obtained from Equations 7.9 and 7.10 as

$$S_e = \frac{ae}{b\lambda + ce} \quad (7.11)$$

We shall return to the predator-prey model to investigate the dynamic interaction between fish and sharks, particularly in the vicinity of the equilibrium point (F_e, S_e) .

7.2.1 TRIM FUNCTION

Figure 7.2 shows a Simulink diagram for simulating the predator-prey ecosystem modeled by Equations 7.5 and 7.6. The model name is “*Fish_Sharks.mdl*.” Parameters “*a*,” “*b*,” “*c*,” “*e*,” and “*lam*” are assigned values in the MATLAB M-file “*Ch7_Fish_Sharks.m*.” There are no inputs and the two states are designated as outputs.

A function “*trim*” is called from MATLAB to search for equilibrium points associated with a named Simulink model file. The “*trim*” function call is

```
[x, u, y, dx] = trim ('Fish_sharks', x0)
```

The second parameter “*x0*” is the starting point in state space in the search for an equilibrium point. The output contains the equilibrium state “*x*,” input and output vectors “*u*” and “*y*” at equilibrium, respectively, and the value of the state derivative vector at the equilibrium point. Empty vectors are returned when there are no inputs or outputs defined. Should the numerical search algorithm fail to converge to an equilibrium point, a different starting point will sometimes fix the problem.

Optional parameters are available for constraining selected components of the state, input, and output vectors. For example, instead of a true equilibrium point, we may look for points in the state space where a subset of the state derivative vector is zero.

Numerical values of the parameters in Equations 7.5 and 7.6 were arbitrarily chosen as $a = 50$, $b = 1$, $c = 5$, $e = 2$, and $\lambda = 10$. Running M-file “*Ch7_Fish_Sharks.m*” with starting point $x0 = (18.75; 3.75)$ results in

```
x = 25.0000 u = Empty matrix: 0 - by - 1 y = 25.0000
      5.0000
dx = 1.0e - 011 * - 0.1954
      0.0026
```

According to Equations 7.10 and 7.11,

$$F_e = \frac{a\lambda}{b\lambda + ce} = \frac{50(10)}{1(10) + 5(2)} = 25, \quad S_e = \frac{50(2)}{1(10) + 5(2)} = 5$$

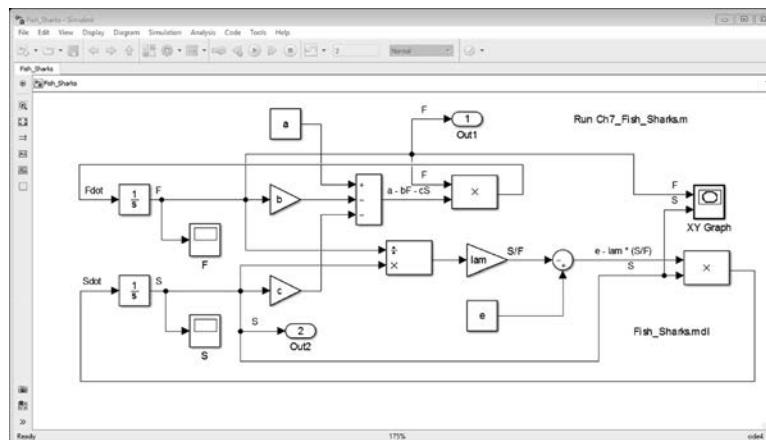


FIGURE 7.2 Simulink diagram of predator-prey model.

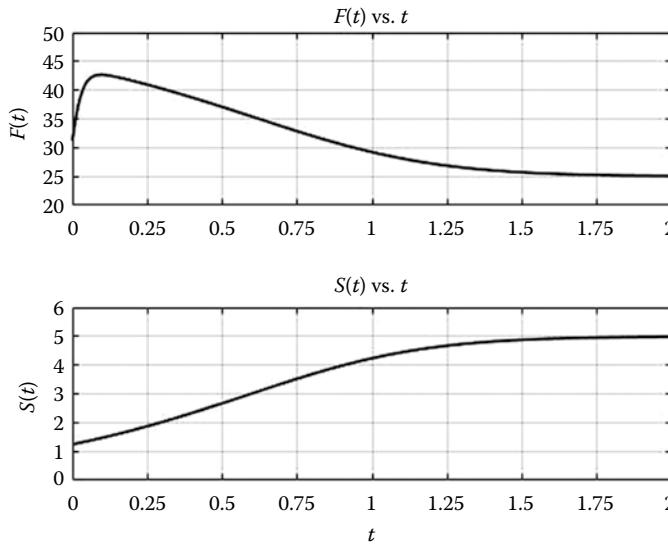


FIGURE 7.3 Transient response of ecosystem.

in agreement with the results of the “`trim`” function call. Note that the ordering of the state vector “`x`” must be known. This will be addressed later in Section 7.4. The small “`dx`” values assure the accuracy of the located equilibrium point.

The transient response and state trajectory produced by the Simulink “scope” and “XY Graph” starting from the point $(1.25F_e, 0.25S_e) = (31.25, 1.25)$ are shown in Figures 7.3 and 7.4.

The behavior of the system starting from 100 randomly selected points in a region including the equilibrium point is shown in Figure 7.5. It appears that the equilibrium point is indeed stable since all trajectories terminate there.

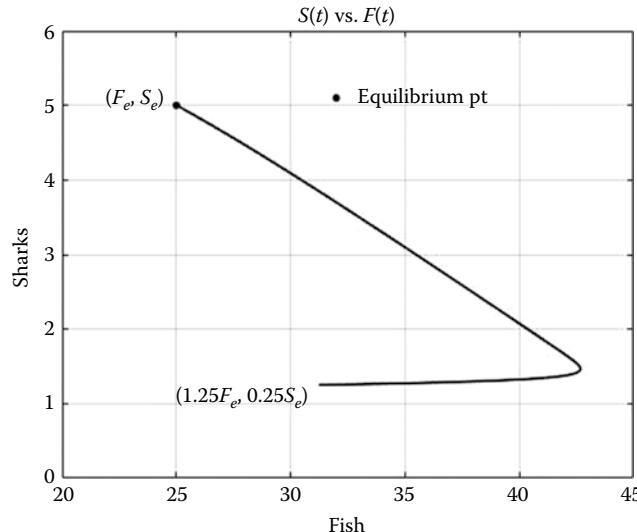


FIGURE 7.4 State trajectory of ecosystem.

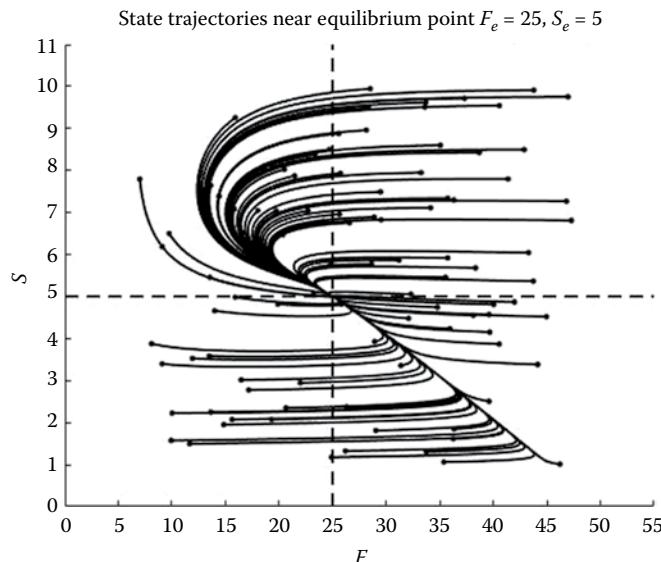


FIGURE 7.5 State trajectories demonstrating stability of equilibrium point.

7.2.2 EQUILIBRIUM POINT FOR A NONAUTONOMOUS SYSTEM

The “`trim`” function can be used to locate the steady state of a forced system subjected to a constant input(s). **Figure 7.6** is a simplified diagram of a transducer that converts a low-level acoustic pressure signal $p(t)$ to an output voltage $v_R(t)$. The movable membrane and fixed plate form a capacitor. The sound waves deflect the membrane changing the separation between it and the back plate and, therefore, the capacitance. A bias voltage is applied to produce an electrical charge on the membrane. The motion of the membrane is opposed by damping and elastic forces as well as an electrostatic force.

The mathematical model consists of the following differential and algebraic equations describing the circuit and the forces acting on the membrane:

$$R \frac{dQ}{dt} + v_C = E_0, \quad v_R = E_0 - v_C \quad (7.12)$$

$$Q = Cv_C \quad (7.13)$$

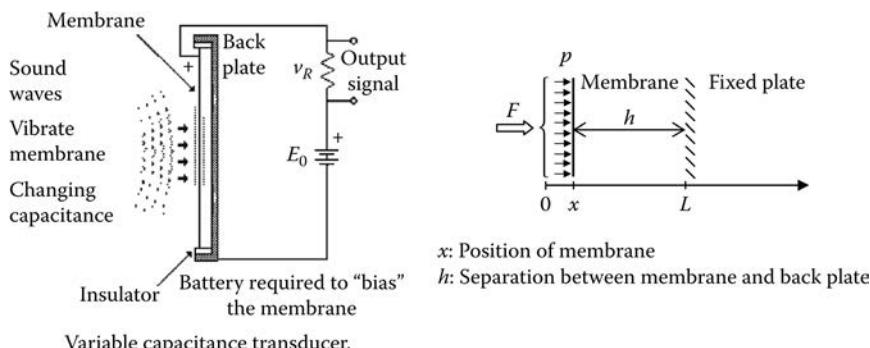


FIGURE 7.6 Variable capacitance transducer.

$$C = \frac{B}{h} = \frac{B}{L-x} \quad (7.14)$$

$$m \frac{d^2x}{dt^2} + \mu \frac{dx}{dt} + kx = -F_e + F \quad (7.15)$$

$$F_e = \frac{Q^2}{2B}, \quad F = pA \quad (7.16)$$

where

$Q = Q(t)$ is the electric charge on the capacitor (C)

$v_C = v_C(t)$ is the voltage across the capacitor (V)

$v_R = v_R(t)$ is the output voltage across the resistor (V)

$C = C(t)$ is the variable capacitance of the capacitor (F)

$h = h(t)$ is the separation between the movable membrane and back plate (mm)

$x = x(t)$ is the membrane displacement from equilibrium, that is, when $p = E_0 = 0$ (mm)

$F_e = F_e(t)$ is the electrostatic force on the membrane (N)

$F = F(t)$ is the force acting on the membrane due to pressure p (N)

$p = p(t)$ is the input acoustic pressure acting uniformly on the membrane (psi)

E_0 is the bias voltage on the capacitor (V)

m is the mass of membrane (g)

μ is the damping coefficient (N/(mm/s))

k is the elastic constant (N/mm)

Choosing the states and output

$$\begin{aligned} x_1 &= x, \quad x_2 = Q, \quad x_3 = \dot{x} \\ y_1 &= x, \quad y_2 = h, \quad y_3 = C, \quad y_4 = F_e, \quad y_5 = F, \quad y_6 = v_C, \quad y_7 = Q, \quad y_8 = v_R \end{aligned}$$

leads to the state equations (see Exercise 7.4)

$$\dot{x}_1 = x_3, \quad \dot{x}_2 = \frac{1}{BR}[-x_2(L-x_1) + BE_0], \quad \dot{x}_3 = \frac{1}{m}\left[-kx_1 - \frac{x_2^3}{2B} - \mu x_3 + Ap\right] \quad (7.17)$$

$$y_1 = x_1, \quad y_2 = L - x_1, \quad y_3 = \frac{B}{L-x_1}, \quad y_4 = \frac{x_2^2}{2B} \quad (7.18)$$

$$y_5 = Ap, \quad y_6 = \frac{x_2(L-x_1)}{B}, \quad y_7 = x_2, \quad y_8 = E_0 - \frac{x_2(L-x_1)}{B} \quad (7.19)$$

For constant inputs E_0 and $p(t) = p_0$, the equilibrium states are found by setting the state derivatives in Equation 7.17 to zero resulting in

$$x_{3,e} = 0, \quad -x_{2,e}(L-x_{1,e}) + BE_0 = 0, \quad -kx_{1,e} - \frac{x_{2,e}^2}{2B} + Ap_0 = 0 \quad (7.20)$$

Eliminating $x_{2,e}$ from the two equations yields a third-order polynomial in $x_{1,e}$.

$$kx_{1,e}^3 - (2kL + Ap_0)x_{1,e}^2 + L(kL + 2Ap_0)x_{1,e} + 0.5BE_0^2 - AL^2p_0 = 0 \quad (7.21)$$

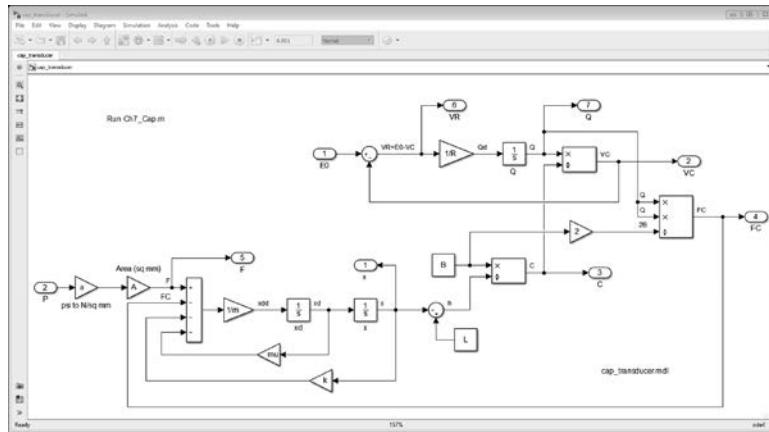


FIGURE 7.7 Simulink diagram of capacitive transducer for use by “Trim” function.

Equation 7.21 is solved in the M-file “Ch7_cap.m” using the following baseline parameter values:

$$\begin{aligned} A &= \pi (20\text{ mm})^2, L = 10\text{ mm}, m = 5\text{ g}, \mu = 0.01\text{ N}/(\text{mm/s}), \\ k &= 0.5\text{ N/mm}, R = 100\Omega, B = 5 \times 10^{-5}\text{ F-mm}, \\ E_0 &= 48\text{ V}, p_0 = 0.01\text{ psi} \end{aligned}$$

The single real root of Equation 7.21 is $x_{1,e} = x_e = 0.17208282239091\text{ mm}$. From the second of the equations in Equation 7.20, $x_{2,e} = Q_e = 2.442023021386376 \times 10^{-4}\text{ C}$.

A Simulink diagram of the system is shown in Figure 7.7. For reference by the “trim” function, the inputs are “E0” and “p0,” the states are “x,” “Q,” and “xd,” and the eight outputs are designated as shown. The “trim” function call in the M-file “Ch7_Cap.m” is

```
[x, u, y, dx] = trim ('cap_transducer', x0, u0, y0 ix, iu iy)
```

where the outputs “x,” “u,” and “y” are the computed equilibrium values of the state, input, and output vectors, respectively. The last argument “dx” is the state derivative vector that is identically zero at true equilibrium conditions. The input parameters “x0, u0, and y0” are used to set initial guesses for the equilibrium state, input, and output while the remaining arguments “ix, iu, and iy” serve to constrain selected components of the state, input, and output vectors at equilibrium.

Running script file “Ch7_Cap.m” produces the results shown in Table 7.1.

Note that the second input $u_{2,e} = 0.00006894413789$ is the equivalent of $p_0 = 0.001\text{ psi}$ converted to N/m^2 . The equilibrium state from the “trim” function call is in agreement with the solution to the equilibrium equations in Equation 7.20 obtained by running the M-file “Ch7_Cap.m.”

TABLE 7.1
“Trim” Function Results

$x_{1,e} = 0.17208282239053,$	$x_{2,e} = 0.00024420230214,$	$x_{3,e} = -0.000000000000000$
$u_{1,e} = 48.00000000000000,$	$u_{2,e} = 0.00006894413789$	
$y_{1,e} = 0.17208282239053,$	$y_{2,e} = 9.82791717760947,$	$y_{3,e} = 0.00000508754796,$
$y_{5,e} = 0.08663775883916,$	$y_{6,e} = 48.00000000001074,$	$y_{4,e} = 0.00059634764370,$
$d_{x1} = 10^{-7} \times$ $-0.000000000000000,$	$d_{x2} = 10^{-7} \times$ $-0.00000107363007,$	$d_{x3} = 10^{-7} \times$ -0.38882785879935

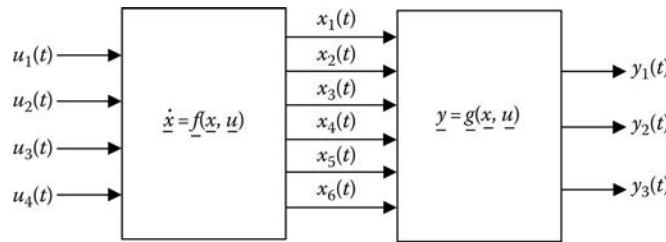


FIGURE 7.8 Dynamic system with equilibrium conditions specified for one input, two states, and one output.

A common use of the “trim” function is in applications where a subset of the equilibrium state and/or output vector is specified and the goal is to determine the input conditions resulting in the partially or fully specified equilibrium state. Figure 7.8 portrays a block diagram of a system with four inputs, six states, and three outputs.

Instead of specifying constants for inputs $u_1(t)$, $u_2(t)$, $u_3(t)$, and $u_4(t)$, to establish an equilibrium state $x_e = [x_{1,e} \ x_{2,e} \ x_{3,e} \ x_{4,e} \ x_{5,e} \ x_{6,e}]^T$ and output $y_e = [y_{1,e} \ y_{2,e} \ y_{3,e}]^T$, only input u_2 is fixed. Equilibrium levels of x_2 , x_6 , and y_1 are also fixed, and the steady-state equations of the system must be solved subject to these constraints. The solution will include the values for the nonconstrained inputs u_1 , u_3 , and u_4 along with the equilibrium values for the nonconstrained state and output components. Keep in mind that there may be no feasible solution or several solutions depending on the values assigned to the constrained variables.

To be more specific, consider an aircraft flying in level flight at a given altitude with constant speed, heading, and angle of attack. Certain constraints are imposed on the state vector of translational (longitudinal, lateral, and vertical) velocities (u , v , and w) and angular (roll, pitch, and yaw) velocities (r , p , and q) in a body reference coordinate system. The pilot wishes to know the throttle position and input settings that control the orientation of the control surfaces in order for the plane to achieve steady-state “trim” flight conditions.

The following example (Beltrami 1993) illustrates the point for an ecological system.

EXAMPLE 7.1

The growth rate of fish in a confined space at a fishery is modeled by

$$g(x) = \frac{u_R}{x} + rx\left(1 - \frac{x}{k}\right) - \varepsilon u_E \quad (7.22)$$

where

$x = x(t)$ is the density of fish measured in tons per square mile

Parameters r and k determine the natural growth rate function $rx(1 - x/k)$ of fish in the absence of external inputs related to harvesting and restocking

Input u_E is a measure of the effort (ships, gear, manpower, etc.) per year expended in harvesting

The parameter ε represents the efficiency of catching fish, measured as a fraction of each ton of fish caught per unit of effort

Finally, the first term accounts for restocking of fish with u_R , the restocking rate measured in tons of fish per square mile per year.

- Find the state derivative function and verify whether the given units for parameters and variables are consistent. In particular, determine the units of r , k , and ε .

- b. Find the equation relating the equilibrium state x_e and the constant inputs \bar{u}_R and \bar{u}_E where $\bar{u}_R = \bar{u}_E$, $t \geq 0$ and $u_E(t) = \bar{u}_E$, $t \geq 0$.
 Baseline numerical values of the system parameters are $k = 4$, $r = 2$, and $\varepsilon = 0.1$.
- c. Find x_e when $\bar{u}_R = 0.3$ tons = mi²/year and $\bar{u}_E = 10$ effort units/year.
- d. Repeat part (c) using the “trim” function. In addition, use x_e found in part (c) and fix $\bar{u}_R = 0.3$ to find the equilibrium value of u_E . Repeat using x_e found in part (c) and fix $\bar{u}_E = 10$ to find the equilibrium value of u_R .
- e. Change the numerical value of \bar{u}_E to 20 and show that there exist three real solutions for x_e .
- f. Verify the results in part (e) using the “trim” function with initial states $x_0 = 0, 2$, and 5.
- g. Show that the middle equilibrium point is unstable and the remaining two are stable. Verify the nature of the equilibrium points by simulation.

- a. The state derivative function is obtained from

$$g(x) = \frac{1}{x} \frac{dx}{dt} = \frac{u_R}{x} + rx \left(1 - \frac{x}{k}\right) - \varepsilon u_E \quad (7.23)$$

$$\Rightarrow f(x, u_R, u_E) = \frac{dx}{dt} = u_R + rx^2 \left(1 - \frac{x}{k}\right) - \varepsilon u_E x \quad (7.24)$$

The units for each term in Equation 7.24 are tons/mi²/year, that is,

$$\begin{aligned} \frac{\text{tons}/\text{mi}^2}{\text{year}} &= \frac{\text{tons}/\text{mi}^2}{\text{year}} + \left(\frac{1/\text{year}}{\text{tons}/\text{mi}^2}\right) \left(\frac{\text{tons}}{\text{mi}^2}\right)^2 \\ &\quad - \left(\frac{1}{\text{effort}}\right) \left(\frac{\text{effort}}{\text{year}}\right) (\text{tons}/\text{mi}^2) \end{aligned}$$

The units for r , k , and g are (1/year)/(tons/mi²), tons/(mi²), and 1/effort, respectively.

- b. Setting the state derivative function in Equation 7.24 to zero gives

$$\bar{u}_R + rx_e^2 \left(1 - \frac{x_e}{k}\right) - \varepsilon \bar{u}_E x_e = 0 \quad (7.25)$$

- c. Substituting the given values for r , k , ε , \bar{u}_R , and \bar{u}_E into Equation 7.25 produces a cubic polynomial in x_e . The M-file “Ch7_Ex7_1.m” employs the “roots” function to find the roots. The results are 3.4740, 0.2630 $\pm j0.3218$.

- d. “Ch7_Ex7_1.m” contains the statements

```
% A.1 Given uR_bar and uE_bar, find xe
uR_bar = 0.3; uE_bar = 10;
x0 = 10; ix = [];
u0 = [uR_bar;uE_bar];
iu = [1,2]; y0 = 0; iy = [];
[x, u, y, dx] = trim('fishery_1', x0, u0, y0, ix, iu, iy)
```

where “fishery_1” is the Simulink file name of the model shown in Figure 7.9. Note that “iu = [1, 2]” constrains the inputs to $\bar{u}_R = 3$ and $\bar{u}_E = 10$. The results are given in Table 7.2. The second part of part (d) is implemented using the following statements (see Table 7.2 for results):

```
% A.2 Given uR_bar and xe, find uE_bar
uR_bar = 0.3;
x0 = 3.4740;
```

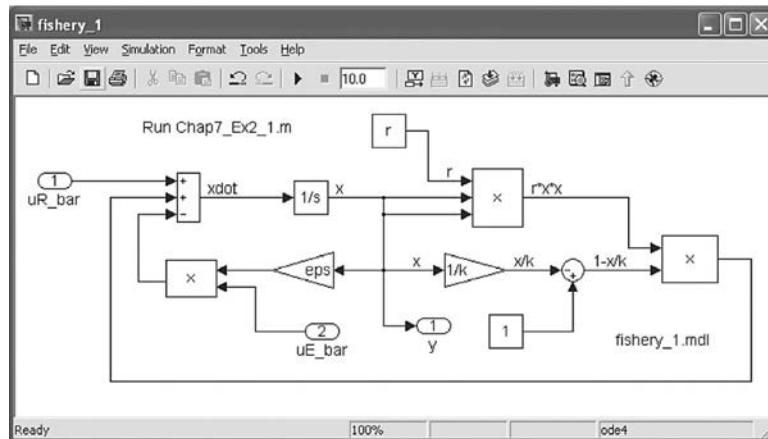


FIGURE 7.9 Simulink diagram of fishery system dynamics.

TABLE 7.2
Results of Using “trim” Function for Different Conditions

Case	Given	Given	Initial Guess	Result	“xd”
A.1	$\bar{u}_R = 0.3$	$\bar{u}_E = 10$	$x_0 = 10$	$x_e = 3.4740$	-2.8234×10^{-8}
A.2	$\bar{u}_R = 0.3$	$x_e = 3.4740$	$x_0 = 10$	$\bar{u}_E = 10$	5.1090×10^{-11}
A.3	$\bar{u}_E = 10$	$x_e = 3.4740$	$x_0 = 10$	$\bar{u}_R = 0.3$	1.2261×10^{-12}
B.1	$\bar{u}_R = 0.3$	$\bar{u}_E = 20$	$x_0 = 0$	$x_e = 0.1814$	2.1407×10^{-12}
B.2	$\bar{u}_R = 0.3$	$\bar{u}_E = 20$	$x_0 = 2$	$x_e = 1.3278$	0
B.3	$\bar{u}_R = 0.3$	$\bar{u}_E = 20$	$x_0 = 5$	$x_e = 2.4908$	-3.3323×10^{-10}

```
ix = 1;
u0 = [uR_bar;0];
iu = 1;
y0 = 0; iy = [];
[x, u, y, dx] = trim("fishery_1", x0, u0, y0, ix, iu, iy)
```

The third part of part (d) is accomplished in a similar fashion except that \bar{u}_E and x_e are fixed and \bar{u}_R is returned by the “trim” function. In this case, the assignments “ $ix = 1$ ” and “ $iu = 1$ ” fix $x_e = 3.4740$ and $u_R = \bar{u}_R$.

- e. When $\bar{u}_E = 20$, the “roots” function in “Ch7_Ex7_1.m” yields three solutions to Equation 7.25, namely, 0.1814, 1.3278, and 2.4908.
- f. Using the “trim” function with initial state guesses of 0, 2, and 5 produces the identical equilibrium states (see Cases B.1, B.2, and B.3 in Table 7.2).
- g. The stability of each equilibrium point can be ascertained by looking at a graph of the growth rate function shown in Figure 7.10. Note that the fish density increases wherever $g(x)$ is positive as indicated by right-pointing arrows and conversely decreases in regions where $g(x)$ is negative, shown with left-pointing arrows. Fish densities initially located in the region $(x_e)_1 < x < (x_e)_2$ will move towards $(x_e)_1 = 0.1814$ whereas initial densities satisfying $(x_e)_2 < x < (x_e)_3$ eventually approach $(x_e)_3 = 2.4908$. Consequently, the equilibrium point $(x_e)_2 = 1.3278$ is unstable.

The system is simulated using “fishery_2.mdl” (not shown) that is identical to “fishery_1.mdl” except for the input blocks that are replaced by “constant” blocks and the “output” block is removed. Results are shown in Figure 7.11.

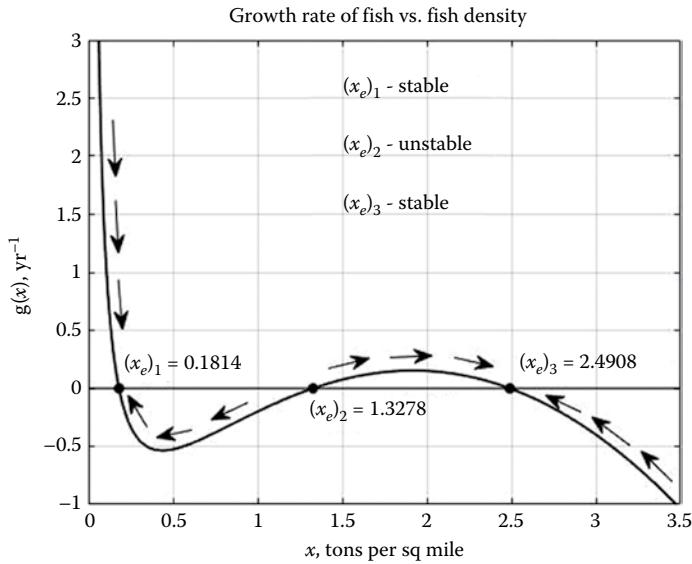


FIGURE 7.10 Graph of fish growth rate and equilibrium points.

Note that the middle responses in the second graph starting at $x_0 = 1.3$, slightly less than $(x_e)_2 = 1.3278$ and $x_0 = 1.35$ and slightly more than $(x_e)_2 = 1.3278$, diverge from the neighborhood of $(x_e)_2$, the unstable equilibrium point.

Before we proceed further, it is interesting to consider the market place's influence on the fish supply. If we adopt a very rudimentary model for the harvesting effort u_E , one that says the rate of change of harvesting depends solely on net profit as measured by the difference between revenue and cost, then $u_E(t)$ is governed by the first-order differential equation

$$\frac{du_E}{dt} = \alpha(R - C) \quad (7.26)$$

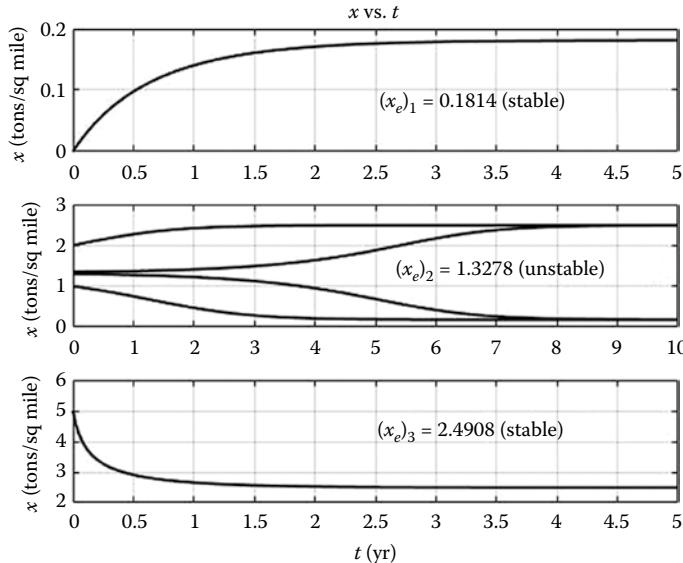


FIGURE 7.11 State responses starting from several initial points.

where

R and C are the revenue and cost, respectively, in \$/year/mi²
 α is a constant

Assuming revenue depends on harvesting $\varepsilon u_E x$ and selling price p leads to

$$R = \varepsilon u_E x \cdot p \quad (7.27)$$

Where p is the selling price in \$/ton. The cost is a function of effort and restocking, that is,

$$C = c_E u_E + c_R u_R \quad (7.28)$$

where

c_E is in \$/effort/year/mi²
 c_R is in \$/ton

Equations 7.26 through 7.28 give

$$\frac{du_E}{dx} = \alpha[p\varepsilon u_E x - (c_E u_E + c_R u_R)] \quad (7.29)$$

The expanded system dynamics are now modeled by the coupled nonlinear differential equations given in Equations 7.24 and 7.29. A block diagram of the system displaying the system parameters, input, states, and outputs is shown in Figure 7.12.

System parameters r, k, \dots, c_R can be thought of as inputs to the system. However, they are distinguished from the system input u_R because they generally remain fixed at assigned values. When they do vary, fluctuations (often unpredictable) occur with less frequency than the input.

Suppose the set of parameters in Figure 7.12 are fixed at baseline values except for the selling price p . Viewing the system as being “driven” by the conventional input restocking rate u_R as well as p , we can employ the “trim” function to search for the equilibrium state and output vectors. The M-file “Ch7_fishery_w_economics.m” uses $k = 4$, $r = 2$, $\varepsilon = 0.1$, $\alpha = 0.75$, $c_E = 2$, and $c_R = 1$ for the fixed parameters and varies u_R and p as inputs in the “trim” function call. The results are shown in Table 7.3.

Certain combinations of \bar{u}_R and \bar{p} produce no solution, which raises the question of whether there may in fact be other equilibrium states in addition to the one given in Table 7.3 for the combinations considered. There is no certainty when using a search algorithm. All we can do is begin the search from different starting points in the hope of finding additional equilibrium states, should they exist. The results in Table 7.3 were obtained using a starting guess of $x = 10$ and $u_E = 20$.

We now explore the possibility of the existence of an analytical solution for the equilibrium state. The algebraic equations resulting from setting the state derivative functions in Equations 7.24 and 7.29 to zero are

$$0 = \bar{u}_R + rx_e^2 \left(1 - \frac{x_e}{k} \right) - \varepsilon(u_E)_e x_e \quad (7.30)$$

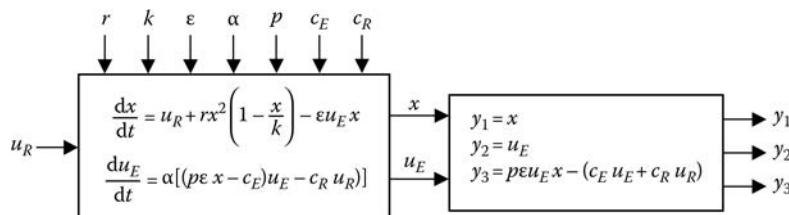


FIGURE 7.12 Block diagram of system showing parameters, input, states, and output.

TABLE 7.3
Equilibrium States x_e and $(u_E)_e$ as a Function of Input (\bar{u}_R, \bar{p})

\bar{u}_R	\bar{p}		
	6	7	8
0.3	$x_e = 3.3772$ $(u_E)_e = 11.4053$	$x_e = 2.8821$ $(u_E)_e = 17.1501$	$x_e = 2.5189$ $(u_E)_e = 19.8447$
0.5	$x_e = 3.4052$ $(u_E)_e = 11.5954$	$x_e = 2.8975$ $(u_E)_e = 17.6981$	$x_e = 2.5304$ $(u_E)_e = 20.5694$
1	$x_e = 3.4716$ $(u_E)_e = 12.0522$	$x_e = 2.9321$ $(u_E)_e = 19.0668$	$x_e = 2.5559$ $(u_E)_e = 22.3675$

$$0 = \bar{p}\varepsilon x_e - c_E(u_E)_e - c_R \bar{u}_R \quad (7.31)$$

Solving for x_e in Equation 7.31 gives

$$x_e = \frac{1}{\bar{p}\varepsilon(u_E)_e} [c_E(u_E)_e + c_R \bar{u}_R] \quad (7.32)$$

and substituting the result for x_e in Equation 7.30 produces a fourth-order polynomial in $(u_E)_e$. The details are left for an exercise problem; however, the result is

$$\beta_4(u_E)_e^4 + \beta_3(u_E)_e^3 + \beta_2(u_E)_e^2 + \beta_1(u_E)_e + \beta_0 = 0 \quad (7.33)$$

where

$$\begin{aligned} \beta_4 &= -k\bar{p}\varepsilon^3 c_E \\ \beta_3 &= -k\bar{p}^2\varepsilon^3 \bar{u}_R (\bar{p} - c_R) + r c_E^2 (k\bar{p}\varepsilon - c_E) \\ \beta_2 &= r c_E c_R \bar{u}_R (2k\bar{p}\varepsilon - 3c_E) \\ \beta_1 &= -r(c_R \bar{u}_R)^2 (k\bar{p}\varepsilon - 3c_E) \\ \beta_0 &= -r(c_R \bar{u}_R)^3 \end{aligned} \quad (7.34)$$

For $\bar{u}_R = 0.3$ and $\bar{p} = 6$, the solutions from “Ch7_fishery_w_economics.m” are

$$\begin{aligned} (u_E)_e &= 11.4053, 0.7500, -0.1471 \pm j0.0168 \\ x_e &= 3.3772, 4.0000, -0.0219 \pm j0.3842 \end{aligned}$$

There are two feasible solutions, namely, $x_e = 3.3772$, $(u_E)_e = 11.4053$ and $x_e = 4.0000$, $(u_E)_e = 0.7500$. The “trim” function has converged to the first solution (see Table 7.3).

The values shown in Table 7.3 can be verified by simulation. For example, Figure 7.13 is a simulation of the system initially at equilibrium with inputs $u_R = \bar{u}_R = 0.3$, $p = \bar{p} = 6$, and $x_e = 3.3772$, $(u_E)_e = 11.4053$. Step changes in u_R and p occur at $t = 1$ year. The new inputs correspond to the lower right corner of Table 7.3, namely, $u_R = \bar{u}_R = 1$, $p = \bar{p} = 8$. The new equilibrium state agrees with the values shown in the table. Refer to M-file “Ch7_Fig2_12.m.”

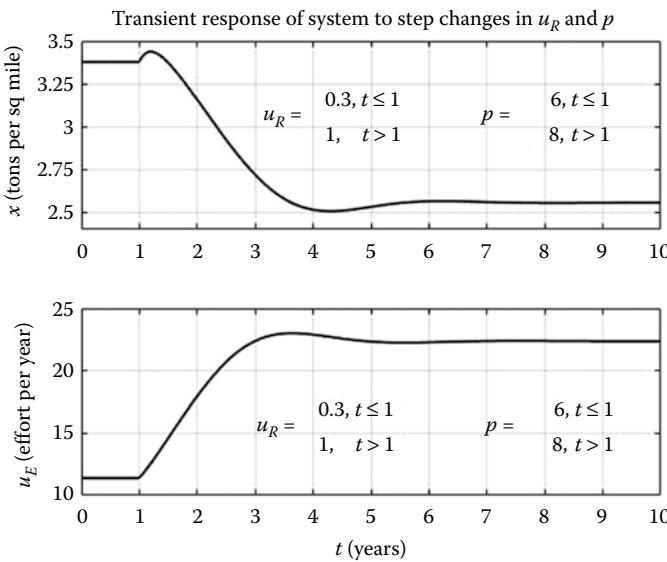


FIGURE 7.13 Simulated transient response to step changes in u_R and p .

EXERCISES

7.1 An alternate predator-prey model for fish and sharks is

$$\frac{dF}{dt} = aF - bSF$$

$$\frac{dS}{dt} = -cS + dFS$$

- Find the nontrivial equilibrium point (F_e, S_e) in terms of parameters a, b, c , and d .
 - Use the “trim” function to find the equilibrium point when the system parameters are $a = 0.1$, $b = 0.03$, $c = 0.02$, and $d = 0.0025$, and compare the answer with the value obtained using the result in part (a).
 - Verify by simulation that the equilibrium point (F_e, S_e) is “neutrally stable,” which means that sustained oscillations in fish and shark populations occur regardless of the initial conditions $F(0) \neq 0$ and $S(0) \neq 0$. Plot time histories $F(t)$ and $S(t)$, $t \geq 0$ and a phase plot S vs. F .
- 7.2 For the predator-prey system governed by Equations 7.5 and 7.6,
- Show that the points (F_e, S_e) in Table E7.2 are equilibrium points.

TABLE E7.2

F_e	S_e
0	0
$\frac{a}{b}$	0
$A\lambda$	ae
$b\lambda + ce$	$b\lambda + ce$

- Investigate the local stability of each equilibrium point by simulation of the system with initial conditions in the neighborhood of each point. Draw the phase trajectories for each case.

- c. The system parameters are $a = 50$, $b = 2.5$, $c = 4$, $e = 2$, and $\lambda = 8$. Use the “trim” function starting at different points in the F – S plane to try and locate the last two equilibrium points.
 - d. The system parameters are $a = 40$, $b = 4$, $c = 3$, $e = 2$, and $\lambda = 5$. Use the “trim” function with S constrained to zero to find the equilibrium point $(a/b, 0)$.
 - e. The system parameters are $a = 50$, $b = 2.5$, $c = 0$, $e = 2$, and $\lambda = 8$. Simulate the system and obtain time histories of $F(t)$ and $S(t)$ along with a phase trajectory when the initial populations are $F(0) = 5$ and $S(0) = 2$.
- 7.3 A three-species predator–prey model (Edelstein-Keshet 1988) is

$$\frac{dx}{dt} = axz + \beta xy - yx$$

$$\frac{dy}{dt} = \delta y - \varepsilon xy$$

$$\frac{dz}{dt} = \mu z(\nu - z) - \lambda xy$$

where

x is a predator

y and z are its prey

- a. Express the nontrivial equilibrium pt (x_e, y_e, z_e) in terms of the system parameters.
 - b. Use the “trim” function to find the equilibrium pt when the system parameters are $\alpha = 0.075$, $\beta = 0.009$, $\gamma = 0.2$, $\delta = 0.1$, $\varepsilon = 0.025$, $\mu = 0.0015$, $\nu = 10$, and $\lambda = 0.003$. Compare the answer with the value obtained using the result in part (a).
 - c. The equilibrium point is asymptotically stable. Verify by simulating the response of the autonomous system starting from various randomly selected points in the neighborhood of the equilibrium point.
- 7.4 Derive the state Equations 7.17–7.19.

7.3 OPTIMIZATION OF SIMULINK MODELS

System designers often resort to simulation to verify whether a newly designed system performs in a manner consistent with a set of predefined requirements and constraints. Generally speaking, multiple simulations are necessary to “observe” how the system responds to a range of inputs and parameter variations. For some systems, a subset of the inputs and parameters that affect its performance are controllable. For example, ground vehicle performance can be characterized by fuel economy, vehicle handling, ride comfort, acceleration, emergency braking, and so forth. Given a single unambiguous measure of system performance, the design objective reduces to a determination of numerical values for the controllable system parameters (wheel base, springs and shocks, carburetor design, weight, steering ratio, and so forth) resulting in optimal performance. In contrast, a simulation model used to predict weather relies on knowledge of atmospheric conditions to forecast future weather patterns. Neither inputs (at least not yet) nor the resulting weather is controllable.

Inherent in the process of optimizing system performance is the ability to observe or, somehow, measure performance, that is, acquire data about the system as the parameters are varied. Experimenting with the real system is oftentimes impractical for reasons of expense and time consumption or even dangerous depending on the levels of the system parameters. Herein lies the value of simulation in optimizing a system’s performance. The simulation model can be “exercised” in a systematic way to achieve optimum or near optimum results without the previously cited pitfalls of dealing with the physical system. A simple example of system optimization follows.

Figure 7.14 portrays a pair of objects, one designated the target and the other object intent on destroying it by firing a projectile weapon at it. The target is assumed to be a point moving at constant velocity v_T in a circular trajectory of radius L with the attacker permanently positioned at the center of the coordinate system. The attacker fires its weapon along a fixed direction denoted by the azimuth angle θ . The projectile is subjected to a linear drag force. Both objects are assumed to be at the same elevation for the entire time. The system is similar in nature to a surface vessel under attack by a torpedo fired from a submarine.

The mathematical model begins with equations describing the trajectories of the target and projectile. The angular velocity of the target is

$$\dot{\varphi} = \frac{v_T}{L} \quad (7.35)$$

The x and y coordinates of the target are related to the increasing angle φ (measured with respect to the y -axis) according to

$$x_T = L \sin \varphi, y_T = L \cos \varphi \quad (7.36)$$

The projectile's motion is governed by

$$m\ddot{v}_p + \mu v_p = 0 \quad (7.37)$$

where

m is the projectile mass

μ is the drag coefficient for determining the linear drag force acting on the projectile

Resolving the projectile's velocity into x and y coordinates,

$$\dot{x}_p = v_p \cos \theta, \dot{y}_p = v_p \sin \theta \quad (7.38)$$

The distance separating the target and projectile is given by

$$D = [(x_T - x_p)^2 + (y_T - y_p)^2]^{1/2} \quad (7.39)$$

A Simulink diagram incorporating Equations 7.35–7.39 is shown in **Figure 7.15**. Since the intended purpose of firing the projectile is to intercept the target, the performance measure of the system is taken as the separation between the target and projectile at the moment the projectile

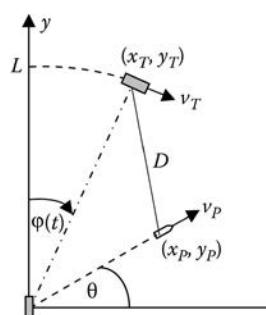


FIGURE 7.14 Diagram showing movement and position of target and projectile.

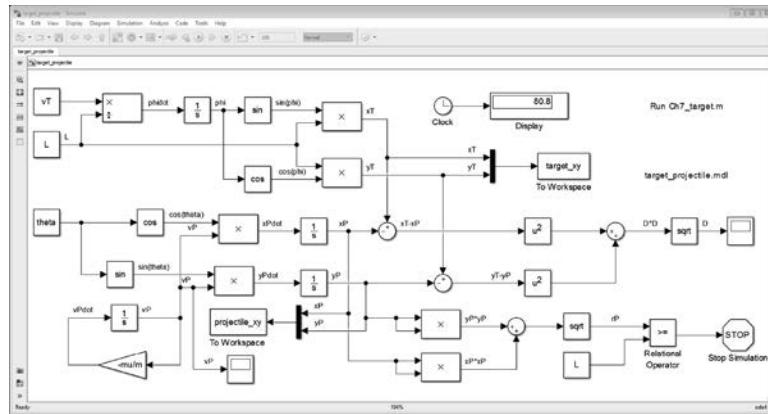


FIGURE 7.15 Simulink diagram of target and projectile system.

has traveled a distance L . This distance is denoted as D_{final} . Note the presence of a “Relational Operator” block for terminating the simulation when the projectile’s distance “ xP ” exceeds “ L .” The simulation final time is chosen as some arbitrarily large number ensuring that the simulation is halted at the appropriate time, which incidentally is monitored in the “Display” block.

The firing angle θ is treated as a controllable parameter. Our objective is to find θ_{opt} , that is, the projectile firing angle that minimizes the performance measure D_{final} (ideally to zero). A number of calls are made from the M-file “*Ch7_target.m*” to the simulation model “*target_projectile.mdl*” to explore the relationship between D_{final} and θ . The result is shown in [Figure 7.16](#).

The function $D_{\text{final}}(\theta)$ is seen to possess a single minimum in the neighborhood of 70° when the remaining system parameter values are as shown in [Figure 7.16](#). We must perform a search for θ_{opt} where

$$D_{\text{final}}(\theta_{\text{opt}}) = \underset{\theta \geq 0}{\text{Min}} D_{\text{final}}(\theta) \quad (7.40)$$

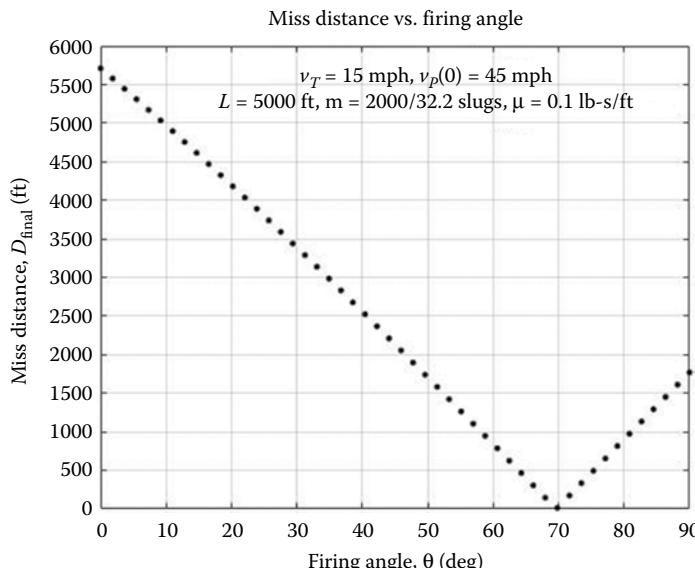


FIGURE 7.16 Graph of miss distance D_{final} vs. projectile firing angle θ .

The M-file “*Ch7_opt_search.m*” performs a very rudimentary search for the optimum angle θ_{opt} . It begins by incrementing θ (starting from zero) until it finds an angle θ_U where $D_{\text{final}}(\theta_U)$ is greater than D_{final} at the previous firing angle. Since the previous point could be to the right of the minimum, the angle prior to the previous one is designated θ_L and the interval ($\theta_L \leq \theta \leq \theta_U$) is guaranteed to contain θ_{opt} . A second sweep, with a finer gradation of θ values, is initiated, beginning at θ_L . It continues until θ_{opt} is found or the entire interval ($\theta_L \leq \theta \leq \theta_U$) is traversed. θ_{opt} is detected when $D_{\text{final}}(\theta)$ is below some threshold, 10 ft in this case. A second sweep is tried with even finer divisions if the first one is unsuccessful. It must be borne in mind that each value of θ requires a simulation run to find $D_{\text{final}}(\theta)$. The two search phases are illustrated in Figure 7.17.

The optimization toolbox includes a number of algorithms for iteratively searching parameter spaces to locate local minima and maxima of a function that depends on the parameters. Optimum seeking methods are available for both unconstrained and constrained optimization. The optimization toolbox and Simulink complement each other when the performance measure (objective function in optimization terminology) at some point in the parameter space depends on the dynamic response of a system. That is, the actual system response must be observed or simulated to obtain a numerical value of the objective function. This could be a final value of some output (dependent variable) or perhaps a certain function of several dependent variables. A common situation is where the objective function is evaluated as the integral of an appropriate function of the system’s outputs.

In situations where the objective function dependence on the system’s parameters is expressible in analytical or tabular form, a dynamic simulation is unnecessary and Simulink is not required. In either case, a unique value for the objective function at different locations in the parameter space must be available to the optimization routine.

Before we delve more into the practical aspects of optimization, let us take a look at how the MATLAB optimization toolbox can be used to find the optimum firing angle θ_{opt} in the previous example. The first step is the creation of a MATLAB function file to evaluate the objective function D_{final} for a given value of firing angle θ . The function M-file “*obj_fcn_D*” is listed as follows:

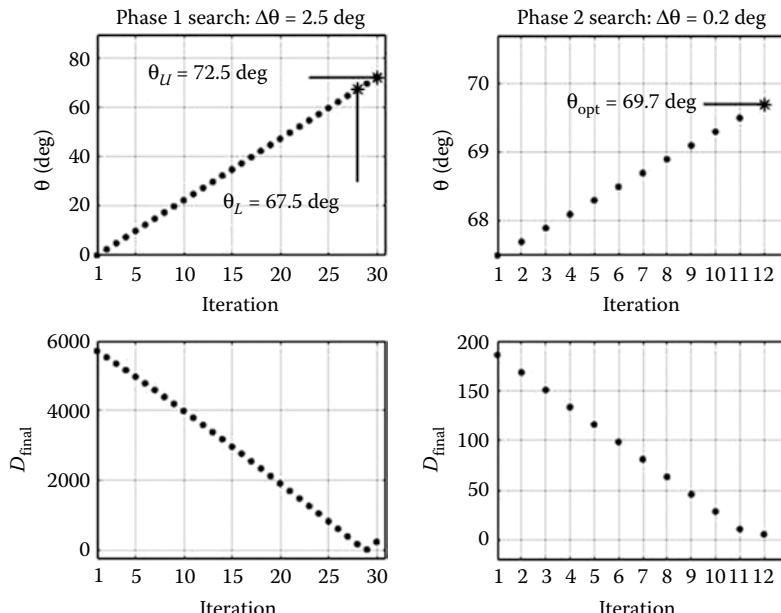


FIGURE 7.17 Two-phase search for optimum firing angle.

```

function f = obj_fcn_D (angle, L, m, mu, vT, vP)
% Objective function for finding D_final
T = 0.05; % integration step (sec) for RK-4
tfinal = 200; % final sim time (sec)
opts = simset ('SrcWorkSpace', 'current', 'DstWorkSpace', 'current'); theta
= angle; % firing angle (rad) for 'CON' Simulink model block sim('target_
projectile', tfinal, opts); %run sim and return array D
f = D(end); % objective function: D_final (ft)

```

The first argument of “*obj_fcn_D.m*” is “angle” (the optimization parameter θ), and the remaining arguments are simply parameters passed to the function from the main program “*Ch7_Toolbox_opt_search.m*.” The main program initializes the starting value of θ in the variable “*angle_init*” and then calls the optimization toolbox function “*fminunc*” to start the search θ_{opt} . The preferred way of calling “*fminunc*” depends on the version of MATLAB in use. Prior to MATLAB 6.0 (R12), the correct syntax was

```
[opt_angle_rad, FVAL] = fminunc ('obj_fcn_D', angle_init, [], L, m, mu, vT,
vP_initial) % optimum angle (rad)
```

For MATLAB 6.0 (R12) and later, the string ‘*obj_fcn_D*’ was replaced by the function handle ‘@*obj_fcn_D*’ for faster calls to the objective function.

θ_{opt} and the minimum D_{final} are returned in “*opt_angle_rad*” and “*FVAL*” if the search algorithm converges to a solution. “*Ch7_Toolbox_opt_search.m*” contains additional statements to simulate the system using the optimum firing angles returned by “*fminunc*” when the target speed is 15 and 75 mph. Target and projectile trajectories are shown in [Figures 7.18](#) and [7.19](#). The projectile’s position is plotted at 2.5 s intervals. The elapsed time in both cases is 80.8 s.

It is not surprising that elapsed time t_f is the same for both target speeds. The time is easily found by recognizing that the solution to Equation 7.37 is given by

$$v_p(t) = v_p(0)e^{-(\mu/m)t}, \quad t \geq 0 \quad (7.41)$$

and then integrating to obtain $s_p(t)$, the distance traveled by the projectile

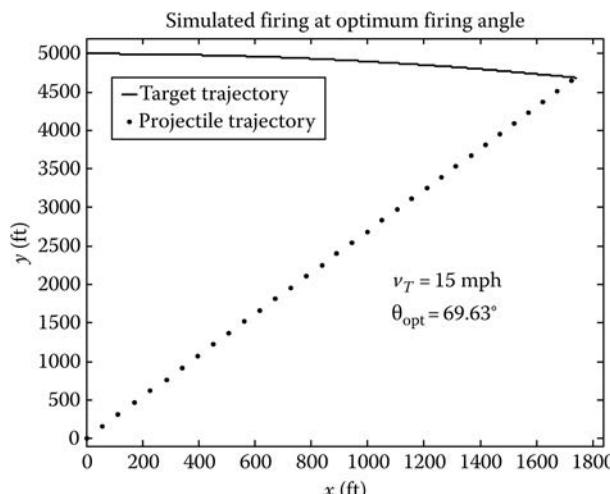


FIGURE 7.18 Target and projectile motion for optimum firing angle ($v_T = 15$ mph).

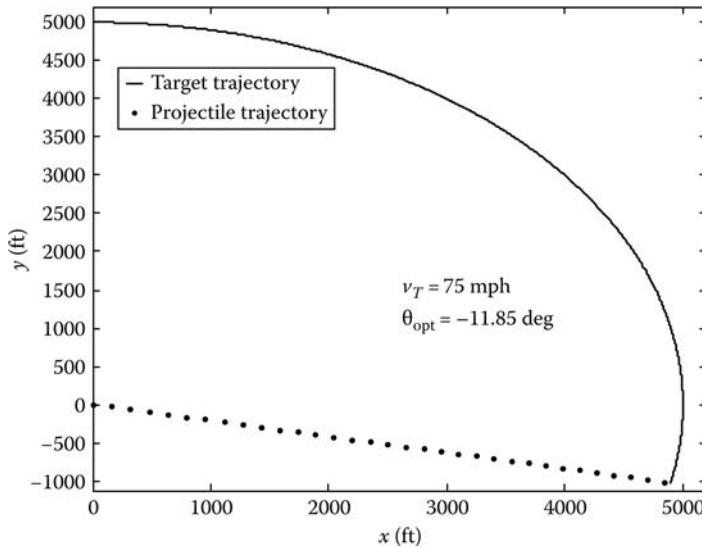


FIGURE 7.19 Target and projectile motion for optimum firing angle ($v_T = 75$ mph).

$$s_p(t) = \int_0^t v_p(0) e^{-(\mu/m)\tau} d\tau \quad (7.42)$$

$$= \frac{m}{\mu} v_p(0) [1 - e^{-(\mu/m)t}], \quad t \geq 0 \quad (7.43)$$

Setting $s_p(t_f) = L$ and solving for t_f give

$$\begin{aligned} t_f &= -\frac{m}{\mu} \ln \left[1 - \frac{\mu L}{mv_p(0)} \right] \\ &= -\frac{2000/32.3}{0.1} \ln \left[1 - \frac{0.1(5000)}{(2000/32.2)45(5280/3600)} \right] = 80.79 \text{ s} \end{aligned} \quad (7.44)$$

There are a number of system parameters that can be varied to study their effect on the optimum firing angle. Suppose we wish to investigate the relationship between θ_{opt} and the target's speed v_T . The M-file “*Ch7_opt_theta_vT.m*” sequences through a range of target speeds $v_T = 5, 15, 25, \dots, 65, 75$ mph and finds the optimum firing angle for each speed. The result is shown in Figure 7.20 where it is apparent that the relationship is linear. Could this have been predicted?

Referring to Figure 7.14, at time t_f , when the projectile has struck the target,

$$\theta = \frac{\pi}{2} - j(t_f) = \frac{\pi}{2} - \left(\frac{v_T}{L} \right) t_f \quad (v_T \text{ in ft/s, } \theta \text{ in rad}) \quad (7.45)$$

$$\Rightarrow \theta = 90 - \left[\frac{v_T}{5000} \left(\frac{5280}{3600} \right) \left(\frac{180}{\pi} \right) \right] 80.79 \quad (v_T \text{ in mph, } \theta \text{ in deg}) \quad (7.46)$$

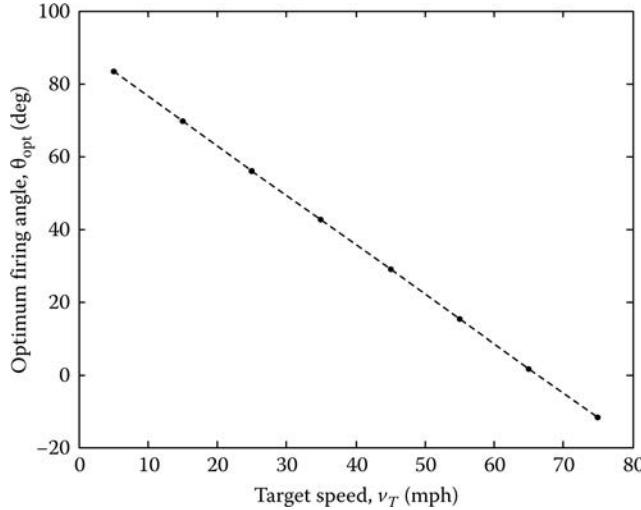


FIGURE 7.20 Results of target speed sensitivity analysis.

$$\Rightarrow \theta = 90 - 1.3579v_T \quad (7.47)$$

which is the equation of the line shown in [Figure 7.20](#).

The search algorithm used to find the optimum firing angle, illustrated in [Figure 7.17](#), is rather simple. More sophisticated algorithms rely on the local topography of the objective function to guide the search for the local optimum point. The gradient vector (to be defined shortly) is computed at a point in multidimensional parameter space and used to arrive at a new direction and distance for continuing the search. The gradient vector reduces to the first derivative for one-dimensional searches.

Following is an example of a one-dimensional parameter search using the slope, that is, first derivative to locate the minimum of an objective function. In the target-projectile system, the projectile decelerates with time due to the linear drag force. Suppose the target attempts to “outrun” the projectile by traveling in the y -direction starting from the point $(0, L)$, (see [Figure 7.14](#)) at constant speed v_T . The target is in the clear if the pursuing projectile is moving slower than the target, that is, $v_p(t) < v_T$ at some point in time and $y_T(t) \geq y_p(t)$ have been true up to that time.

We focus on the minimum separation between the target and projectile to see how close the two come. For a set of fixed parameters L , m , μ , v_T and $v_p(0) \geq v_T$, the time at which the minimum separation occurs is required. Position of the target is given by

$$y_T(t) = L + v_T t, \quad t \geq 0 \quad (7.48)$$

From Equation 7.42, the distance traveled and, hence, position of the projectile are

$$y_P(t) = \tau v_p(0)(1 - e^{-t/\tau}), \quad t \geq 0 \quad \text{where } \tau = \frac{m}{\mu} \quad (7.49)$$

The separation between the target and projectile as a function of time is

$$D(t) = y_T(t) - y_P(t) = L + v_T t - \tau v_p(0)(1 - e^{-t/\tau}), \quad t \geq 0 \quad (7.50)$$

A graph of Equation 7.50 with the nominal parameter values is shown in [Figure 7.21](#). Differentiating $D(t)$ in Equation 7.50 gives

$$\frac{d}{dt} D(t) = v_T - v_P(0)e^{-t/\tau}, \quad t \geq 0 \quad (7.51)$$

A search algorithm is implemented in “*Ch7_min_sep_search.m*,” which sequences through values of t based on the first derivative. Specifically,

$$t_{i+1} = t_i - \frac{d}{dt} D(t_i) \cdot \Delta, \quad i = 0, 1, 2, \dots \quad (7.52)$$

The search terminates when the magnitude of the derivative falls below a threshold that was set to 0.1 and Δ was fixed at a value of 10.

[Figure 7.22](#) shows the results of two searches for the minimum separation. The one on the left starts at $t_0 = 0$ s and the other begins at $t_0 = 800$ s. The two searches quickly locate the same minimum separation of 572.4 ft at $t = 349.7$ s. Note the derivative function approaching zero from opposite directions as the two searches progress.

This example illustrates the power of using the derivative to scale the step size between sequential points in the search for the optimum value of the objective function.

From elementary calculus, the local minima and maxima of a continuously differentiable function occur at critical points where the first derivative is equal to zero. Hence, from Equation 7.51, we can find t_{\min} as follows

$$\begin{aligned} \left. \frac{d}{dt} D(t) \right|_{t=t_{\min}} &= v_T - v_P(0)e^{-t/\tau} \Big|_{t=t_{\min}} = 0 \\ \Rightarrow t_{\min} &= -\tau \ln \left(\frac{v_T}{v_P(0)} \right) = -\frac{2000/32.2}{0.15} \ln \left(\frac{15}{35} \right) = 350.8 \text{ s} \end{aligned} \quad (7.53)$$

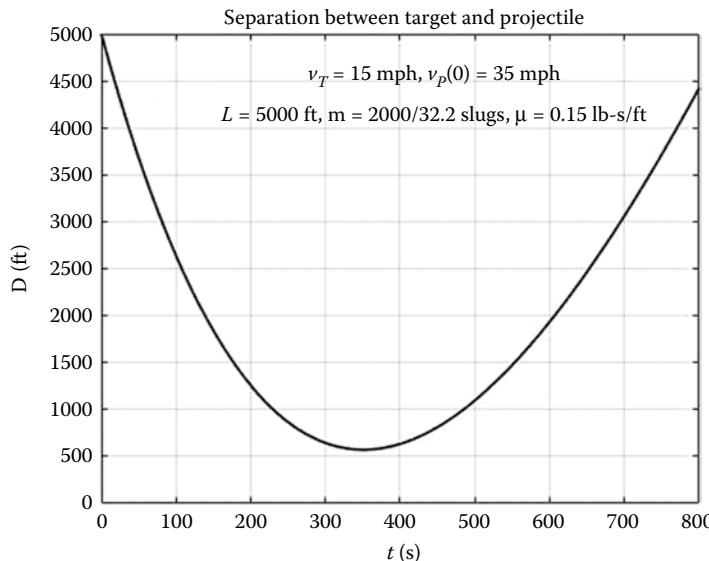


FIGURE 7.21 Graph of separation distance vs. time.

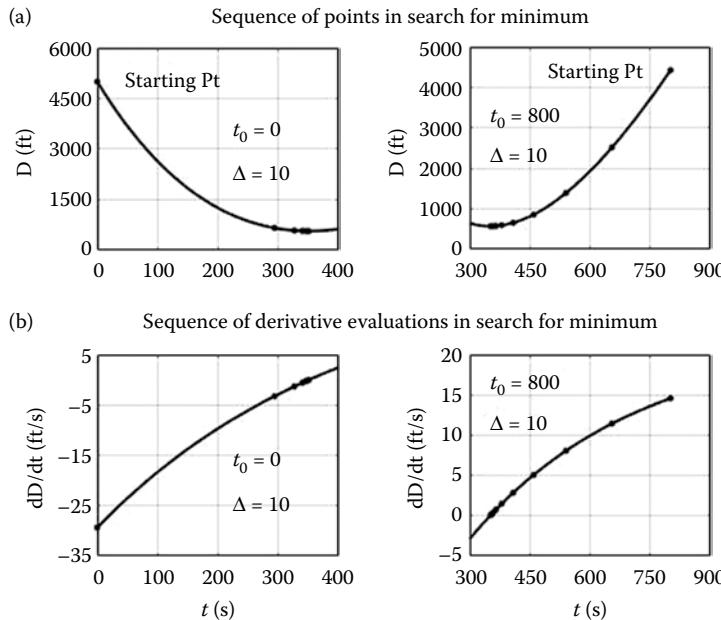


FIGURE 7.22 Results of two searches for minimum separation. (a) Sequence of points in search for minimum. (b) Sequence of derivative evaluations in search for minimum.

Substituting t_{\min} in Equation 7.53 for t into Equation 7.50 gives (after simplification),

$$D_{\min} = L - \tau \left[v_P(0) - v_T + v_T \ln \left(\frac{v_T}{v_P(0)} \right) \right] \quad (7.54)$$

$$= 5000 - \frac{2000/32.2}{0.15} \left[35 - 15 + 15 \ln \left(\frac{15}{35} \right) \right] \left(\frac{5280}{3600} \right) = 572.3 \text{ ft} \quad (7.55)$$

The numerical values obtained analytically in Equations 7.53 and 7.55 are in agreement with the values obtained from the iterative search for the minimum separation.

Analytical solutions for finding the optimum point are seldom possible. When the system dynamics are modeled by nonlinear equations, iterative searches using simulation to obtain the objective function and numerical derivative approximations are often the only recourse. For example, the existence of nonlinear damping functions in the target–projectile system would necessitate a simulation-based approach to finding the minimum separation.

The optimization toolbox employs a different search method when the objective function and its derivative (partial derivatives in the multivariable case) are expressible in analytic form. The M-file “*obj_fcnD_sep.m*” includes definitions of both the objective function and its first derivative. The essential statements are

```
function [f,g] = obj_fcn_D_sep(t, L, tau, vT, vP_initial)
f = L+vT*t-tau*vP_initial*(1-exp(-t/tau));
g = vT-vP_initial*exp(-t/tau);
```

and the calling program “*Ch7_Toolbox_opt_sep_search.m*” references the function file “*obj_fcn_D_sep.m*” using

```
options = optimset ('GradObj', 'on');
t_min = fminunc (@obj_fcn_D_sep, t_init, options, L, tau, vT, vP_initial)
```

The “options” declaration is required to enable the gradient search method, which uses the first derivative of the objective function given in “*obj_fcn_D_sep.m*,” when the call to “*fminunc*” is made. The results obtained from running the M-file “*Ch7_Toolbox_opt_sep_search.m*” are identical with the analytical values given in Equations 7.53 and 7.55.

7.3.1 GRADIENT VECTOR

Our experience in the previous example taught us that knowledge of the slope, that is, first derivative of the objective function, could be used to reduce the number of iterations required to locate a local optimum. The same holds for objective functions involving several parameters. Instead of a single derivative, a gradient vector with components equal to the partial derivatives of the objective function with respect to each parameter is computed. The gradient vector of a multivariable function at a point in parameter space points in the direction of maximum increase of the function. Furthermore, the magnitude of the gradient vector is a measure of the rate of increase in the objective function in the direction of the gradient.

Consider the function

$$f(x_1, x_2) = c_1(x_1 - h)^2 + c_2(x_2 - k)^2, -\infty < x_1 < \infty, -\infty < x_2 < \infty \quad (7.56)$$

The gradient vector at the point (x_1, x_2) is

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f(x_1, x_2)}{\partial x_1} \\ \frac{\partial f(x_1, x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2c_1(x_1 - h) \\ 2c_2(x_2 - k) \end{bmatrix} \quad (7.57)$$

[Figure 7.23](#) portrays the objective function as a surface $z = f(x_1, x_2)$ for the case where $h = 5$, $k = 10$, $c_1 = 1$, and $c_2 = 4$. Several contours that are projections of constant z in the $x_1 - x_2$ plane are also shown. The global minimum occurs at $x_1 = h = 5$ and $x_2 = k = 10$ and the minimum value is $f(h, k) = f(5, 10) = 0$.

Since the gradient vector at (x_1, x_2) points in the direction of maximum increase of $f(x_1, x_2)$, the orthogonal direction that coincides with the tangent to the contour at (x_1, x_2) represents the direction of zero change in $f(x_1, x_2)$. The negative of the gradient vector is drawn at several points in the $x_1 - x_2$ parameter space in [Figure 7.24](#) because $-\nabla f(x_1, x_2)$ points in the direction of maximum decrease of $f(x_1, x_2)$, and we are looking at minimizing the objective function.

[Table 7.4](#) includes the points shown in [Figure 7.24](#), the value of z for the contour, which the points lie on, the negative gradient vector, and its magnitude. The lengths of the negative gradient vectors are drawn proportional to their magnitudes given in the table.

A multivariable function like $f(x_1, x_2)$ is expandable about a point (\bar{x}_1, \bar{x}_2) using a two-dimensional Taylor Series, that is,

$$f(x_1, x_2) = f(\bar{x}_1, \bar{x}_2) + \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_1}(x_1 - \bar{x}_1) + \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_2}(x_2 - \bar{x}_2) + h.o.t. \quad (7.58)$$

where *h.o.t.* represents higher order terms involving powers and products of $(x_1 - \bar{x}_1)$ and $(x_2 - \bar{x}_2)$. To a first-order approximation, the change in $f(x_1, x_2)$ about the point (\bar{x}_1, \bar{x}_2) is

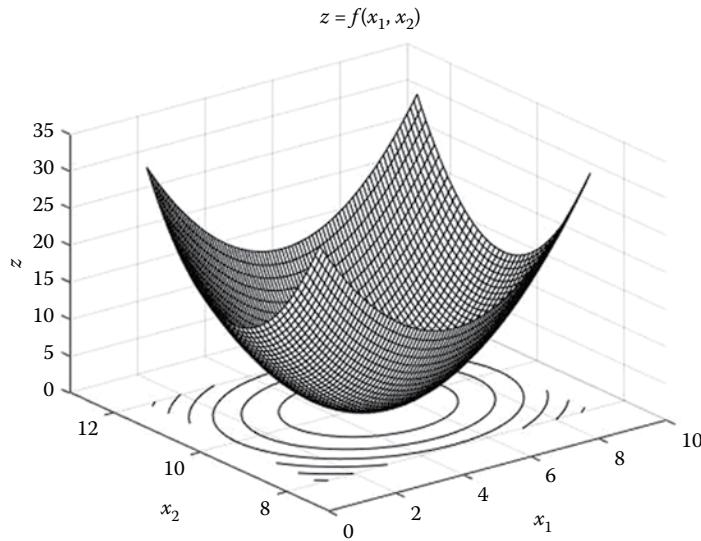


FIGURE 7.23 Graph of surface $z = f(x_1, x_2)$ and several contours $z = \text{const.}$

$$f(x_1, x_2) - f(\bar{x}_1, \bar{x}_2) \approx \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_1} \Delta x_1 + \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_2} \Delta x_2 \quad (7.59)$$

$$\Rightarrow \Delta f(\bar{x}_1, \bar{x}_2) \approx \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_1} \Delta x_1 + \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_2} \Delta x_2 \quad (7.60)$$

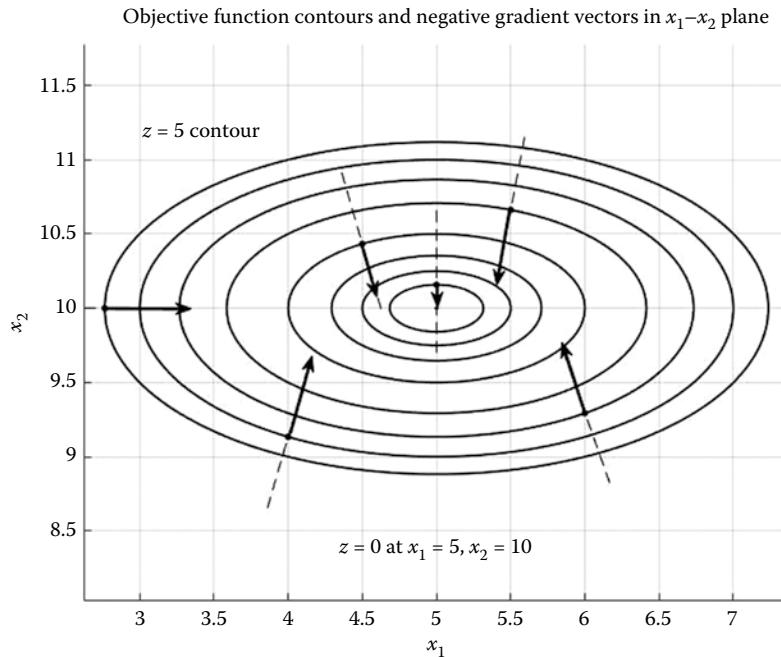


FIGURE 7.24 Contours of the objective function $f(x_1, x_2) = (x_1 - 5)^2 + 4(x_2 - 10)^2$.

TABLE 7.4
Contour and Gradient Data for Points Shown in Figure 7.24

(x_1, x_2)	Contour	$-\nabla f(x_1, x_2)$	$\ \nabla f(x_1, x_2)\ $
(2.7639, 10)	5	$\begin{bmatrix} 4.4721 \\ 0 \\ 2 \\ 6.9282 \\ -2 \\ 5.6569 \end{bmatrix}$	4.47211
(4, 9.1340)	4	$\begin{bmatrix} 4.4721 \\ 0 \\ 2 \\ 6.9282 \\ -2 \\ 5.6569 \end{bmatrix}$	7.2111
(6, 9.2929)	3	$\begin{bmatrix} 4.4721 \\ 0 \\ 2 \\ 6.9282 \\ -2 \\ 5.6569 \end{bmatrix}$	6.0000
(5.5, 10.6614)	2	$\begin{bmatrix} 4.4721 \\ 0 \\ 2 \\ 6.9282 \\ -2 \\ 5.6569 \end{bmatrix}$	5.3852
(4.5, 10.4330)	1	$\begin{bmatrix} 4.4721 \\ 0 \\ 2 \\ 6.9282 \\ -2 \\ 5.6569 \end{bmatrix}$	3.6056
(5, 10.15, 8.1)	0.1	$\begin{bmatrix} 4.4721 \\ 0 \\ 2 \\ 6.9282 \\ -2 \\ 5.6569 \end{bmatrix}$	1.2649

$$\Rightarrow \Delta f(\bar{x}_1, \bar{x}_2) \approx \begin{bmatrix} \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_1} \\ \frac{\partial f(\bar{x}_1, \bar{x}_2)}{\partial x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} = \nabla f(\bar{x}_1, \bar{x}_2)^T \Delta \underline{x} \quad (7.61)$$

and therefore, $\Delta f(\bar{x}_1, \bar{x}_2) \approx 0$ provided the gradient vector is identically zero at (\bar{x}_1, \bar{x}_2) . Quite understandably, the search for local extremes (minima and maxima) of the objective function $f(x_1, x_2)$ is based on finding points where the gradient vector $\nabla f(x_1, x_2) = [0 \ 0]^T$. The gradient vector also vanishes at a saddle point, which is neither a local minimum nor maximum. A test involving the matrix of second partials at points where the gradient is zero can distinguish between local extrema and saddle points.

Optimum seeking methods search for extreme values (minima and maxima) of an objective function using the gradient vector in some way (Converse 1970; Miller 1975, 2000; Hasdorff 1976; Daniels 1978; Bryson 1999). The references include both constrained and unconstrained optimization problems. In constrained optimization, a subset of the parameters are constrained in some fashion limiting the region of feasible solutions for finding the optimum. Typically, the constraints are inequalities reflecting limitation of system resources or existence of physical boundaries for safe operation.

7.3.2 OPTIMIZING MULTIPARAMETER OBJECTIVE FUNCTIONS REQUIRING SIMULINK MODELS

We now focus on multiparameter objective functions, which require execution of a Simulink model to evaluate. The following example is one of a control system where the objective is to minimize a performance measure by choosing two parameters associated with the controller. The performance measure is obtained from simulation and the optimization toolbox is used to find the optimum control settings.

A block diagram of a heading control system for a ship is shown in Figure 7.25. The ship's autopilot and power amplifier are an ideal proportional-derivative (PD) controller that converts an error signal to an amplified voltage for driving the steering gear connected to the ship's rudder. The steering gear, rudder, and hull dynamics are combined into a single ship dynamics transfer function. A gyro compass in the feedback loops senses the ship's heading and sends a voltage to the autopilot.

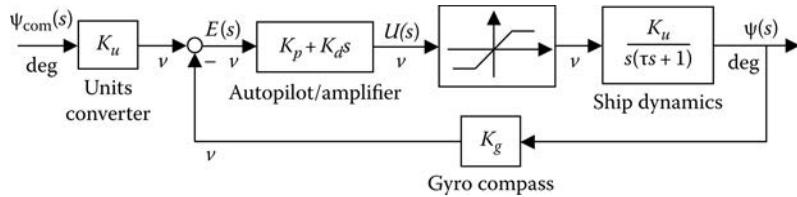


FIGURE 7.25 Block diagram of ship heading control system.

A saturation block is inserted between the controller and ship transfer function to account for the limited power available to the steering system. The units converter transforms the commanded heading from degree to volts for compatibility with the autopilot's electronics.

The control parameters K_p and K_d are to be selected to optimize the system response to a step input in command heading. There are numerous measures that can be used to characterize the step response. Five specific measures are enumerated as follows:

1. Rise time t_r : Time required for response to go from 10 to 90% of its final heading
2. Maximum overshoot, OS_{max} : Difference between maximum heading and final heading in underdamped systems
3. Maximum heading rate, $|\dot{\psi}_{max}|$: Maximum rate of change in ship's heading
4. Integral squared error, ISE: Integral of squared error from time zero to infinity
5. Integral absolute error, IAE: Integral of absolute value of error from zero to infinity

The objective function f is assumed to be a function of these measures, that is,

$$f = f(t_r, OS_{max}, |\dot{\psi}_{max}|, ISE, IAE) = F(K_p, K_d) \quad (7.62)$$

Note that the objective function is implicitly dependent on K_p and K_d because each of the measures t_r , OS_{max} , $|\dot{\psi}_{max}|$, ISE, and IAE depends on these parameters. The goal is to find the optimum value f_{opt} where

$$f_{opt} = \underset{K_p > 0, K_d > 0}{\text{Min}} F(K_p, K_d) \quad (7.63)$$

In this example, f is set to a linear combination of the five measures. Hence,

$$f(t_r, OS_{max}, |\dot{\psi}_{max}|, ISE, IAE) = c_1 t_r + c_2 OS_{max} + c_3 |\dot{\psi}_{max}| + c_4 ISE + c_5 IAE \quad (7.64)$$

The constants c_1 , c_2 , c_3 , c_4 , and c_5 determine the weights of each measure. For example, if the goal is to minimize the integral squared error (ISE),

$$\text{ISE} = \int_0^{\infty} e^2(t) dt = \int_0^{\infty} [\psi_{com} - \psi(t)]^2 dt \quad (7.65)$$

the weights are set to $c_1 = c_2 = c_3 = c_5 = 0$ and $c_4 = 1$.

The constrained optimization routine “fmincon” in the optimization toolbox implements a search for f_{opt} subject to parameter constraints. The statement

```
[opt_Kp_Kd, FVAL, EXITFLAG, OUTPUT] = fmincon (@obj_fcn_ship, Kp_Kd_
init, A, B, Aeq, Beq, LB, UB, NONLCON, OPTIONS, Kg, L, Ks, tau, t1,
theta_com, c)
```

in “*Ch7_Toolbox_opt_ship.m*” invokes a constrained search for the optimum values of parameters K_p and K_d . The arguments “ A , B , A_{eq} , B_{eq} , LB , UB , $NONLCON$ ” define the constraints. “ LB ” and “ UB ” are used to set lower and upper bounds on the parameters, and the remaining arguments are empty arrays not applicable in this example.

Before we look at the results, it is instructive to visualize the objective function surface with respect to the K_p - K_d plane. The objective function in this example is

$$f = t_r + OS_{max} + |\dot{\psi}_{max}| = F(K_p, K_d) \quad (7.66)$$

It is shown in [Figure 7.26](#) for the region $0 \leq K_p \leq 25$, $0 \leq K_d \leq 25$. The data points for drawing the surface were obtained by repeated calls to the Simulink model “*ship.mdl*” from the M-file “*Ch7_ship_control.m*.“ The simulated step responses were executed for 100 s, a period of time sufficient to allow the transient response to vanish, except for heavily damped cases (low K_p , high K_d). Numerical values of the system parameters are $K_u = K_g = 10$ V/rad, $K_s = 0.04$ rad/s/V, and $\tau = 10$ s, and the autopilot/amplifier saturates at 25 V. The commanded heading ψ_{com} was set to 30°.

In runs where the ship’s heading had yet to reach 90% of the final heading (which did not occur for the points shown in [Figure 7.26](#)), the rise time was set to 100 s. When the ship’s heading failed to reach the final heading, the overshoot was set to zero. The final heading is the commanded heading for all combinations of K_p and K_d resulting in a stable response.

The Simulink block diagram for the model “*ship.mdl*” is shown in [Figure 7.27](#).

The “PID” block in [Figure 7.27](#) is present in the “Simulink Extras” library. It is an ideal PID controller with parameters P , I , and D in the transfer function

$$G(s) = P + \frac{I}{s} + D_s \quad (7.67)$$

For simulation runs, P assumed the value of K_p , I was zero, and D assumed the value of K_d . The optimization toolbox search algorithm started from the point (1, 1) in the K_p - K_d plane. A gradient search is not used since the gradient of the objective function is not available in analytic form.

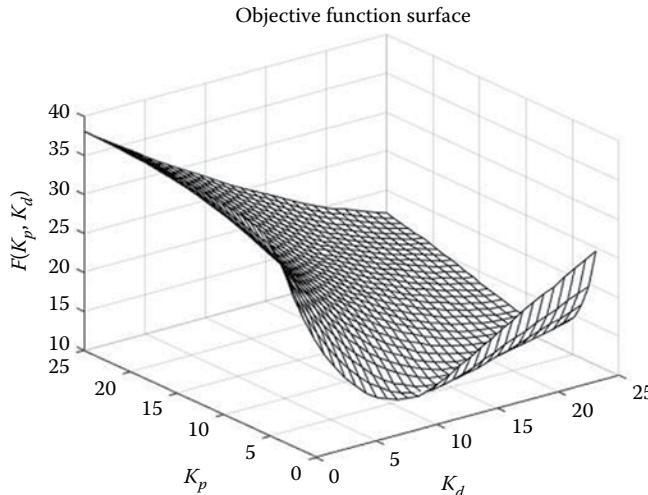


FIGURE 7.26 Objective function surface $f = t_r + OS_{max} + |\dot{\psi}_{max}| = F(K_p, K_d)$.

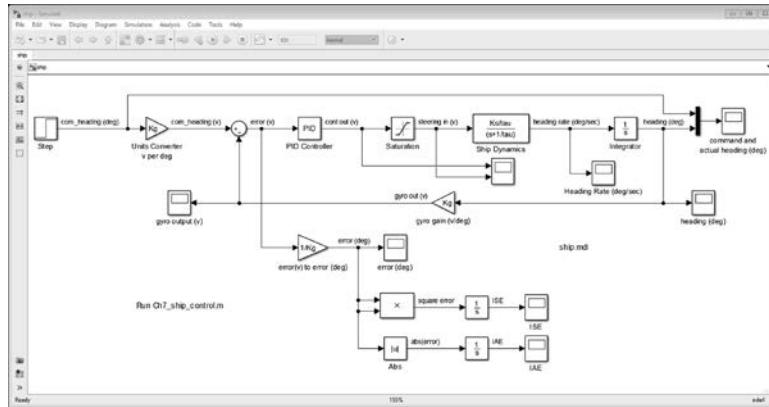


FIGURE 7.27 Simulink block diagram for ship heading step response.

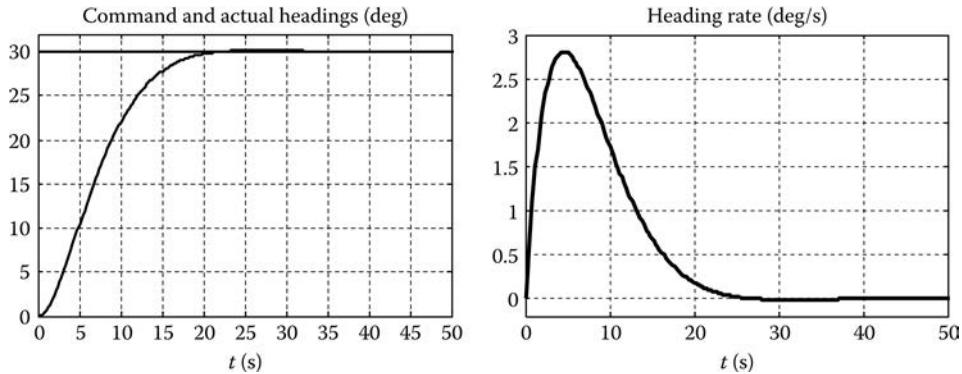


FIGURE 7.28 Results with optimal parameter settings: $(K_p)_{\text{opt}} = 1.3011$, $(K_d)_{\text{opt}} = 7.1913$.

A gradient-based search would require numerical approximations to the gradient at a number of points along the way to finding the optimum. The number of objective function evaluations would likely increase significantly depending upon the algorithm's efficiency in locating the optimum. In this example, a “medium-scale, SQP, Quasi-Newton, and line-search” algorithm (see optimization toolbox reference manual), was used successfully to find the optimum solution, namely, $(K_p)_{\text{opt}} = 1.3011$, $(K_d)_{\text{opt}} = 7.1913$, and $f_{\text{opt}} = 14.4395$. A total of 344 function evaluations and, hence, the same number of simulation runs were required.

The simulation was run with the optimal parameter settings to verify the objective function value. The ship's heading and heading rate are shown in Figure 7.28. The rise time, maximum overshoot, and maximum heading rate are 11.5000 s, 0.1154 deg, and 2.8241 deg/s, respectively. From Equation 7.66, $f = 11.5000 + 0.1154 + 2.8241 = 14.4395$.

7.3.3 PARAMETER IDENTIFICATION

Knowledge of the system parameters appearing in the differential and algebraic equations used to model continuous-time dynamic systems is often imperfect. When the simulationist is reasonably confident that the model's structure is suitable for its intended purpose, the subject of parameter identification arises (see Figure 7.1). A simple approach predicated on minimizing the differences between observed and simulated responses is presented in the following example.

The decreasing concentration of a chemical in solution follows a law from reaction kinetics that states

$$\frac{dx}{dt} = -kx^n \quad (k > 0, n > 0) \quad (7.68)$$

where

$x = x(t)$ is the concentration

k is a rate constant

n is the order of the reaction

Suppose the concentration of a chemical in solution was measured and recorded once a minute for 60 min. The values at 5 min intervals are shown in [Table 7.5](#).

The problem before us is to estimate the reaction constant k and reaction order n . We will do this by simulating the response for the chemical concentration starting with guessed values for k and n . The observed and simulated responses are used to compute the sum of squared errors, that is,

$$SSE = f(k, n) = \sum_{i=0}^{60} [\hat{x}_i - x_i]^2 \quad (7.69)$$

where

$x_i = x(t_i)$, $i = 0, 1, 2, \dots, 60$ are simulated concentrations a minute apart

$\hat{x}_i = \hat{x}(t_i)$, $i = 0, 1, 2, \dots, 60$ are values of concentration measured at one-minute intervals, some of which are shown in [Table 7.5](#)

Minimizing the objective function $f(k, n)$ yields the optimal estimates of the reaction parameters.

Observed concentrations \hat{x}_i , $i = 0, 1, 2, \dots, 60$ are obtained by running the M-file “*Ch7_reaction_kinetics.m*,” which calls the Simulink model “*chemical.mdl*” with $k = 0.125$ and $n = 2.3$, representative of the true reaction. The Simulink block diagram is shown in [Figure 7.29](#). A search constrained to the first quadrant of the k - n plane is performed using one of the routines from the optimization toolbox. The search concludes with $k_{\text{opt}} = 0.1256$, $n_{\text{opt}} = 2.3037$ and $SSE = f(k_{\text{opt}}, n_{\text{opt}}) = 3.2303 \times 10^{-7}$.

A graph of the simulated concentration response with the optimal parameter values is shown in [Figure 7.30](#). As expected, the observed concentrations fall on the simulated concentration response curve.

7.3.4 EXAMPLE OF A SIMPLE GRADIENT SEARCH

The common feature of all gradient search algorithms is their reliance on calculation of the gradient vector at a point in the parameter space. The logic for choosing a direction and step size leading to the next point along with the frequency of gradient calculations is what distinguishes one gradient

TABLE 7.5
Measured Concentration of Chemical in Solution at the End of 5 min Intervals

t (min)	0	5	10	15	20	25	30
\hat{x} (mol/L)	0.5000	0.4015	0.3386	0.2945	0.2617	0.2363	0.2159
t (min)	35	40	45	50	55	60	
\hat{x} (mol/L)	0.1991	0.1851	0.1731	0.1628	0.1538	0.1459	

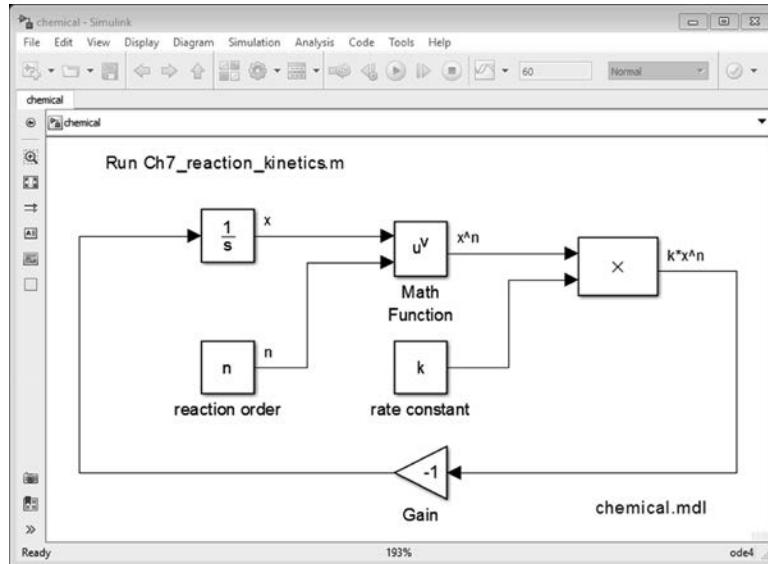


FIGURE 7.29 Simulink diagram for chemical reaction.

search algorithm from another. The gradient search presented in this section is intended to demonstrate how to exploit the property of the gradient vector to find a local minimum of an objective function. It is less efficient in comparison with established gradient search algorithms reported in the literature.

The focus of our attention is a bowl-shaped tank shown in Figure 7.31. The bowl is the lower half of a sphere of radius R . Water flow into and out of the tank is controlled by the valves located in the inflow and exiting pipes. The inflow $F_1(t)$ is maintained at a constant value F_1 . The outflow $F_2(t)$ is

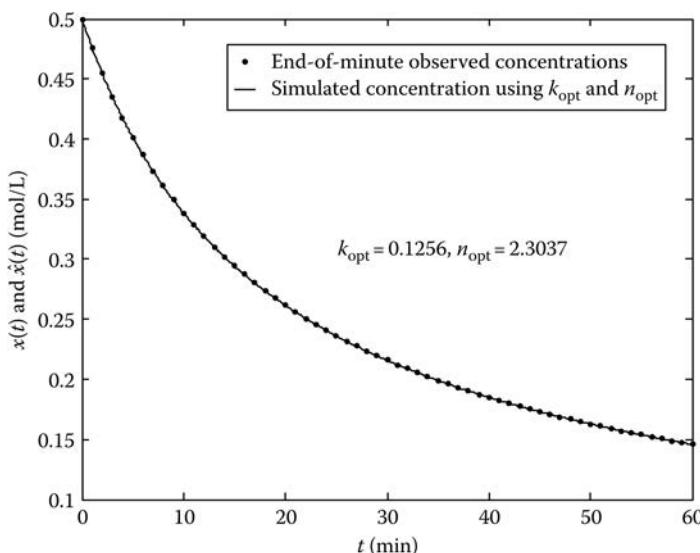


FIGURE 7.30 Graph of observed and simulated ($k = k_{opt}$, $n = n_{opt}$) concentration with optimal parameters.

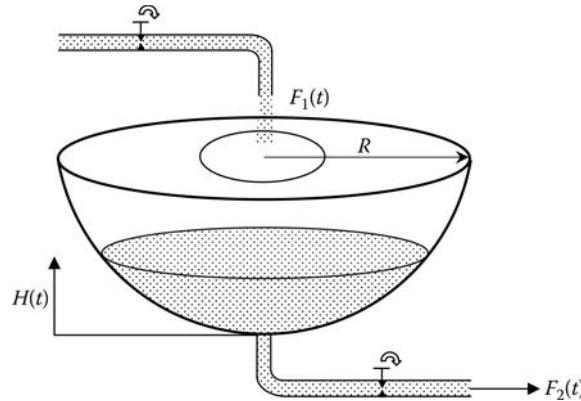


FIGURE 7.31 Hemispherical bowl with flows in and out.

a function of water level $H(t)$ and the opening of the valve in the discharge line, which effectively determines the constant c in the equation

$$F_2(t) = c[H(t)]^{1/2} \quad (7.70)$$

Conservation of mass requires

$$\frac{d}{dt}V(t) = F_1(t) - F_2(t) \quad (7.71)$$

where the volume $V(t)$ is related to the water level by

$$V(t) = \frac{1}{3}\pi H^2(t)[3R - H(t)] \quad (7.72)$$

Differentiating Equation 7.72 with respect to t and simplifying yield

$$\frac{d}{dt}V(t) = \pi[2R - H(t)]H(t)\frac{d}{dt}H(t) \quad (7.73)$$

Combining Equations 7.70, 7.71, and 7.73 results in the differential equation model

$$\pi[2R - H(t)]H(t)\frac{d}{dt}H(t) = F_1(t) - c[H(t)]^{1/2} \quad (7.74)$$

The term $\pi[2R - H(t)]H(t)$ is equal to the cross-sectional area of the bowl at the water level $H(t)$, that is,

$$A(H) = \pi(2R - H)H \quad (7.75)$$

And, therefore, Equation 7.74 is expressible as

$$A(H)\frac{dH}{dt} = F_1 - cH^{1/2} \quad (7.76)$$

The objective is to fill the tank in a specified period of time. The inflow F_1 and discharge constant c are the controllable parameters at our disposal. Before we discuss the gradient search, the objective function must be defined. Since the goal is to fill the tank in a given period of time, say T_{des} , the objective function is defined as

$$F(t_{\text{fill}}) = \begin{cases} A \left(\frac{t_{\text{fill}}}{T_L} - 1 \right)^2, & 0 \leq t_{\text{fill}} < T_L \\ 0, & T_L \leq t_{\text{fill}} \leq T_H \\ B \left(\frac{t_{\text{fill}} - T_H}{T_{\text{max}} - T_H} \right)^2, & T_H < t_{\text{fill}} \leq T_{\text{max}} \\ B, & T_{\text{max}} < t_{\text{fill}} \end{cases} \quad (7.77)$$

$F(t_{\text{fill}})$ is zero whenever the time to fill the tank t_{fill} falls between $T_L = T_{\text{des}} - \Delta/2$ and $T_H = T_{\text{des}} + \Delta/2$ where Δ is the width of the interval centered at T_{des} . The constant T_{max} is an arbitrarily chosen upper limit. A and B determine the objective function at the points $t_{\text{fill}} = 0$ and $t_{\text{fill}} = T_{\text{max}}$. A graph of $F(t_{\text{fill}})$ is shown in [Figure 7.32](#).

It is helpful to visualize the objective function surface over the F_1-c plane. For convenience, let the maximum inflow be $(F_1)_{\text{max}} = 10 \text{ ft}^3/\text{min}$ when the inlet valve is wide open. Furthermore, a maximum value of $c_{\text{max}} = 2 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$ is assumed, corresponding to a wide-open valve in the discharge line.

The objective function surface is shown in [Figure 7.33](#). It is plotted in the M-file “Ch7_globe_fill_surface.m,” which loops through a range of F_1 and c values, calling the Simulink model file “globe.mdl” to determine the fill time t_{fill} . The simulation terminates when the tank is full, that is, $H(t) = R = 5 \text{ ft}$, or failing that when the simulated time reaches $T_{\text{max}} = 300 \text{ min}$.

It appears from looking at [Figure 7.33](#) that the surface contains a ridge extending from $c = 0$ to $c = c_{\text{max}} = 2$ (with corresponding F_1 values) for which the objective function is zero. Indeed, this is consistent with our intuition, which suggests the likelihood of numerous combinations of F_1 and c yielding a tank fill time between $T_L = 145 \text{ min}$ and $T_H = 155 \text{ min}$ and, thus, $F(F_1, c) = 0$.

Another distinguishing characteristic in the surface’s topology is the plateau at an elevation of 50 (the value of B) corresponding to points (F_1, c) for which the tank fill time is greater than or equal

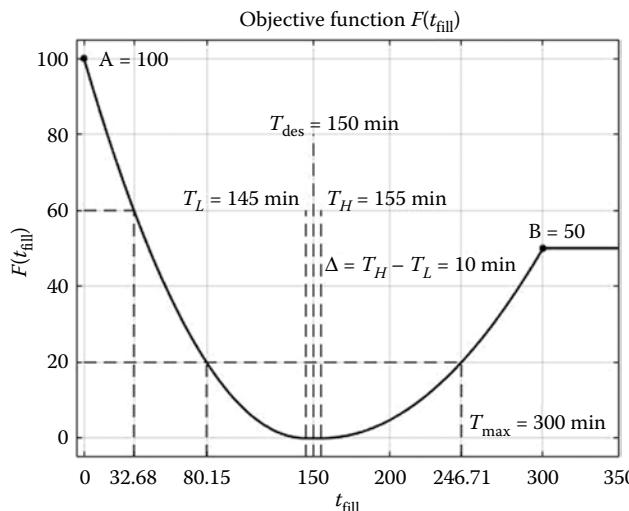


FIGURE 7.32 Graph of objective function $F(t_{\text{fill}})$.

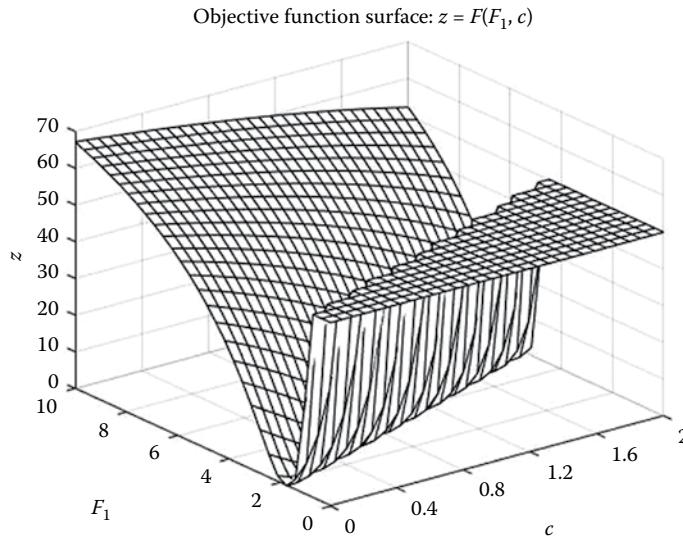


FIGURE 7.33 Objective function surface $z = F(F_1, c)$ for tank-filling problem.

to 300 min or else the tank never fills. The challenge will be for the gradient search algorithm to find points in parameter space along the aforementioned ridge where the objective function is a minimum, that is, zero.

The Simulink block diagram is shown in Figure 7.34.

The parameters F_1 and c and the tank fill time t_{fill} are visible in Simulink “display” blocks. Note the limited integrator with upper limit set to $R = 5$ ft, which also happens to be identical to the threshold parameter of the “Switch” block. Consequently, the simulation is halted when the tank is full, that is, the level $H(t) \geq R$.

A variable-step “ode45” Dormand Prince numerical integrator with default tolerance settings is used to control the truncation error. Execution times are reduced by a significant amount compared to one of the RK fixed-step integrators with suitably chosen integration step (see Exercise 7.10).

The gradient search implemented in “Ch7_grad_search_globe_fill.m” is outlined in flow chart form in Figure 7.35. It begins with a user-selected starting point (F_1, c) in F_1-c parameter space.

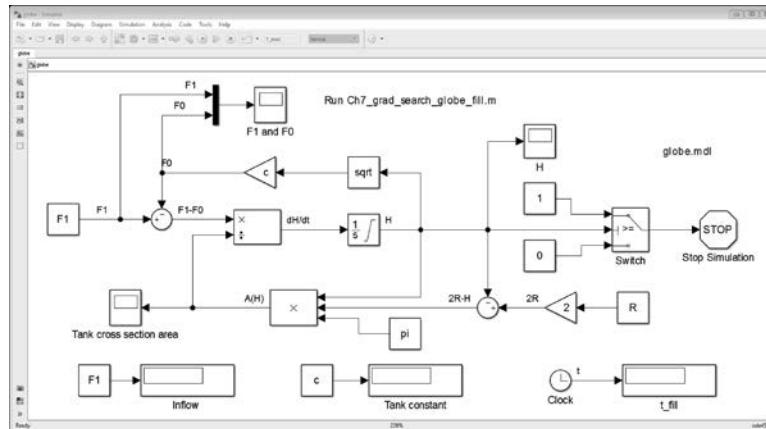


FIGURE 7.34 Simulink diagram for hemispherical tank-filling simulation.

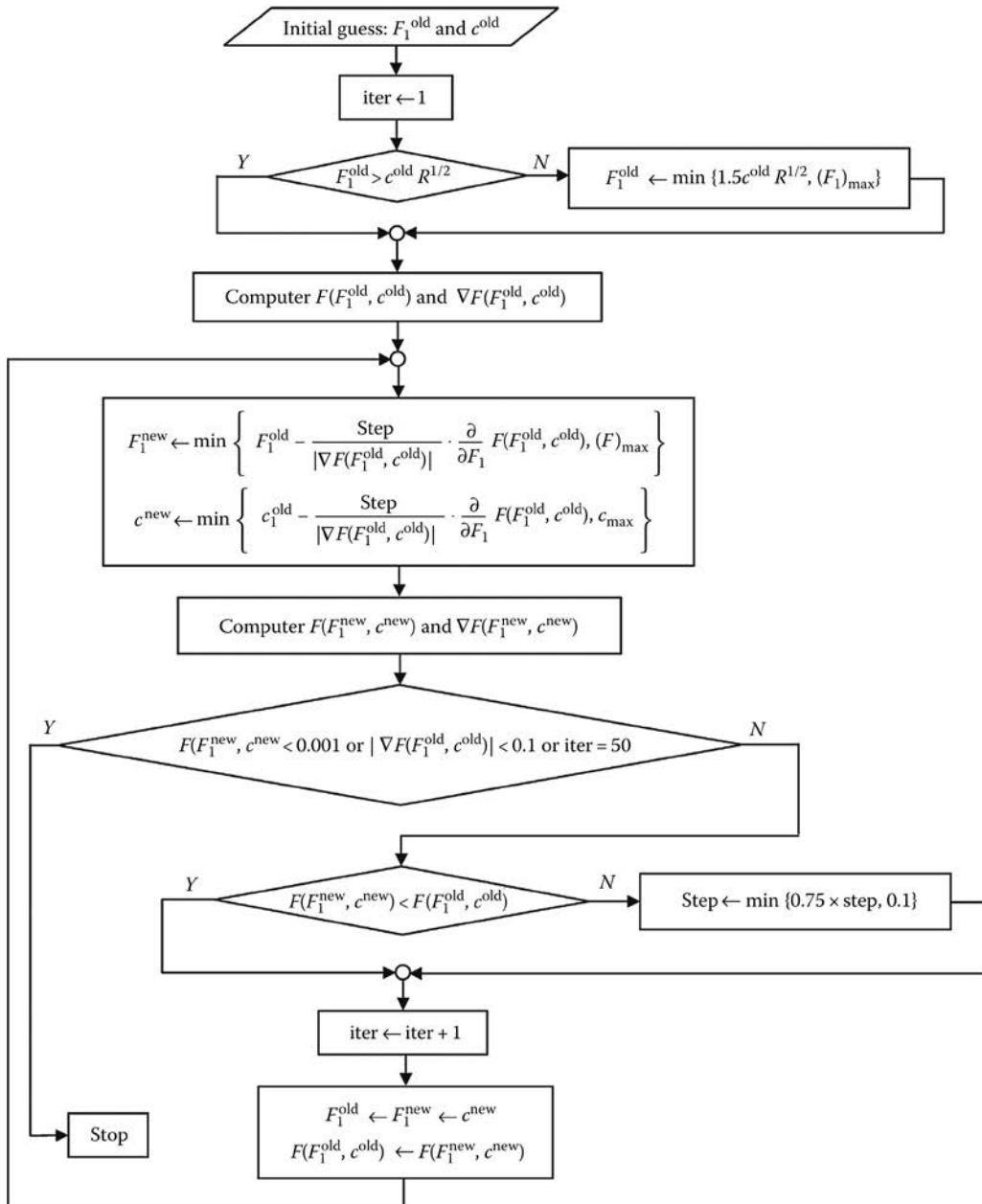


FIGURE 7.35 Flow chart for gradient search algorithm.

Prior to the calculation of the gradient, the point is checked to verify the possibility of the tank filling up. At steady state, we know from Equation 7.76 that

$$(F_1)_{ss} - c(H_{ss})^{1/2} = 0 \quad (7.78)$$

and, therefore, imposing the constraint

$$F_1 > cR^{1/2} \quad (7.79)$$

guarantees that the water will eventually attain a level of $R = 5$ ft, although not necessarily in less than $T_{\max} = 300$ min. If the initial point fails to satisfy the inequality in Equation 7.79, the initial inflow F_1 is adjusted according to

$$F_1 = \min \left\{ 1.5cR^{1/2}, (F_1)_{\max} \right\} \quad (7.80)$$

The remaining blocks in the flow chart are for computing the objective function, the gradient vector, and for determining how big a step to take in the negative gradient direction in searching for a minimum, that is, points where $F(F_1, c) = 0$.

The so-called steepest descent gradient searches (Wilde 1964) look for the optimum distance to travel in the negative gradient direction before changing directions. The optimum distance is determined by the local minimum of the objective function along the negative gradient direction. When the local minimum is reached, the gradient vector is recalculated, and the search proceeds in the new direction that happens to be orthogonal to the previous search direction. Hence, with steepest descent as described, the search consists of a sequence of orthogonal moves from point to point. The distance between consecutive points varies, generally decreasing as the optimum is approached.

The gradient search illustrated in Figure 7.35 is not of the steepest descent type; rather, it consistently takes a single step in the negative gradient direction from one point to the next and then recomputes the gradient vector.

The magnitude of the step is altered based on a comparison of the objective function at neighboring points, that is, after taking a full step, if the new objective function is greater than the previous value, the step size is reduced by 25% next time around. A lower threshold on step size is imposed to prevent the search from “slowing down to a crawl.” Compared to steepest descent, the steps are either too small or too large, and the search will require more gradient calculations. Even worse, the new gradient direction may steer the search away from the minimum altogether, and the method fails to converge.

The search is terminated using a stop condition based on the magnitude of the gradient vector, the value of the objective function, and the number of steps taken. After considerable experimentation, the tolerances were chosen to stop the search if

$$|\nabla F(F_1, c)| = \left\| \begin{bmatrix} \frac{\partial}{\partial F_1} F(F_1, c) \\ \frac{\partial}{\partial c_1} F(F_1, c) \end{bmatrix} \right\| \leq 0.1 \quad \text{or} \quad F(F_1, c) \leq 0.001 \quad \text{or} \quad \# \text{ steps} = 50 \quad (7.81)$$

The gradient vector $\nabla F(F_1, c)$ is calculated numerically using a central difference approximation formula, namely,

$$\frac{\partial}{\partial F_1} F(\bar{F}_1, \bar{c}) = \frac{F(\bar{F}_1 + \Delta F_1, \bar{c}) - F(\bar{F}_1 - \Delta F_1, \bar{c})}{2\Delta F_1} \quad (7.82)$$

$$\frac{\partial}{\partial c} F(\bar{F}_1, \bar{c}) = \frac{F(\bar{F}_1, \bar{c} + \Delta c) - F(\bar{F}_1, \bar{c} - \Delta c)}{2\Delta c} \quad (7.83)$$

where the deviations ΔF_1 and Δc are 0.005 and 0.01, respectively. The gradient vector is computed by calling the MATLAB function “*gradF_globe.m*” from the M-file “*Ch7_grad_search_globe_fill.m*” with arguments F_1 and c . The components in Equations 7.82 and 7.83 are returned as outputs.

Results of successful gradient searches starting from randomly chosen starting points in the region $0 \leq F_1 \leq (F_1)_{\max} = 10$, $0 \leq c \leq c_{\max} = 2$ of F_1 - c parameter space are shown in [Table 7.6](#). The search failed to locate the minimum on a few occasions.

A different approach to finding the optimum points located along the ridge in [Figure 7.33](#) is to plot the $F(F_1, c) = 0$ contour. Other contours $F(F_1, c) = F_0$, (F_0 constant) can be plotted as well by searching for points in the F_1 - c plane, which result in filling times corresponding to the required contour values. [Figure 7.32](#) shows the two filling times that result in $F(F_1, c) = 20$ and the single filling time that leads to $F(F_1, c) = 60$.

With $F_0 = 20$, the next step is fixing the parameter c and varying F_1 until the two values that lead to $t_{\min} = 80.15$ min and $t_{\text{fill}} = 246.71$ min are found. The search for F_1 is constrained to the interval $(F_1)_{\min} \leq F_1 \leq (F_1)_{\max}$, where $(F_1)_{\min}$ is the minimum flow needed to fill the hemispherical tank and is given by

$$(F_1)_{\min} = cR^{1/2} \quad (7.84)$$

where c is the current fixed value. In other words, points $\{(c, F_1) | F_1 < (F_1)_{\min}\}$ are infeasible and not searched. The process is repeated for c ranging from $c_{\min} = 0$ to $c_{\max} = 2 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$.

A similar process occurs when $B \leq F_0 \leq A$, except in this case, there is a single value of fill time corresponding to F_0 (see [Figure 7.32](#)). For $F(F_1, c) = F_0 = 60$, the fill time is 32.68 min. The $F_0 = 20$ and $F_0 = 60$ contours are shown in [Figure 7.36](#). The $F_0 = 0$ contour is also shown. The upper portion corresponds to fill times of $t_{\text{fill}} = T_L = 145$ min, and the lower segment is for $t_{\text{fill}} = T_H = 155$ min. (see M-file “*Ch7_Fig7_36.m*.”) Note that only three values of the parameter c were used, namely, c_{\min} , $(c_{\min} + c_{\max})/2$, and c_{\max} , when searching for the corresponding value of F_1 because the contours appear to be linear.

The M-file “*Ch7_globe_contours.m*” can be used to draw the contours ranging from $F_0 = 0$ up to a maximum value $(F_0)_{\max}$ corresponding to $c = c_{\min} = 0$ and $F_1 = (F_1)_{\max} = 10 \text{ ft}^3/\text{min}$. The contour for $F_0 = (F_0)_{\max}$ is a single point located at (0,10) in [Figure 7.36](#). “*Ch7_globe_contours.m*” reports the value of $(F_0)_{\max}$ along with the corresponding fill time, which happens to be the shortest time in which the tank can be filled. From [Figure 7.33](#), $(F_0)_{\max}$ appears to be approximately 68.

TABLE 7.6
Summary of Gradient Search Results for Five Starting Points in F_1 - c Plane

	#1	#2	#3	#4	#5
$(F_1)_{\text{start}}$	2.026	9.218	1.762	0.578	8.131
c_{start}	1.344	1.476	0.811	0.705	0.019
$F[(F_1)_{\text{start}}, c_{\text{start}}]$	1.756	52.922	6.776	17.013	60.377
$\nabla F[(F_1)_{\text{start}}, c_{\text{start}}]$	$\begin{bmatrix} 11.293 \\ -21.173 \end{bmatrix}$	$\begin{bmatrix} 6.038 \\ -10.780 \end{bmatrix}$	$\begin{bmatrix} -46.318 \\ 86.859 \end{bmatrix}$	$\begin{bmatrix} -96.765 \\ 81.472 \end{bmatrix}$	$\begin{bmatrix} 4.288 \\ -7.400 \end{bmatrix}$
$ \nabla F[(F_1)_{\text{start}}, c_{\text{start}}] $	23.996	12.356	98.437	205.659	8.553
$(F_1)_{\text{opt}}$	5.407	5.279	2.777	2.381	5.393
c_{opt}	1.995	1.898	0.558	0.345	1.995
t_{fill}	153.5	147.8	146.3	147.5	155.0
$F[(F_1)_{\text{opt}}, c_{\text{opt}}]$	0	0	0	0	0
$\nabla F[(F_1)_{\text{opt}}, c_{\text{opt}}]$	$\begin{bmatrix} -0.039 \\ 0.073 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.530 \\ 1.040 \end{bmatrix}$
$ \nabla F[(F_1)_{\text{opt}}, c_{\text{opt}}] $	0.083	0	0	0	1.167
Iterations	8	17	3	9	10

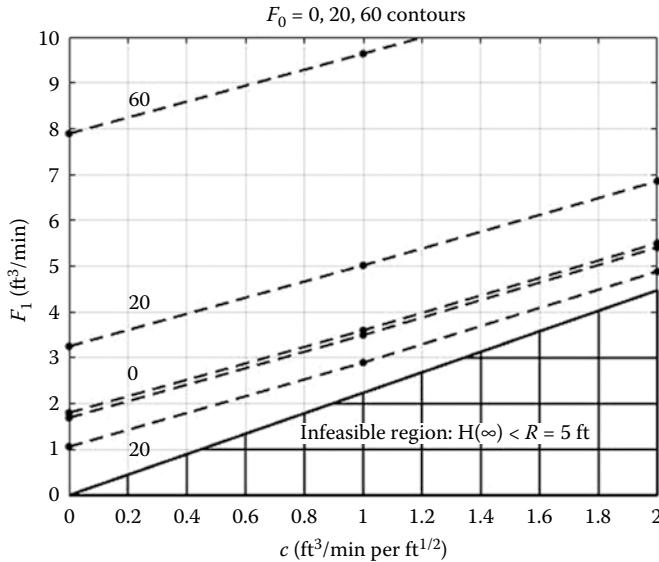


FIGURE 7.36 Objective function contours $F_0 = 0, 20, 60$.

MATLAB will also draw the objective function contours. The statements

```
v = [5 20 60];
contour (cc, F11, z, v)
```

in “*Ch7_Fig7_37.m*” produce the contours corresponding to objective function values of 5, 20, and 60 shown in [Figure 7.37](#). There is substantial agreement between the contours in [Figures 7.36](#) and [7.37](#).

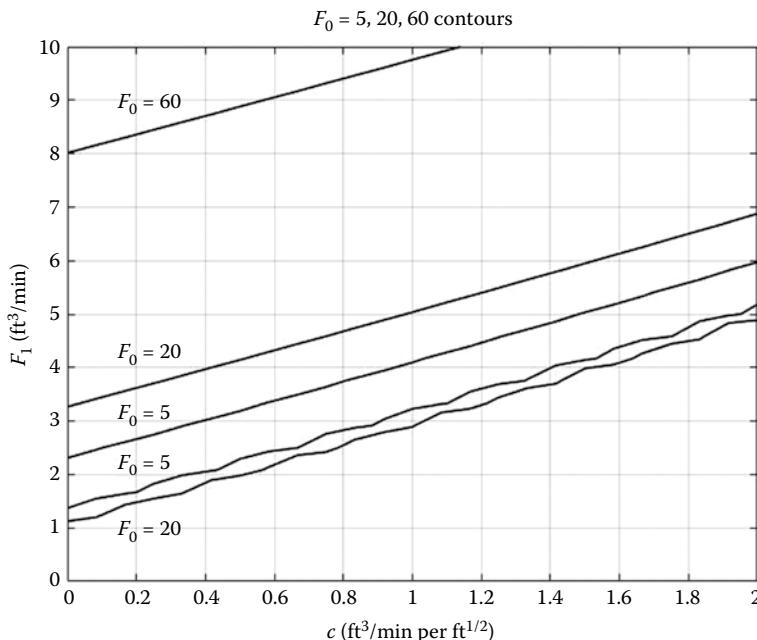


FIGURE 7.37 Graph of several contours of objective function $F(F_1, c)$.

7.3.5 OPTIMIZATION OF SIMULINK DISCRETE-TIME SYSTEM MODELS

We conclude this section with a simplified model of hospital–patient occupancy (McClamroch 1980) using Simulink to simulate the dynamics. The goal will be to investigate the relationship between the average number of scheduled patients per day on the hospital’s utilization of existing capacity. Stochastic systems of this nature, where entities arrive in nondeterministic fashion requiring services of random duration at different stages, are typically studied using discrete-event simulation (Banks 2005). Popular programs for simulating systems of this nature are Process Model (Evans) and ARENA (Kelton 1997).

While Simulink may not be the ideal program to simulate the dynamics of patients flowing through a hospital’s facilities, a macroscopic discrete-time system model that captures some of the important features is still possible. In the model to be formulated, the basic unit of discrete-time is a day.

The types of daily arrivals and departures from the hospital are accounted for by

$$e_i = \text{number of emergency arrivals on } (i+1)\text{st day}$$

$$s_i = \text{number of scheduled arrivals on } (i+1)\text{st day}$$

$$d_i = \text{number of departures on } (i+1)\text{st day}$$

$$m_i = \text{number of deaths on } (i+1)\text{st day}$$

Letting x_i denote the number of occupied beds at the end of the i th day and L the total number of beds, a simple model describing the hospital’s daily occupancy is

$$x_{i+1} = \text{Min}\{L, x_i + u_i\}, \quad i = 0, 1, 2, 3, \dots \quad (7.85)$$

where $u_i = s_i + e_i - d_i - m_i$. The components of u_i in Equation 7.85 are assumed to be normally distributed, that is,

$$s_i \sim N(\mu_S, \sigma_S^2), e_i \sim N(\mu_E, \sigma_E^2), d_i \sim N(\mu_D, \sigma_D^2), m_i \sim N(\mu_M, \sigma_M^2)$$

where $\mu_S, \mu_E, \mu_D, \mu_M$ and $\sigma_S^2, \sigma_E^2, \sigma_D^2, \sigma_M^2$ are the respective means and variances.

Typical sequences of u_i and x_i are shown in Figure 7.38. Note that u_i represents a summation of input components (arrivals and departures) during the $(i+1)$ st day.

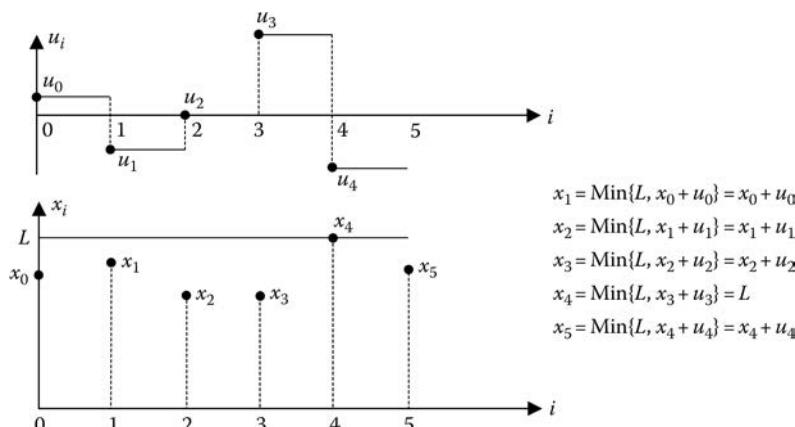


FIGURE 7.38 Illustration of discrete-time input and output relationship in Equation 7.85.

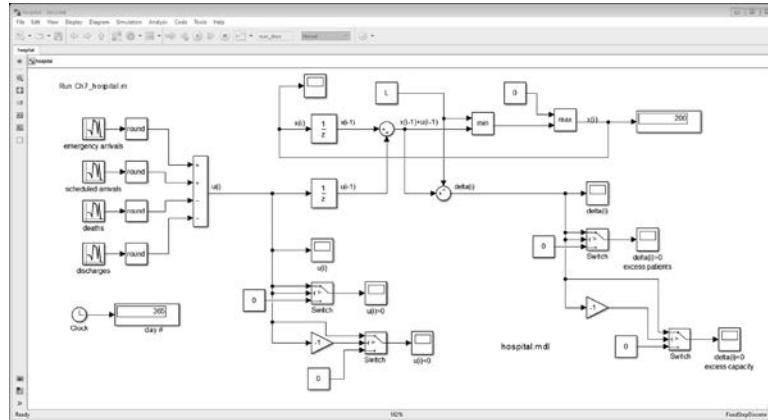


FIGURE 7.39 Simulink diagram of hospital occupancy.

A Simulink block diagram of the nonlinear, first-order, discrete-time system is shown in Figure 7.39.

In addition to generating the input components and implementation of the difference equation, additional blocks are used to decompose the input “ $u(i)$ ” into two series, called “ $u(i) > 0$ ” and “ $u(i) < 0$.” The first series “ $u(i) > 0$ ” is the subset of positive values in “ $u(i)$ ” corresponding to days where the number of new patients exceeds the number of patients discharged or who have died. At the end of those days, the hospital’s occupancy either increases (relative to the previous day) or else remains constant at its capacity.

The second series “ $u(i) < 0$ ” is the subset of negative values in “ $u(i)$ ” corresponding to days when the number of discharged and dying patients surpasses the number of arrivals and the hospital’s occupancy at the end of the day is diminished from the previous day.

Note also the presence of two Simulink “switch” blocks feeding “scopes” labeled “ $\text{delta}(i) > 0$ ” and “ $\text{delta}(i) < 0$.“ The former outputs a time series showing on which days and by how much the demand for beds exceeds the hospital’s capacity. The numerical values represent the overflow demand, that is, the amount of additional beds required to accommodate the influx of additional patients. The scope labeled “ $\text{delta}(i) < 0$ ” shows the days when the hospital is operating at less than capacity and by how much.

Typical profiles for “ $u(i) > 0$,” “ $u(i) < 0$,” “ $\text{delta}(i) > 0$,” and “ $\text{delta}(i) < 0$ ” are shown in Figures 7.40–7.43 for the case where the average number of admissions exceeds the average number of discharges plus deaths.

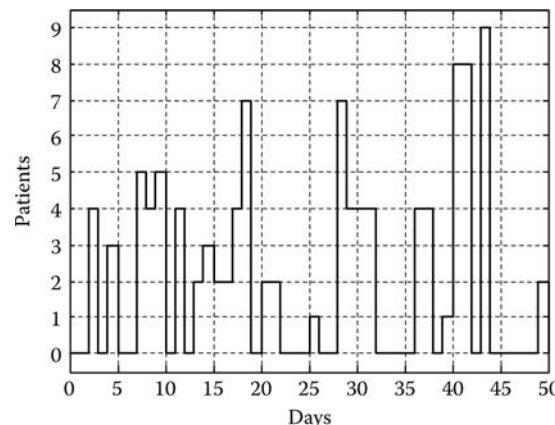


FIGURE 7.40 Typical “ $u(i) > 0$ ” profile—Days with excess of new patients.

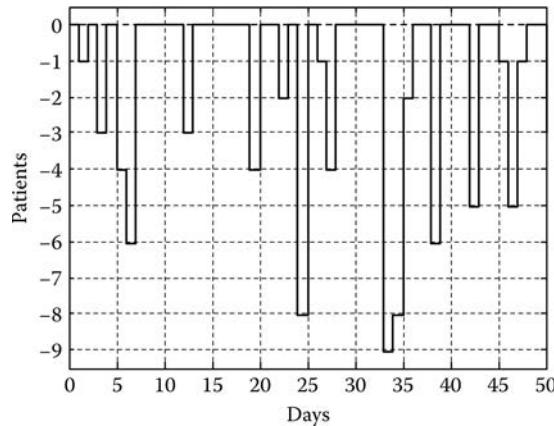


FIGURE 7.41 “ $u(i) < 0$ ” profile—Excess discharged and dying patients.

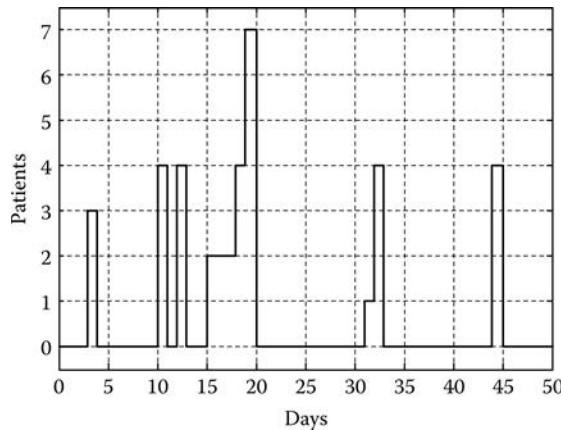


FIGURE 7.42 Typical “ $\delta(i) > 0$ ” profile—Days when capacity exceeded.

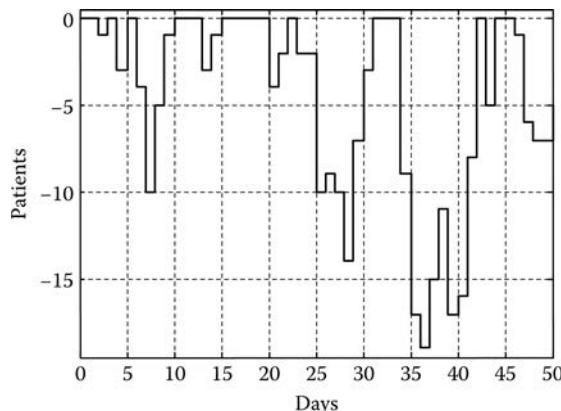


FIGURE 7.43 Typical “ $\delta(i) < 0$ ” profile—Days at less than capacity.

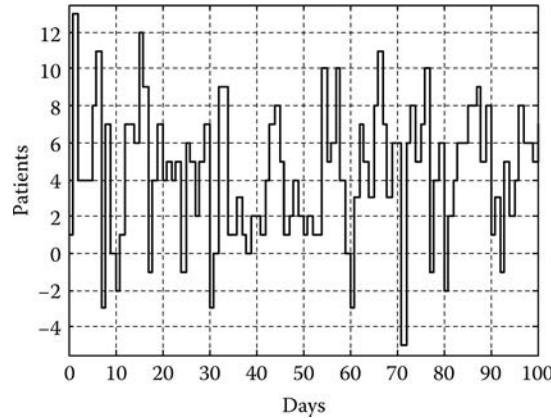


FIGURE 7.44 Daily net patient input.

Figures 7.44 and 7.45 show results of a single run for 100 days under the following conditions:

$$\mu_S = 21, \sigma_S^2 = 4, \mu_E = 5, \sigma_E^2 = 2$$

$$\mu_D = 23, \sigma_D^2 = 9, \mu_M = 2, \sigma_M^2 = 0.25$$

$$L = 200, x_0 = 200$$

Hospital occupancy fluctuates between 195 and 200 corresponding to occupancy rates ranging from 97.5 to 100%. The high occupancy rates are consistent with the condition $\mu_S + \mu_E > \mu_D + \mu_M$, that is, the average daily arrival of new patients is greater than the average number of patients leaving the hospital. From Figure 7.45, it is clear there are a number of days when patients may have been scheduled for admittance but were not admitted. (Keep in mind the simplistic nature of the model that does not account for the hospital's ability to accommodate excess patients.)

A Monte Carlo simulation can be performed to investigate the effect of scheduled arrivals on hospital utilization (occupancy rate) and the number of patients turned away due to lack of beds.

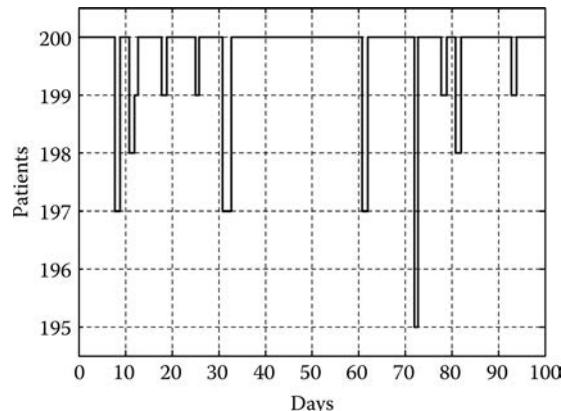


FIGURE 7.45 Hospital occupancy for 100 days.

“Ch7_hospital.m” is an M-file, which varies μ_s , the mean number of scheduled arrivals, and computes a number of performance measures based on 10 simulated records, each containing 1 year (365 days) of information. That is, for a given value of μ_s , 365 days of operation are simulated. The initial occupancy is reset to $x_0 = L$, and the process repeated nine more times. The remaining system parameters are fixed at the baseline values previously given.

A number of performance measures are computed for each value of μ_s :

- An objective function that accounts for the days when the hospital is unable to accept new patients due to excess demand, that is, “ $\Delta(i) > 0$,” and other days when the hospital operates at less than capacity, that is, “ $\Delta(i) < 0$.” It is a weighted average over all 10 records given by

$$F(\mu_s) = c_1 \left\{ \frac{1}{10} \sum_{j=1}^{10} \left[\frac{1}{365} \sum_{\substack{i=1 \\ \Delta(i)>0}}^{365} \Delta(i) \right]_j \right\} + c_2 \left\{ \frac{1}{10} \sum_{j=1}^{10} \left[\frac{1}{365} \sum_{\substack{i=1 \\ \Delta(i)<0}}^{365} |\Delta(i)| \right]_j \right\} \quad (7.86)$$

where c_1 and c_2 are the weights applied to the average number of excess patients per day and the average number of unused beds per day, respectively.

- The percent occupancy averaged over all 10 records (10×365 days).
- The average number of excess patients per day averaged over all 10×365 days.
- The average excess capacity (unused beds) per day averaged over all 10×365 days.

Results are shown in Figure 7.46 for $c_1 = c_2 = 1$.

Note the steep decline in objective function until $\mu_s = 20$, which represents an equilibrium condition in the sense that new arrivals and departures are balanced (on average), that is $\mu_s + \mu_E = \mu_D + \mu_M$. Choosing $c_1 = c_2 = 1$ implies that an unused bed and a nonadmitted patient have equal importance.

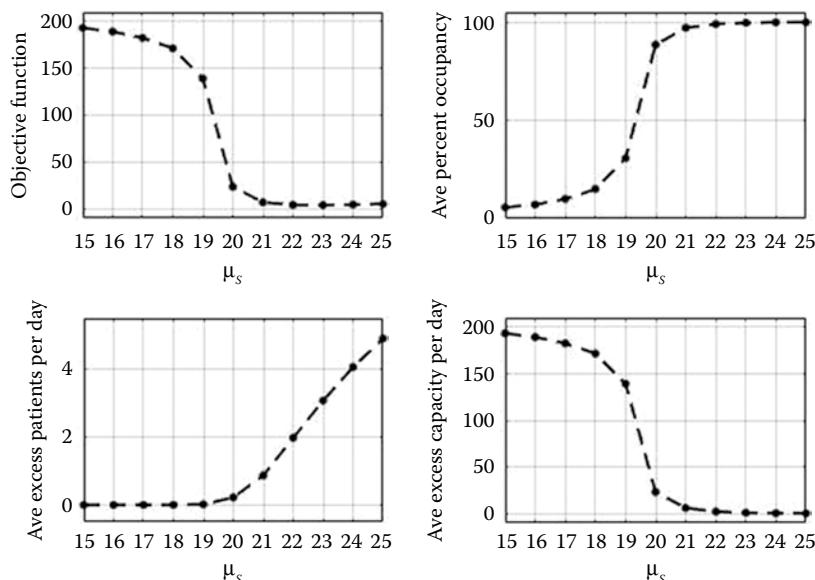


FIGURE 7.46 Objective function ($c_1 = c_2 = 1$) and other performance measures.

A few points to consider as we conclude this section are as follows:

1. Can you explain why the graphs of the objective function and the average excess capacity are nearly identical?
2. What should the hospital's admitting policy with respect to scheduled number of arrivals be to assure 100% occupancy rates?
3. What will happen to the objective function as the mean number of arrivals continues to increase beyond 25 as shown in [Figure 7.46](#).
4. What is the implication of changing the weight c_2 from 1 to 5 and what effect will it have on the objective function?
5. What is the significance of setting $\sigma_s^2 = 0$?
6. Is there a difference between simulating ten 1 year periods and one 10 year period as far as the Monte Carlo simulation is concerned?

EXERCISES

- 7.5 Suppose the movement of the target in [Figure 7.14](#) is along the circular path of radius $R = 2.5$ mi with speed given by

$$v(s) = V + v_0 \sin\left(\frac{2\pi s}{s_0}\right), \quad s \geq 0$$

where s is the distance traveled along the circular trajectory. The mean speed V and amplitude v_0 are uniformly distributed according to

$$V \sim U(20, 40 \text{ mph}), v_0 \sim U(0, 10 \text{ mph})$$

and the period $s_0 = 2000$ ft. The projectile's dynamics are defined by the parameters $m = 4000/32.2$ slugs, $\mu = 0.15$ lb s/ft, and $v_p(0) = 60$ mph.

Use the MATLAB random number generator to generate values for V and v_0 .

- a. Simulate single firings of the projectile corresponding to firing angles of $\theta = 0^\circ, 5^\circ, 10^\circ, \dots, 90^\circ$. Halt the simulation when the projectile has traveled a distance greater than R mi. Plot the miss distance (minimum separation between target and projectile) vs. the firing angle.
 - b. From the graph in part (a), estimate the optimum firing angle, that is, the one that results in the projectile striking the target.
 - c. Using the same values of V and v_0 , write an M-file to find the optimum firing angle. The use of MATLAB's optimization toolbox is optional.
 - d. Verify the result in part (c) by simulation.
- 7.6 A projectile of mass m is fired with initial velocity v_0 at an angle α_0 from the horizontal direction. Its position, while in flight, is given by coordinates (x, y) , and its velocity is represented by v as shown in [Figure E7.6a](#). The projectile is subject to a linear drag force f_D in the tangential direction and a constant gravitational force mg in the vertical direction.

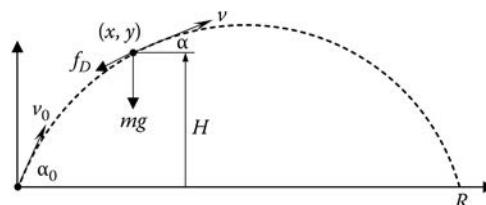


FIGURE E7.6A

The equations of motion are

$$\frac{d^2x}{dt^2} = -\frac{f_D}{m} \cos \alpha, \quad \frac{d^2y}{dt^2} = -\frac{f_D}{m} \sin \alpha - g$$

$$f_D = \mu |v| = \mu \left[\left(\frac{dx}{dt} \right)^2 + \left(\frac{dy}{dt} \right)^2 \right]^{1/2}$$

$$\tan \alpha = \frac{dy/dt}{dx/dt}$$

Baseline values of the system parameters are

$$m = 0.25 \text{ slugs}, v_0 = 500 \text{ ft/s}, c = 0.015 \text{ lbs/ft}, \alpha_0 = 45^\circ$$

Use the Simulink diagram as shown in [Figure E7.6b](#) or construct your own Simulink model to answer the following questions.

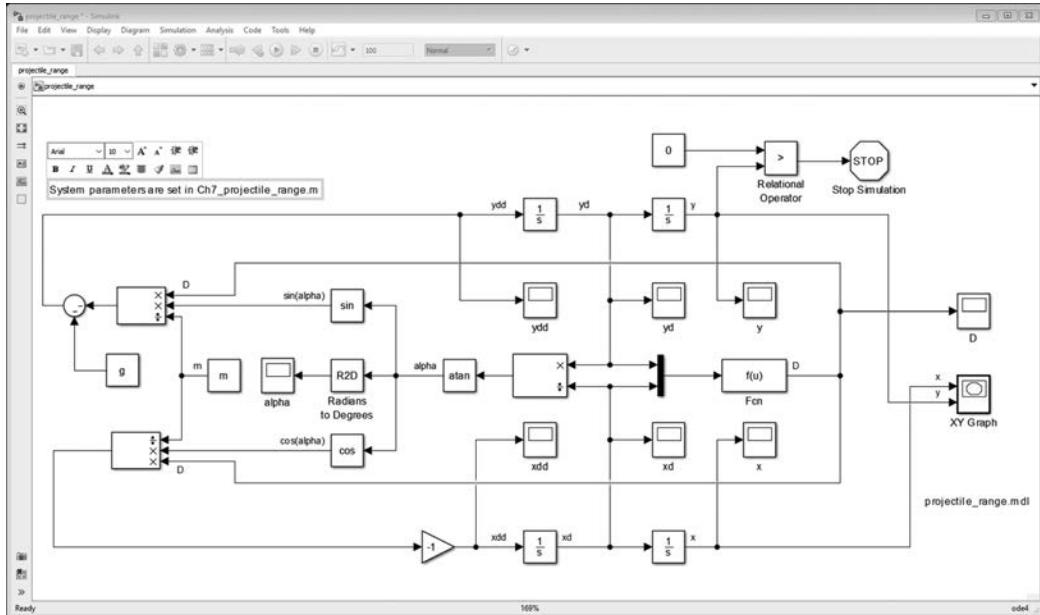


FIGURE E7.6B

- The projectile is fired in the vertical direction. Find
 - The analytical solution $y(t)$, $t \geq 0$ and plot it for the time period when $y \geq 0$
 - The peak altitude H attained by the projectile
 - The time t_p , when the peak altitude is reached
- Find the maximum altitude H by optimization, that is, search for the time when $-y(t)$ is a minimum. Compare the result with your answer in part (a).
- Find the horizontal distance R corresponding to firing angles $\alpha_0 = 0^\circ, 5^\circ, 10^\circ, \dots, 90^\circ$ and plot the results. Estimate the firing angle α_0 that maximizes R .

- d. Find the value $(\alpha_0)^{\text{opt}}$ that maximizes $R(\alpha_0)$ by the minimization of the objective function $F(\alpha_0) = -R(\alpha_0)$ subject to $0^\circ \leq \alpha_0 \leq 90^\circ$.
- e. Find the initial velocity v_0 that results in a peak altitude of $H = 1200$ ft when the projectile is fired at an angle of 45° . Formulate this as an optimization problem and then find the optimum solution.
- f. Let $\bar{H} = 1500$ ft and $\bar{R} = 2000$ ft be the design values of peak altitude and down range distance. The objective is to find combinations of initial firing angles and initial velocities resulting in $H = \bar{H}$ and $R = \bar{R}$. Choose the objective function to be minimized as

$$F(\alpha_0, v_0) = (e_H^2 + e_R^2)^{1/2} = [(H - \bar{H})^2 + (R - \bar{R})^2]^{1/2}$$

and plot the surface $F(\alpha_0, v_0)$ as well as several equally spaced (in numerical value) contours for $0^\circ \leq \alpha_0 \leq 90^\circ$, $0 \leq v_0 \leq 600$ ft/s.

- g. Write an optimization program that starts from an initial point (α_0, v_0) in parameter space and locates an optimum point $\{(\alpha_0)^{\text{opt}}, (v_0)^{\text{opt}}\}$ where the objective function $F(\alpha_0, v_0)$ is a minimum. Fill in [Table E7.6](#):

TABLE E7.6

Initial $\alpha_0(^{\circ})$	Initial v_c (fps)	$(\alpha_0)^{\text{opt}}$	$(v_0)^{\text{opt}}$	Max H	R	$F[(\alpha_0)^{\text{opt}}, (v_0)^{\text{opt}}]$	Number of Iterations
20	400						
65	100						
45	300						
80	500						
5	200						

- 7.7 An alternative method for controlling the heading of a ship is to use rate feedback as shown in [Figure E7.7](#). The saturation block is omitted. Using the same parameter values as in the text,

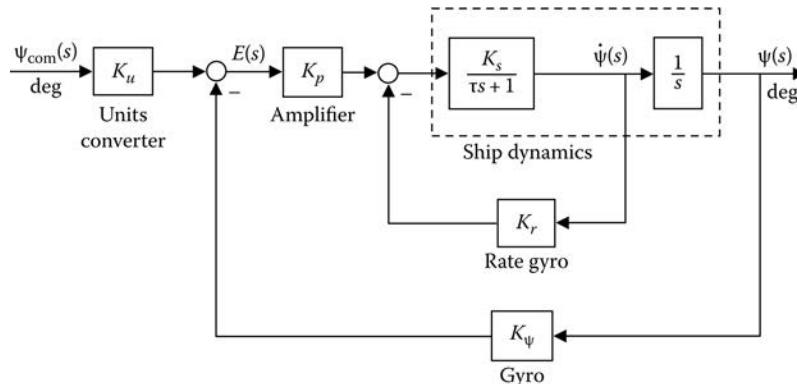


FIGURE E7.7

- a. Find the amplifier gain K_p and rate gyro gain K_r , which minimizes the ISE in response to a step command heading of 5° .
 - b. Repeat part (a) for an IAE objective function.
- 7.8 Repeat the steps in estimating the parameters k and n if the observed chemical concentrations at the end of 5 min intervals are as given in the [Table E7.8](#). Draw a graph similar to the one in [Figure 7.30](#) showing simulated and observed values after 1 min intervals for the first hour.

TABLE E7.8

<i>t</i> (min)	0	5	10	15	20	25	30
\hat{x} (mol/L)	2.0000	1.3333	1.0000	0.8000	0.6667	0.5714	0.5000
<i>t</i> (min)	35	40	45	50	55	60	
\hat{x} (mol/L)	0.4444	0.4000	0.3636	0.3333	0.3077	0.2857	

- 7.9 Suppose the hemispherical tank shown in Figure 7.31 is turned upside down.
- How does the mathematical model of the system change?
 - Modify the Simulink diagram to reflect the new configuration.
 - Show that the surface plot in Figure 7.33 and zero contour plot in Figure 7.36 remain unchanged.
 - Repeat parts (a) and (b) and generate new surface and zero contour plots if the tank is cylindrical with radius $R = 5$ ft.
- 7.10 Compare the execution time required to draw the surface in Figure 7.33 where the objective function is evaluated over a 40-by-40 grid of points in the F_1 - c plane when the numerical integrator is a fixed RK-4 integrator with step size 0.01 s and the default variable-step ode45 (Dormand Prince).
- Hint:* Insert the MATLAB commands “tic” and “toc” at the beginning and end of the MATLAB statements. The execution time will be returned by “toc.”
- 7.11 The objective function defined in Equation 7.77, and shown in Figure 7.32, is modified to

$$F(t_{\text{fill}}) = \begin{cases} A, & 0 \leq t_{\text{fill}} < T_{\min} \\ A \left(\frac{t_{\text{fill}}}{T_L} - 1 \right)^2, & T_{\min} \leq t_{\text{fill}} < T_L \\ 0, & T_L \leq t_{\text{fill}} \leq T_H \\ B \left(\frac{t_{\text{fill}} - T_H}{T_{\max} - T_H} \right)^2, & T_H \leq t_{\text{fill}} < T_{\max} \\ B, & T_{\max} \leq t_{\text{fill}} \end{cases}$$

where T_{\min} is the shortest time possible for a hemispherical tank with radius $R = 7.5$ ft to fill. The controllable parameters are confined to the ranges $0 \leq c \leq 4 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$ and $0 \leq F_1 \leq 20 \text{ ft}^3/\text{min}$. The end points where the objective function is zero are $T_L = 190$ min and $T_H = 210$ min. Finally, $T_{\max} = 500$ min. The numerical values of A and B , the limiting values of the objective function, are $A = 80$ and $B = 40$.

- Find T_{\min} .
 - Generate a new surface plot similar to the one shown in Figure 7.33 for the region $0 \leq F_1 < 20$, $0 < c < 4$.
 - Modify the objective function definition in “Ch7_globe_contours.m,” and plot the contours corresponding to objective function values 0, 10, 20, ..., $(F_0)_{\max}$ where $(F_0)_{\max}$ is the objective function value corresponding to a fill time of T_{\min} .
 - Find several optimum points $(F_1^{\text{opt}}, c^{\text{opt}})$ on the $F(F_1, c) = 0$ contour.
 - Run a simulation of the globe filling with $F_1 = F_1^{\text{opt}}$ and $c = c^{\text{opt}}$ from part (d) and verify that the fill time falls between T_L and T_H .
- 7.12 Write a program to implement a gradient-based search algorithm to find a point (F_1, c) where the objective function is zero. Test the algorithm starting from
- $F_1 = 4 \text{ ft}^3/\text{min}$, $c = 1 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$
 - $F_1 = 7.5 \text{ ft}^3/\text{min}$, $c = 2 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$
 - $F_1 = 1 \text{ ft}^3/\text{min}$, $c = 0 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$

- 7.13 For the hospital occupancy model, do a Monte Carlo simulation and plot the objective function $F(\mu_S)$ for $\mu_S = 15, 16, \dots, 30$ with weights $c_1 = 1, c_2 = 5$. Use baseline values given in the text for the system parameters. Assume the hospital is initially operating at full occupancy.
- 7.14 Suppose the hospital has a holding facility where new patients wait for a bed to become available. Let the state variables in the discrete-time model be $x_B(i)$, the number of patients in rooms with beds at the end of the i th day, and $x_H(i)$, the number waiting for an assigned bed in the holding area at the end of the i th day. Patients are transferred from the holding area to a room with a bed on days when the number of emergency and scheduled arrivals is less than the number of discharged and dying patients. The number of beds is L_B ; the holding area can accommodate L_H patients. Repeat the Monte Carlo simulation described in the text using the baseline values of the system parameters and $L_H = 15$. Plot graphs similar to the ones in Figure 7.46. The weights are $c_1 = c_2 = 1$. Note that the occupancy rate is based on the number of patients with beds, that is, with $L_B = 200$ and $L_H = 15$, the occupancy rate is 90% if $x_B = 180, x_H = 0$, and 100% if $x_B = 200, x_H = 5$.
- 7.15 Investigate the effect of variability in the number of scheduled arrivals on the hospital's occupancy rate. Choose the mean $\mu_S = 21$ scheduled patients per day and simulate the percent occupancy as a function of the standard deviation σ_s where σ_s ranges from zero to three scheduled patients per day.
- 7.16 Use Monte Carlo simulation to obtain an empirical probability density function for random variable Y , the hospital's percent occupancy. Use the following values for the system parameters:

$$\mu_S = 24, \sigma_S^2 = 9, \mu_E = 6, \sigma_E^2 = 4, \mu_D = 28, \sigma_D^2 = 9,$$

$$\mu_M = 2, \sigma_M^2 = 0.25, L = 200, x_0 = 200$$

Hint: Simulate 100 records of sufficient length (in days) to obtain 100 observations $y_1, y_2, y_3, \dots, y_{100}$ where $y_i, i = 1, 2, 3, \dots, 100$ is the percent occupancy corresponding to the i th record. Plot the results

- 7.17 Consider a loan in the amount of P dollars to be repaid in n equal monthly installments of A dollars with interest at i per month. The unpaid balance P_k made after the k th payment is given by

$$P_k = P_{k-1} + iP_{k-1} - A = P_{k-1}(1+i) - A, k = 1, 2, \dots, n$$

A Simulink block diagram is shown in Figure E7.17. Note that the loan amount P is the initial condition of the “Unit Delay” block and the simulation stop time is set to n . Also, be sure to set the “Solver options Type” to “Fixed-step,” “Fixed-step size” to 1, and the integrator to “discrete no continuous states.”

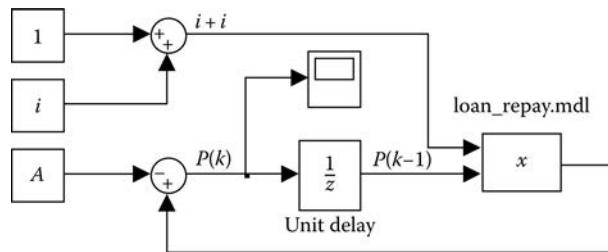


FIGURE E7.17

The terms of a car loan are $P = \$30,000$, $n = 48$ months, and $i = 0.005$ (0.5% per month). For a fixed value of monthly payment A , the unpaid balance at the end of the loan period is P_{48} . Positive values of P_{48} means A is too low and the loan has not been paid off in its entirety.

A negative value of P_{48} implies A is too much and overpayment of the loan has occurred. The correct amount of the monthly payment A to retire the loan after the last (48th) payment is the value of A for which the unpaid balance at the end of the loan period is zero.

- Prepare a graph of P_{48} vs. A, for $A = \$600, \$625, \$650, \dots, \800 . Estimate the correct value of A to repay the loan.
- Write your own or use MATLAB's optimization toolbox to determine the correct A by finding the value of A, which minimizes the objective function P_{48} .
- Plot P_k vs. k , $k = 0, 1, 2, 3, \dots, 48$ using the value of A found in part (b). Compare your answer for A with the correct value of A, which can be obtained from the formula

$$A = p \left[\frac{i(1+i)^n}{(1+i)^n - 1} \right]$$

7.4 LINEARIZATION

[Chapter 4](#) introduced a number of important concepts instrumental in analyzing the behavior of linear systems. By linear systems, we are referring to actual systems modeled by linear algebraic and differential equations. Real-world systems are inherently nonlinear. However, in certain regions, they may respond in a way that a linear model provides an acceptable representation of the system's dynamics. Whenever we employ linear models to describe nonlinear systems, it must be with the understanding that the system remains within its so-called linear-operating region.

Consider the simple mechanical spring shown in [Figure 7.47](#). Its deflection x from equilibrium depends on the magnitude and direction of the applied force F.

Measurements of deflection and force over a range of forces resulting in fracture from excessive compression or elongation produce a graph like the one shown in [Figure 7.48](#).

The linear region of the spring is the section of the operating characteristic where x is proportional to F. Known as Hooke's law, the familiar form is

$$F = kx \quad (7.87)$$

where k is the spring constant, a measure of its stiffness. The linear model in Equation 7.87 is a valid model of the spring provided the applied force is confined to $F_1 \leq F \leq F_2$.

Numerous components behave in a similar fashion. The current in an electrical resistor is assumed proportional to the voltage across its terminals over a range of currents. Conductive heat flow due to a temperature difference between two points and fluid flow caused by pressure differences at

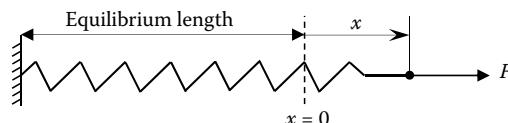


FIGURE 7.47 Deflection of a mechanical spring subjected to an applied force.

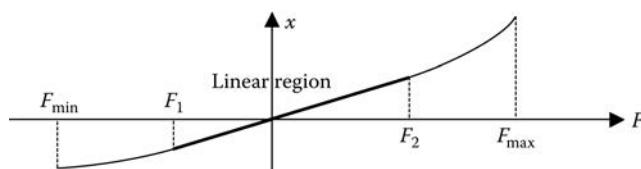


FIGURE 7.48 Operating characteristic of spring showing its linear region.

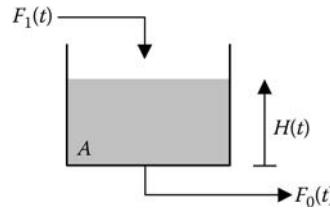


FIGURE 7.49 A liquid tank with input $F_1(t)$ and dependent variables $F_0(t)$ and $H(t)$.

different locations are additional examples of cause-and-effect relationships assumed to be linear over a range of operating conditions.

In the example of the spring, the static operating curve shown in Figure 7.48 can be divided into three distinct regions, that is, points $\{F, x(F)\}$ where

1. $F_{\min} < F < F_1$
2. $F_1 \leq F \leq F_2$ (linear region)
3. $F_2 < F < F_{\max}$

The relation $x = x(F)$ between force and displacement in each region is based on empirical observation as opposed to an analytical model or equation based on scientific principles or natural laws. In contrast, the liquid tank with incompressible fluid shown in Figure 7.49 is modeled by the linear first-order differential equation based on conservation of volume,

$$A \frac{dH}{dt} + F_0 = F_1 \quad (7.88)$$

along with the operating characteristic of the tank, that is, the relationship between the out flow and the liquid level, which applies in both the steady state and otherwise.

$$F_0 = F_0(H) = cH^{1/2}, \quad H \geq 0 \quad (7.89)$$

Equation 7.89 is based on Bernoulli's principle from Physics. The constant c depends on the physical properties of the fluid, tank, and the discharge line.

Equation 7.88 was derived in Section 1.2. The discharge F_0 was assumed proportional to H , that is, $F_0(H) = cH$, resulting in a linear system model of the tank. A “real” tank is nonlinear by virtue of Equation 7.89.

7.4.1 DEVIATION VARIABLES

A linearized tank model can be obtained to approximate the nonlinear tank dynamics. The technique relies upon the concept of an operating point and deviation variables. To illustrate, let us suppose a linearized tank model is required, which provides a reasonable approximation to the nonlinear system provided the inflow, level, and outflow vary only slightly from the steady-state values \bar{F}_1 , \bar{H} , \bar{F}_0 shown in Figure 7.50.

The operating point, for purposes of linearization, is characterized by an inflow \bar{F}_1 and the point (\bar{H}, \bar{F}_0) where

$$\bar{F}_0 = c\bar{H}^{1/2} \quad (7.90)$$

With steady-state conditions at the operating point, $\bar{F}_0 = \bar{F}_1$. From Equation 7.90,

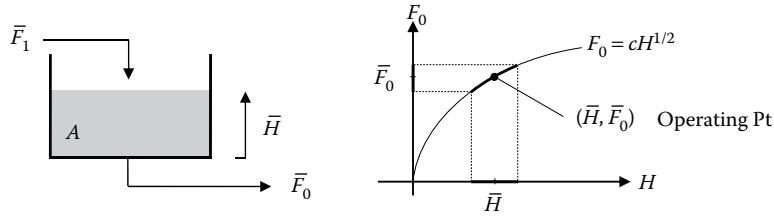


FIGURE 7.50 Operating point (\bar{H}, \bar{F}_0) for tank linearization.

$$\bar{H} = \frac{\bar{F}_0^2}{c^2} = \frac{\bar{F}_1^2}{c^2} \quad (7.91)$$

When the inflow $F_1(t)$ and outputs $H(t)$ and $F_0(t)$ differ from their operating point values, deviation variables $\Delta F_1(t)$, $\Delta H(t)$, and $\Delta F_0(t)$ are introduced according to

$$F_1(t) = \bar{F}_1 + \Delta F_1(t), H(t) = \bar{H} + \Delta H(t), F_0(t) = \bar{F}_0 + \Delta F_0(t) \quad (7.92)$$

Deviation variables relate the differences between actual values of the system variables and their operating point levels, that is,

$$\Delta F_1(t) = F_1(t) - \bar{F}_1, \Delta H(t) = H(t) - \bar{H}, \Delta F_0(t) = F_0(t) - \bar{F}_0 \quad (7.93)$$

Expanding F_0 in Equation 7.89 in a Taylor Series about the operating point (\bar{H}, \bar{F}_0) ,

$$F_0 = \bar{F}_0 + \left. \frac{d}{dH} F_0(H) \right|_{H=\bar{H}} (H - \bar{H}) + \left. \frac{d^2}{dH^2} F_0(H) \right|_{H=\bar{H}} (\bar{H} - \bar{H})^2 + \dots \quad (7.94)$$

$$\Rightarrow F_0 - \bar{F}_0 = \Delta F_0 + \left. \frac{d}{dH} F_0(H) \right|_{H=\bar{H}} \Delta H + \left. \frac{d^2}{dH^2} F_0(H) \right|_{H=\bar{H}} \Delta H^2 + \dots \quad (7.95)$$

If ΔH is small in absolute value, then the ΔH^2 term and all succeeding terms are higher order terms that can be ignored (to a first-order approximation). The result is a first-order Taylor Series approximation for the deviation flow ΔF_0 , namely,

$$\Delta F_0 \approx \left. \frac{d}{dH} F_0(H) \right|_{H=\bar{H}} \Delta H = F'_0(\bar{H}) \Delta H \quad (7.96)$$

Differentiating Equation 7.89 to find the first derivative $F'_0(H)$ and evaluating the result at $H = \bar{H}$ lead to

$$\Delta F_0 \approx \frac{1}{2} c \bar{H}^{-1/2} \Delta H \quad (7.97)$$

where the accuracy depends on the magnitude of ΔH (more about this point later).

Substituting expressions in Equation 7.92 for $F_1(t)$, $H(t)$, and $F_0(t)$ into Equation 7.88 gives

$$A \frac{d}{dt} [\bar{H} + \Delta H(t)] + \bar{F}_0 + \Delta F_0(t) = \bar{F}_1 + \Delta F_1(t) \quad (7.98)$$

$$\Rightarrow A \frac{d}{dt} \bar{H} + A \frac{d}{dt} \Delta H(t) + \bar{F}_0 + \Delta F_0(t) = \bar{F}_1 + \Delta F_1(t) \quad (7.99)$$

Knowing $\bar{F}_0 = \bar{F}_1$ and the fact that $A(d/dt) \bar{H} = 0$ leads to

$$A \frac{d}{dt} \Delta H(t) + \Delta F_0(t) = \Delta F_1(t) \quad (7.100)$$

Substituting the approximation in Equation 7.96 for $\Delta F_0(t)$ into Equation 7.100 results in the first-order linearized differential equation model

$$A \frac{d}{dt} \Delta H(t) + F'_0(\bar{H}) \Delta H(t) = \Delta F_1(t) \quad (7.101)$$

The nonlinear-operating characteristic for the tank, Equation 7.89, has been approximated by the linear relationship of Equation 7.96, which can be written as

$$F_0 = \bar{F}_0 + F'_0(\bar{H})(H - \bar{H}) \quad (7.102)$$

Equation 7.102 is the equation of the line tangent to the curve $F_0 = F_0(H)$ at the operating point (\bar{H}, \bar{F}_0) . **Figure 7.51** illustrates the case when the tank constant $c = 0.5 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$ and the operating point $(\bar{H}, \bar{F}_0) = (9 \text{ ft}, 1.5 \text{ ft}^3/\text{min})$. Note that (\bar{H}, \bar{F}_0) is the origin in a new coordinate system with ΔH in the horizontal direction and ΔF_0 in the vertical direction.

Before we generalize the procedure for linearization of certain types of nonlinearities, we illustrate, through the next example, a case where the nonlinear term in the system model is a product of dependent variables.

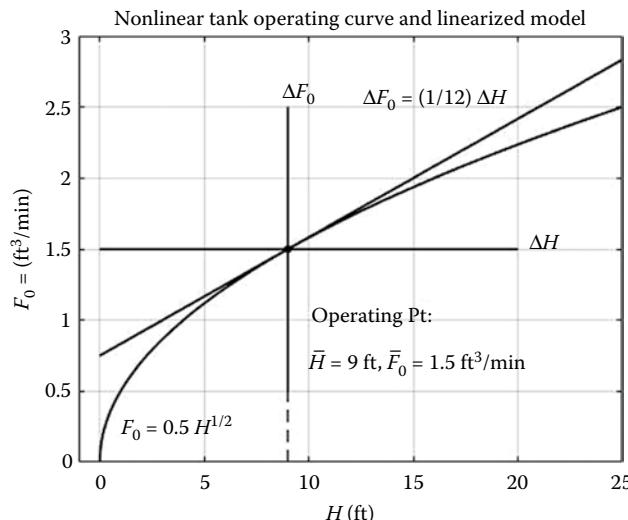


FIGURE 7.51 Nonlinear tank-operating curve and linearized approximation.

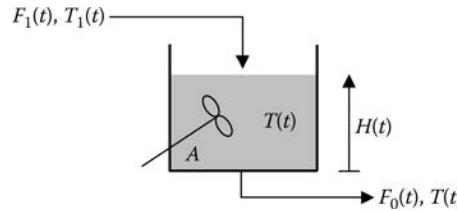


FIGURE 7.52 Stirred tank with inputs $F_1(t)$, $T_1(t)$ and dependent variables $H(t)$, $F_0(t)$, $T(t)$.

Consider the well-stirred tank in [Figure 7.52](#). The temperature of the liquid $T(t)$ as well as its level $H(t)$ is of interest. Accordingly, a second equation is required, one that introduces the additional dependent variable $T(t)$.

The rate at which energy is stored in the liquid holdup is equal to the difference in the rate of energy flowing in and out of the tank. If we substitute the word “mass” for “energy,” we have the principle of conservation of mass, which led to the differential equation for the tank level in [Equation 7.88](#). Applying the principle of conservation of energy in equation form gives

$$\frac{d}{dt}(c_p \gamma VT) = c_p \gamma F_1 T_1 - c_p \gamma F_0 T \quad (7.103)$$

where

$T(t)$ is the uniform liquid temperature in tank, °F

$F_1(t)$ is the input flow rate, ft^3/min

$T_1(t)$ is the liquid temperature entering tank, °F

$F_0(t)$ is the output flow rate, ft^3/min

V is the volume of liquid in tank, ft^3

c_p is the specific heat of liquid ($\text{Btu/lb}\cdot^\circ\text{F}$)

γ is the specific weight of liquid (lb/ft^3)

The left-hand side accounts for the energy accumulation, and the right-hand side represents the difference in energy flows in the two streams. Replacing the tank volume V with the product AH in [Equation 7.103](#) results in

$$A \frac{d}{dt}(HT) + F_0 T = F_1 T_1 \quad (7.104)$$

$$\Rightarrow AH \frac{dT}{dt} + AT \frac{dH}{dt} + F_0 T = F_1 T_1 \quad (7.105)$$

[Equations 7.88](#), [7.89](#), and [7.105](#) comprise the nonlinear mathematical model of the system. [Figure 7.53](#) illustrates the presence of two inputs (independent variables) and three dependent variables. The state variables are $T(t)$ and either $H(t)$ or $F_0(t)$, but not both since they are related algebraically according to [Equation 7.89](#).

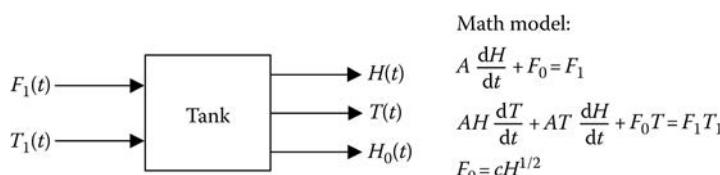


FIGURE 7.53 Nonlinear system: tank with two inputs and three dependent variables.

A steady-state operating point is established where $F_i(t) = \bar{F}_i$ and $T_i(t) = \bar{T}_i$ with dependent variables $H(t) = \bar{H}$, $F_0(t) = \bar{F}_0$, and $T(t) = \bar{T}$. Introducing deviation variables

$$\Delta T = T(t) - \bar{T}, \quad \Delta T_i = T_i(t) - \bar{T}_i \quad (7.106)$$

Equation 7.105 becomes

$$\begin{aligned} A(\bar{H} + \Delta H) \frac{d}{dt}(\bar{T} + \Delta T) + A(\bar{T} + \Delta T) \frac{d}{dt}(\bar{H} + \Delta H) + (\bar{F}_0 + \Delta F_0)(\bar{T} + \Delta T) \\ = (\bar{F}_i + \Delta F_i)(\bar{T}_i + \Delta T_i) \end{aligned} \quad (7.107)$$

Deviation variables are assumed to be small in magnitude, and, therefore, the products $\Delta H \frac{d}{dt} \Delta T$, $\Delta T \frac{d}{dt} \Delta H$, $\Delta F_0 \Delta T$, and $\Delta F_i \Delta T_i$ are negligible by comparison. Equation 7.107 simplifies to

$$A\bar{H} \frac{d}{dt} \Delta T + A\bar{T} \frac{d}{dt} \Delta H + \bar{F}_0 \bar{T} + \bar{F}_0 \Delta T + \bar{T} \Delta F_0 = \bar{F}_i \bar{T}_i + \bar{F}_i \Delta T_i + \bar{T}_i \Delta F_i \quad (7.108)$$

Substituting AF_0 from Equation 7.96 into Equation 7.108 and rearranging terms give

$$A\bar{H} \frac{d\Delta T}{dt} + A\bar{T} \frac{d\Delta H}{dt} + \bar{F}_0 \Delta T + \bar{T} F'_0(\bar{H}) \Delta H = \bar{F}_i \bar{T}_i - \bar{F}_0 \bar{T} + \bar{F}_i \Delta T_i + \bar{T}_i \Delta F_i \quad (7.109)$$

Recognizing that $\bar{F}_0 = \bar{F}_i$ and $\bar{T} = \bar{T}_i$ at the steady-state operating point, Equation 7.109 reduces to

$$A\bar{H} \frac{d\Delta T}{dt} + A\bar{T} \frac{d\Delta H}{dt} + \bar{F}_0 \Delta T + \bar{T} F'_0(\bar{H}) \Delta H = \bar{F}_i \Delta T_i + \bar{T}_i \Delta F_i \quad (7.110)$$

Equations 7.101 and 7.110 are coupled linearized differential equations of the tank. It is left as an exercise problem to show that the state derivatives are expressible as

$$\begin{bmatrix} \frac{d}{dt} \Delta H \\ \frac{d}{dt} \Delta T \end{bmatrix} = \begin{bmatrix} -\frac{F'_0(\bar{H})}{A} & 0 \\ 0 & -\frac{\bar{F}_0}{A\bar{H}} \end{bmatrix} \begin{bmatrix} \Delta H \\ \Delta T \end{bmatrix} + \begin{bmatrix} \frac{1}{A} & 0 \\ 0 & \frac{\bar{F}_0}{A\bar{H}} \end{bmatrix} \begin{bmatrix} \Delta F_i \\ \Delta T_i \end{bmatrix} \quad (7.111)$$

Simulation is an effective way to appreciate the limitations of a linearized model. The following example illustrates the point.

EXAMPLE 7.2

The tank shown in Figure 7.52 with cross-sectional area $A = 100 \text{ ft}^2$ is initially in equilibrium with $\bar{F}_i = \bar{F}_0 = 25 \text{ ft}^3/\text{min}$, $\bar{H} = 9 \text{ ft}$, and $\bar{T}_i = \bar{T} = 150^\circ\text{F}$. The input flow and temperature profiles are shown in Figure 7.54.

- Simulate the transient response of the nonlinear model when $\alpha = \beta = 0.1$.
- Repeat part (a) using the linear state model in Equation 7.111.
- Compare the nonlinear and linearized responses and comment on the results.

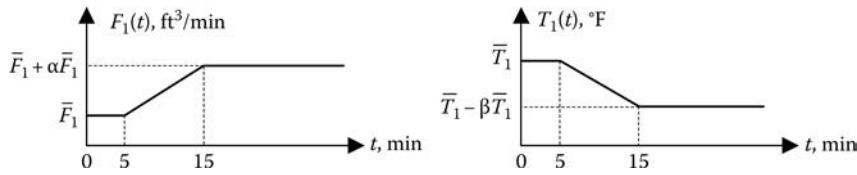


FIGURE 7.54 Tank system input profiles.

- d. Find the time constants τ_H and τ_T of the linearized system and show that $\tau_H = 2\tau_T$.
 - e. Find expressions for $H(\infty)$ and $T(\infty)$ in response to constant inputs $F_1(t) = \hat{F}_1$, $t \geq 0$ and $T_1(t) = \hat{T}_1$, $t \geq 0$ based on the nonlinear model. Compute the numerical values for $H(\infty)$ and $T(\infty)$ when $\hat{F}_1 = (1+\alpha)\bar{F}_1$ and $\hat{T}_1 = (1-\beta)\bar{T}_1$.
 - f. Repeat part (e) using the linearized model and compare the results.
- a. [Figure 7.55](#) is a Simulink diagram for simulating the dynamic response of the nonlinear and linearized system models.
- The level H and temperature T of the nonlinear system are plotted in [Figure 7.56](#). The transient period is on the order of 300 min for level and 200 min for temperature.
- b. The linearized system outputs are shown in [Figure 7.57](#).
 - c. The nonlinear and linearized transient responses for level and temperature are compared in [Figure 7.58](#). For a 10% increase in inlet flow rate above \bar{F}_1 and a 10% decrease in inlet temperature below \bar{T}_1 , the nonlinear system and linearized system approximation exhibit nearly identical transient responses.
 - d. Referring to Equation 7.101, the time constant τ_H of the linearized system is

$$\tau_H = \frac{A}{F'_0(\bar{H})} = \frac{A}{(1/2c\bar{H})^{-1/2}} = \frac{2A\bar{H}^{1/2}}{c} \quad (7.112)$$

Solving for c in Equation 7.90 and substituting the result into Equation 7.112 give

$$\tau_H = \frac{2A\bar{H}^{1/2}}{\bar{F}_0\bar{H}^{-1/2}} = \frac{2A\bar{H}}{\bar{F}_0} = \frac{2(100)(9)}{25} = 72 \text{ min} \quad (7.113)$$

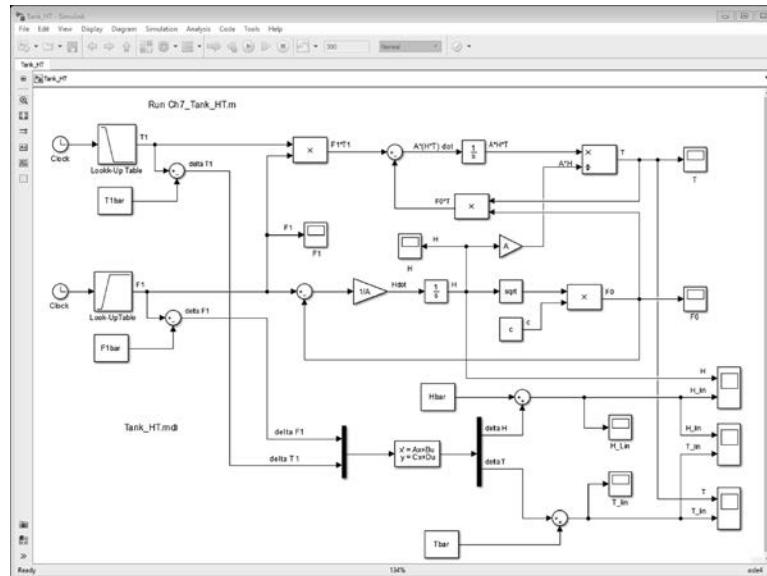


FIGURE 7.55 Simulink diagram for simulation of nonlinear and linearized system.

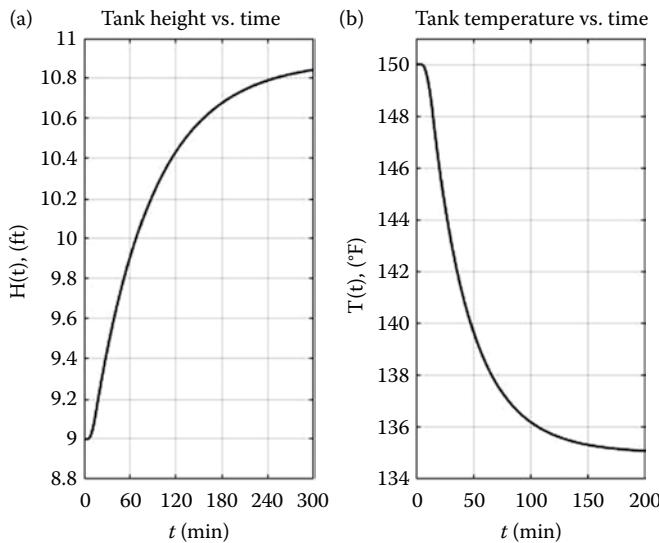


FIGURE 7.56 Nonlinear system response of (a) level and (b) temperature.

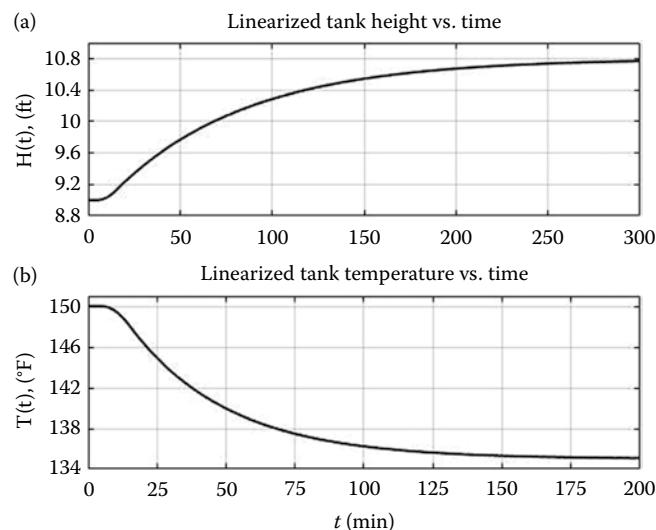


FIGURE 7.57 Linearized system (a) level and (b) temperature transient response ($\alpha = \beta = 0.1$).

From the second of the two state equations in Equation 7.111, the time constant τ_T is

$$\tau_T = \frac{A\bar{H}}{\bar{F}_0} = \frac{100(9)}{25} = 36 \text{ min} \quad (7.114)$$

$\tau_H = 2\tau_T$ follows directly from Equations 7.113 and 7.114.

- e. At steady state ($t = \infty$), dH/dt and dT/dt are zero. Setting dH/dt equal to zero in Equation 7.88 gives

$$F_0(\infty) = F_1(\infty) = \hat{F} \quad (7.115)$$

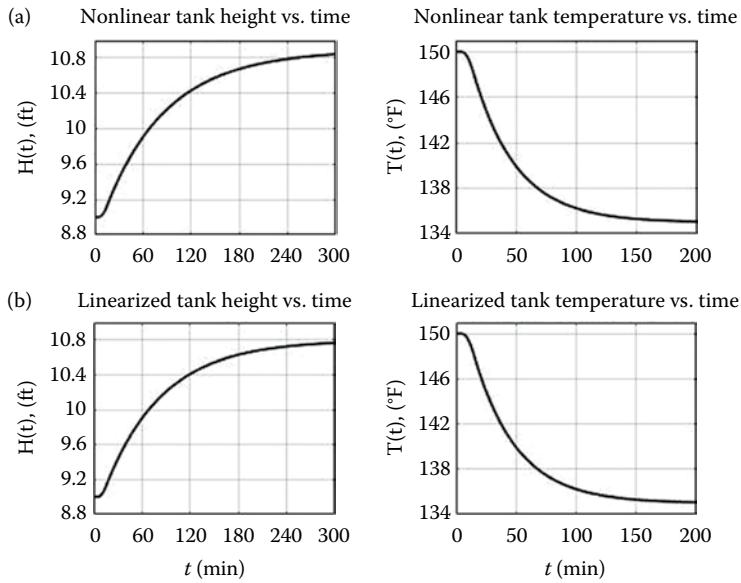


FIGURE 7.58 Comparison of (a) nonlinear and (b) linearized system transient responses.

According to the tank-operating characteristic (Equation 7.89),

$$F_0(\infty) = c[H(\infty)]^{1/2} \quad (7.116)$$

Solving for $H(\infty)$ in Equation 7.116,

$$H(\infty) = \left[\frac{F_0(\infty)}{c} \right]^2 = \left[\frac{\hat{F}_1}{c} \right]^2 \quad (7.117)$$

Setting $(dH/dt) = (dT/dt) = 0$ in Equation 7.105 gives

$$F_0(\infty)T(\infty) = F_1(\infty)T_1(\infty) \Rightarrow T(\infty) = T_1(\infty) = \hat{T}_1 \quad (7.118)$$

The tank constant c is obtained from the given operating point conditions, that is,

$$c = \frac{\bar{F}_0}{\bar{H}^{1/2}} = \frac{25}{9^{1/2}} = \frac{25}{3} \text{ ft}^3/\text{min}/\text{ft}^{1/2} \quad (7.119)$$

The numerical values of $H(\infty)$ and $T(\infty)$ are

$$H(\infty) = \left[\frac{\hat{F}_1}{c} \right]^2 = \left[\frac{(1+\alpha)\bar{F}_1}{c} \right]^2 = \left[\frac{1.1(25)}{25/3} \right]^2 = 10.89 \text{ ft} \quad (7.120)$$

$$T(\infty) = \hat{T}_1 = (1-\beta)\bar{T}_1 = 0.9(150) = 135^\circ\text{F}$$

f. Setting $(d/dt)\Delta H = (d/dt)\Delta T = 0$ in Equation 7.111 and solving for $\Delta H(\infty)$ and $\Delta T(\infty)$ give

$$\begin{bmatrix} \Delta H(\infty) \\ \Delta H(\infty) \end{bmatrix} = - \begin{bmatrix} -\frac{F'_0(\bar{H})}{A} & 0 \\ 0 & -\frac{\bar{F}_0}{A\bar{H}} \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{A} & 0 \\ 0 & -\frac{\bar{F}_0}{A\bar{H}} \end{bmatrix} \begin{bmatrix} \Delta F_1 \\ \Delta T_1 \end{bmatrix} \quad (7.121)$$

$$= - \begin{bmatrix} -\frac{A}{F'_0(\bar{H})} & 0 \\ 0 & -\frac{A\bar{H}}{\bar{F}_0} \end{bmatrix} \begin{bmatrix} \frac{1}{A} & 0 \\ 0 & -\frac{\bar{F}_0}{A\bar{H}} \end{bmatrix} \begin{bmatrix} \Delta F_1 \\ \Delta T_1 \end{bmatrix} \quad (7.122)$$

$$= \begin{bmatrix} \frac{1}{F'_0(\bar{H})} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta F_1 \\ \Delta T_1 \end{bmatrix} = \begin{bmatrix} \frac{1}{F'_0(\bar{H})} \Delta F \\ \Delta T_1 \end{bmatrix} \quad (7.123)$$

The slope $F'_0(H)$ is obtained by differentiation of Equation 7.89 followed by substitution of the values $c = 25/3 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$ and $\bar{H} = 9 \text{ ft}$. The result is $F'_0(\bar{H}) = 25/18 \text{ ft}^3/\text{min}/\text{ft}$. The deviation variables at steady state are

$$\Delta H(\infty) = \frac{1}{F'_0(\bar{H})} \Delta F = \frac{1}{F'_0(\bar{H})} \alpha \bar{F}_1 = \frac{1}{25/18} (0.1)(25) = 1.8 \text{ ft} \quad (7.124)$$

$$\Delta T(\infty) = \Delta T_1 = -\beta \bar{T}_1 = -0.1(150) = -15^\circ \text{F} \quad (7.125)$$

The steady-state level and temperature from the linearized model are

$$H(\infty) = \bar{H} + \Delta H(\infty) = 10.8 \text{ ft} \quad (7.126)$$

$$T(\infty) = \bar{T} + \Delta T(\infty) = 135^\circ \text{F} \quad (7.127)$$

The steady-state level based on the linearized system model differs by 0.09 ft from the value based on the nonlinear model. The steady-state temperatures are the same from both the nonlinear and linearized system models.

7.4.2 LINEARIZATION OF NONLINEAR SYSTEMS IN STATE VARIABLE FORM

The starting point is a nonlinear system model

$$\dot{\underline{x}} = \underline{f}(t, \underline{x}, \underline{u}) \quad (7.128)$$

$$\underline{y} = \underline{g}(t, \underline{x}, \underline{u}) \quad (7.129)$$

where

$\underline{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ is the $n \times 1$ state vector

$\underline{y} = [y_1 \ y_2 \ \dots \ y_p]^T$ is a $p \times 1$ vector of outputs

$\underline{u} = [u_1 \ u_2 \ \dots \ u_m]^T$ is the $m \times 1$ input vector

t is time

Equations 7.128 and 7.129 are short for

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(t, \underline{x}, \underline{u}) \\ f_2(t, \underline{x}, \underline{u}) \\ \vdots \\ f_n(t, \underline{x}, \underline{u}) \end{bmatrix}, \quad \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} = \begin{bmatrix} g_1(t, \underline{x}, \underline{u}) \\ g_2(t, \underline{x}, \underline{u}) \\ \vdots \\ g_p(t, \underline{x}, \underline{u}) \end{bmatrix} \quad (7.130)$$

The objective is to linearize Equations 7.128 and 7.129 about a nominal operating point in the state space $\underline{x}^0 = [x_1^0 \ x_2^0 \ \dots \ x_n^0]^T$ for a given (usually constant) input vector $\underline{u}^0 = [u_1^0 \ u_2^0 \ \dots \ u_m^0]^T$. The first-order Taylor Series approximation of the function $f_i(t, x, u)$ about the point (x^0, u^0) is given by

$$\begin{aligned} \dot{x}_1 &= f_1(t, \underline{x}^0, \underline{u}^0) + \frac{\partial}{\partial x_1} f_1(t, \underline{x}^0, \underline{u}^0)(x_1 - x_1^0) + \frac{\partial}{\partial u_1} f_1(t, \underline{x}^0, \underline{u}^0)(u_1 - u_1^0) + \dots \\ &\quad + \frac{\partial}{\partial x_n} f_1(t, \underline{x}^0, \underline{u}^0)(x_n - x_n^0) + \frac{\partial}{\partial u_1} f_1(t, \underline{x}^0, \underline{u}^0)(u_1 - u_1^0) \\ &\quad + \frac{\partial}{\partial u_2} f_1(t, \underline{x}^0, \underline{u}^0)(u_2 - u_2^0) + \dots + \frac{\partial}{\partial u_m} f_1(t, \underline{x}^0, \underline{u}^0)(u_m - u_m^0) \end{aligned} \quad (7.131)$$

Similar relations hold for $\dot{x}_2, \dots, \dot{x}_n$. Introducing deviation variables

$$\begin{aligned} \Delta x_1 &= x_1 - x_1^0, \quad \Delta x_2 = x_2 - x_2^0, \dots, \Delta x_n = x_n - x_n^0 \\ \Delta u_1 &= u_1 - u_1^0, \quad \Delta u_2 = u_2 - u_2^0, \dots, \Delta u_m = u_m - u_m^0 \end{aligned}$$

leads to the linearized approximation of Equation 7.128 by

$$\Delta \dot{\underline{x}} = A \Delta \underline{x} + B \Delta \underline{u} \quad (7.132)$$

where

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\underline{x}^0, \underline{u}^0) & \frac{\partial f_1}{\partial x_2}(\underline{x}^0, \underline{u}^0) & \dots & \frac{\partial f_1}{\partial x_n}(\underline{x}^0, \underline{u}^0) \\ \frac{\partial f_2}{\partial x_1}(\underline{x}^0, \underline{u}^0) & \frac{\partial f_2}{\partial x_2}(\underline{x}^0, \underline{u}^0) & \dots & \frac{\partial f_2}{\partial x_n}(\underline{x}^0, \underline{u}^0) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\underline{x}^0, \underline{u}^0) & \frac{\partial f_n}{\partial x_2}(\underline{x}^0, \underline{u}^0) & \dots & \frac{\partial f_n}{\partial x_n}(\underline{x}^0, \underline{u}^0) \end{bmatrix} \quad (7.133)$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\underline{x}^0, \underline{u}^0) & \frac{\partial f_1}{\partial u_2}(\underline{x}^0, \underline{u}^0) & \dots & \frac{\partial f_1}{\partial u_m}(\underline{x}^0, \underline{u}^0) \\ \frac{\partial f_2}{\partial u_1}(\underline{x}^0, \underline{u}^0) & \frac{\partial f_2}{\partial u_2}(\underline{x}^0, \underline{u}^0) & \dots & \frac{\partial f_2}{\partial u_m}(\underline{x}^0, \underline{u}^0) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1}(\underline{x}^0, \underline{u}^0) & \frac{\partial f_n}{\partial u_2}(\underline{x}^0, \underline{u}^0) & \dots & \frac{\partial f_n}{\partial u_m}(\underline{x}^0, \underline{u}^0) \end{bmatrix} \quad (7.134)$$

and

$$\begin{aligned}\Delta \underline{\dot{x}} &= [\Delta \dot{x}_1 \ \Delta \dot{x}_2 \dots \Delta \dot{x}_n]^T \\ \Delta \underline{x} &= [\Delta x_1 \ \Delta x_2 \dots \Delta x_n]^T \\ \Delta \underline{u} &= [\Delta u_1 \ \Delta u_2 \dots \Delta u_m]^T\end{aligned}$$

The combined matrix $[A|B]$ of all partials is called the Jacobian matrix of the vector function $f(t, \underline{x}, \underline{u})$ defining the state derivatives. In similar fashion, the linearized approximation to Equation 7.129 is given by

$$\Delta \underline{y} = C \Delta \underline{x} + D \Delta \underline{u} \quad (7.135)$$

where

$$\Delta \underline{y} = [\Delta y_1 \ \Delta y_2 \dots \Delta y_p]^T = [y_1 - y_1^0 \ y_2 - y_2^0 \dots y_p - y_p^0]^T$$

and

$$y_i^0 = g_i(\underline{x}^0, \underline{u}^0), i = 1, 2, \dots, p \quad (7.136)$$

C and D are matrix of partials with components

$$c_{ij} = \frac{\partial g_i}{\partial x_j}(\underline{x}^0, \underline{u}^0), i = 1, 2, \dots, p, j = 1, 2, \dots, n \quad (7.137)$$

$$d_{ij} = \frac{\partial g_i}{\partial u_j}(\underline{x}^0, \underline{u}^0), i = 1, 2, \dots, p, j = 1, 2, \dots, m \quad (7.138)$$

To illustrate the process of linearizing a nonlinear state variable model, consider the inverted pendulum previously introduced in Section 5.4, redrawn in [Figure 7.59](#).

The coupled nonlinear differential equations describing the system (Equations 5.50 and 5.51) can be manipulated to read

$$\ddot{x} = \frac{ml\dot{\theta}^2 \sin \theta - (mg/2)\sin 2\theta + u}{M + m \sin^2 \theta} \quad (7.139)$$

$$\ddot{\theta} = \frac{-(ml/2)\dot{\theta}^2 \sin 2\theta - (m+M)g \sin \theta - u \cos \theta}{l(M + m \sin^2 \theta)} \quad (7.140)$$

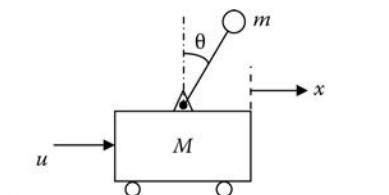


FIGURE 7.59 A nonlinear system: the inverted pendulum.

State variables are x_1, x_2, x_3, x_4 where $x_1 = x, x_2 = \dot{x}, x_3 = \theta, x_4 = \dot{\theta}$. The state derivatives are given by

$$\dot{x}_1 = f_1(\underline{x}, \underline{u}) = x_2 \quad (7.141)$$

$$\dot{x}_2 = f_2(\underline{x}, \underline{u}) = \frac{mlx_4^2 \sin x_3 - (mg/2) \sin 2x_3 + u}{M + m \sin^2 x_3} \quad (7.142)$$

$$\dot{x}_3 = f_3(\underline{x}, \underline{u}) = x_4 \quad (7.143)$$

$$\dot{x}_4 = f_4(\underline{x}, \underline{u}) = \frac{-(ml/2)x_4^2 \sin 2x_3 + (m+M)g \sin x_3 - u \cos x_3}{l(M + m \sin^2 x_3)} \quad (7.144)$$

Choosing the outputs as x and θ ,

$$y_1 = g_1(\underline{x}, \underline{u}) = x_1 \quad (7.145)$$

$$y_2 = g_2(\underline{x}, \underline{u}) = x_3 \quad (7.146)$$

Components of the linearized system matrices A, B, C , and D consist of the partials

$$\begin{aligned} a_{11} &= \frac{\partial f_1}{\partial x_1}(\underline{x}^0, \underline{u}^0) = 0, & a_{12} &= \frac{\partial f_1}{\partial x_2}(\underline{x}^0, \underline{u}^0) = 1, \\ a_{13} &= \frac{\partial f_1}{\partial x_3}(\underline{x}^0, \underline{u}^0) = a_{14} = \frac{\partial f_1}{\partial x_4}(\underline{x}^0, \underline{u}^0) = 0 \end{aligned} \quad (7.147)$$

$$a_{21} = \frac{\partial f_2}{\partial x_1}(\underline{x}^0, \underline{u}^0) = 0, \quad a_{22} = \frac{\partial f_2}{\partial x_2}(\underline{x}^0, \underline{u}^0) = 0, \quad a_{24} = \frac{\partial f_2}{\partial x_4}(\underline{x}^0, \underline{u}^0) = \frac{2mlx_4 \sin x_3}{M + m \sin^2 x_3} \quad (7.148)$$

The component a_{23} is equal to N_1/D_1 evaluated at the operating point $(\underline{x}^0, \underline{u}^0)$ where

$$\begin{aligned} N_1 &= (M + m \sin^2 x_3) \left[mlx_4^2 \cos x_3 - mg(1 - 2 \sin^2 x_3) \right] \\ &\quad - \left[mlx_4^2 \sin x_3 - \frac{mg}{2} \sin 2x_3 + u \right] (m \sin 2x_3) \end{aligned} \quad (7.149)$$

$$D_1 = (M + m \sin^2 x_3)^2 \quad (7.150)$$

$$a_{31} = \frac{\partial f_3}{\partial x_1}(\underline{x}^0, \underline{u}^0) = a_{32} = \frac{\partial f_3}{\partial x_2}(\underline{x}^0, \underline{u}^0) = a_{33} \frac{\partial f_3}{\partial x_3}(\underline{x}^0, \underline{u}^0) = 0, \quad a_{34} = \frac{\partial f_3}{\partial x_4}(\underline{x}^0, \underline{u}^0) = 1 \quad (7.151)$$

$$a_{41} = \frac{\partial f_4}{\partial x_1}(\underline{x}^0, \underline{u}^0) = 0, \quad a_{42} = \frac{\partial f_4}{\partial x_2}(\underline{x}^0, \underline{u}^0) = 0, \quad a_{44} \frac{\partial f_4}{\partial x_4}(\underline{x}^0, \underline{u}^0) = \frac{-mlx_4 \sin 2x_3}{l(M + m \sin^2 x_3)} \quad (7.152)$$

The component a_{43} is equal to N_2/D_2 evaluated at the operating point $(\underline{x}^0, \underline{u}^0)$ where

$$\begin{aligned} N_2 - l(M + m \sin^2 x_3) & \left[-mlx_4^2(1 - 2 \sin^2 x_3) + (m + M)g \cos x_3 + u \sin x_3 \right] \\ & + \left[\frac{m}{2} lx_4^2 \sin^2 2x_3 - (m + M)g \sin x_3 + u \cos x_3 \right] lm \sin 2x_3 \end{aligned} \quad (7.153)$$

$$D_2 = \left[l(M + m \sin^2 x_3) \right]^2 \quad (7.154)$$

$$b_{11} = \frac{\partial f_1}{\partial u}(\underline{x}^0, \underline{u}^0) = 0, \quad b_{21} = \frac{\partial f_2}{\partial u}(\underline{x}^0, \underline{u}^0) = \frac{1}{M + m \sin^2 x_3} \quad (7.155)$$

$$b_{31} = \frac{\partial f_3}{\partial u}(\underline{x}^0, \underline{u}^0) = 0, \quad b_{41} = \frac{\partial f_4}{\partial u}(\underline{x}^0, \underline{u}^0) = \frac{-\cos x_3}{l(M + m \sin^2 x_3)} \quad (7.156)$$

$$c_{11} = \frac{\partial g_1}{\partial x_1}(\underline{x}^0, \underline{u}^0) = 1, \quad c_{12} = \frac{\partial g_1}{\partial x_2}(\underline{x}^0, \underline{u}^0) = 0, \quad c_{13} \frac{\partial g_1}{\partial x_3}(\underline{x}^0, \underline{u}^0) = c_{14} = \frac{\partial g_1}{\partial x_4}(\underline{x}^0, \underline{u}^0) = 0 \quad (7.157)$$

$$c_{21} = \frac{\partial g_2}{\partial x_1}(\underline{x}^0, \underline{u}^0) = c_{22} = \frac{\partial g_2}{\partial x_2}(\underline{x}^0, \underline{u}^0) = 0, \quad c_{23} \frac{\partial g_2}{\partial x_3}(\underline{x}^0, \underline{u}^0) = 1, \quad c_{24} = \frac{\partial g_2}{\partial x_4}(\underline{x}^0, \underline{u}^0) = 0 \quad (7.158)$$

$$d_{11} = \frac{\partial g_1}{\partial u}(\underline{x}^0, \underline{u}^0) = d_{21} = \frac{\partial g_2}{\partial u}(\underline{x}^0, \underline{u}^0) = 0 \quad (7.159)$$

Suppose the steady-state operating point is $\underline{x}^0 = [0 \ 0 \ \pi \ 0]^T$ and input $\underline{u}^0 = 0$. The nonzero elements of matrices A , B , C , and D are

$$\begin{aligned} a_{12} &= 1, \quad a_{23} = -\frac{m}{M} g, \quad a_{34} = 1, \quad a_{43} = -\frac{g}{l} \frac{(m + M)}{M}, \\ b_{21} &= \frac{1}{M}, \quad b_{41} = \frac{1}{lM}, \quad c_{11} = c_{23} = 1 \end{aligned} \quad (7.160)$$

For $M = 3$ kg, $m = 0.1$ kg, $l = 0.75$ m, $g = 9.8$ m/s², the system matrices are

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -0.3267 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -13.5022 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0.333 \\ 0 \\ 0.444 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (7.161)$$

7.4.3 LINMOD FUNCTION

Simulink estimates the matrices A , B , C , and D in the linearized approximation by using small perturbations in the state and input(s) to numerically calculate the partial derivatives. The “linmod”

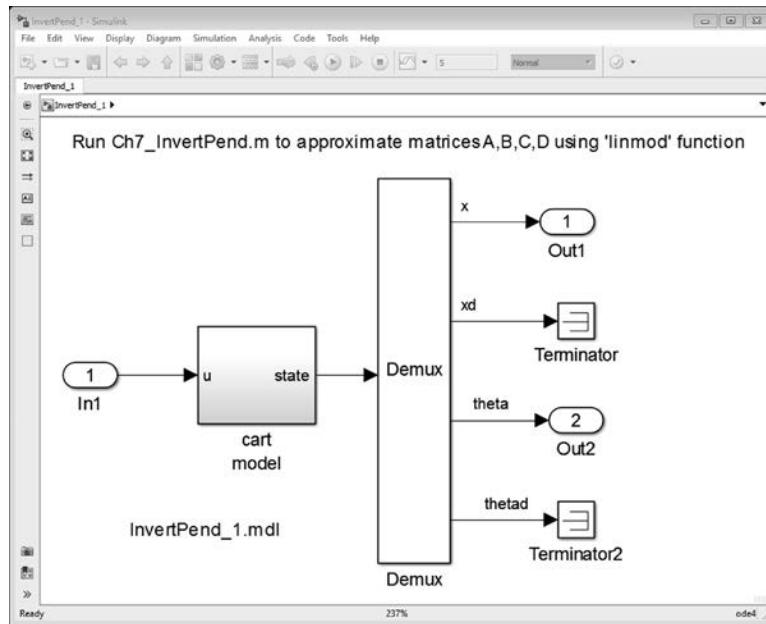


FIGURE 7.60 Simulink model of inverted pendulum showing input and two outputs.

and “`linmod`” functions extract the linearized model coefficient matrices from a Simulink diagram of the nonlinear system.

The top level of a Simulink simulation of the inverted pendulum is shown in [Figure 7.60](#).

The “`cart`” subsystem is shown in [Figure 7.61](#).

The MATLAB statement

```
[sizes, x0, states] = InvertPend_1 ([] , [] , [] , 0) ;
```

in M-file “`Ch7_InvertPend.m`” returns the following information:

sizes = 4	x0 = 0
0	3.1416
2	0
1	0
0	
1	
1	

```
states = 'InvertPend_1/cartmodel/x Integrator'
'InvertPend_1/cart model/theta Integrator'
'InvertPend_1/cart model/xd Integrator'
'InvertPend_1/cart model/thetad Integrator'
```

The first four components of the output ‘`sizes`’ reveal the number of continuous states (4), discrete states (0), outputs (2), and inputs (1) in the Simulink model. The ordering of the states is conveyed by the output vector “`states`,” which in the present case is seen to be “`x`,” “`theta`,” “`xd`,” and “`thetad`.`”` “`X0`” reports the initial values of the state vector in the order defined by the output “`states`.`”` It will soon become apparent why the ordering of the state vector is significant.

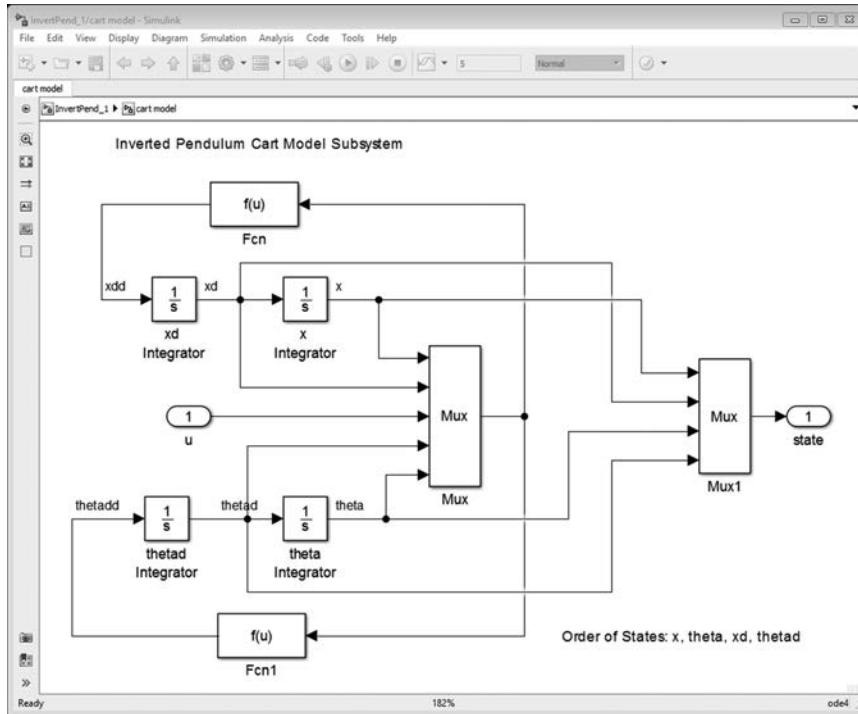


FIGURE 7.61 Cart subsystem showing internal states.

The same M-file “*Ch7_InvertPend.m*” contains the statement

```
[A2, B2, C2, D2] = linmod('InvertPend_1', x_operpt, u0)
```

which returns the linearized system matrices. The first argument “*InvertPend_1*” is the Simulink model file name, while “*x_operpt*” and “*u0*” are arrays with numerical values of the state and input at the operating point.

The “*linmod*” function returns the matrices

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -0.3267 & 0 & 1 \\ 0 & -13.5022 & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0 \\ 0.333 \\ 0.444 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (7.162)$$

The matrices in Equations 7.161 and 7.162 are different as a result of the difference in the ordering of the state vector in the two different linearized models of the system. That is, from Equation 7.161, when the state is $[\Delta x \Delta \dot{x} \Delta \theta \Delta \ddot{\theta}]^T$, we have

$$\begin{bmatrix} \Delta \dot{x}_1 \\ \Delta \dot{x}_2 \\ \Delta \dot{x}_3 \\ \Delta \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \Delta \dot{x} \\ \Delta \ddot{x} \\ \Delta \dot{\theta} \\ \Delta \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -0.3267 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -13.5022 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \dot{x} \\ \Delta \theta \\ \Delta \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.333 \\ 0 \\ 0.444 \end{bmatrix} [\Delta u] \quad (7.163)$$

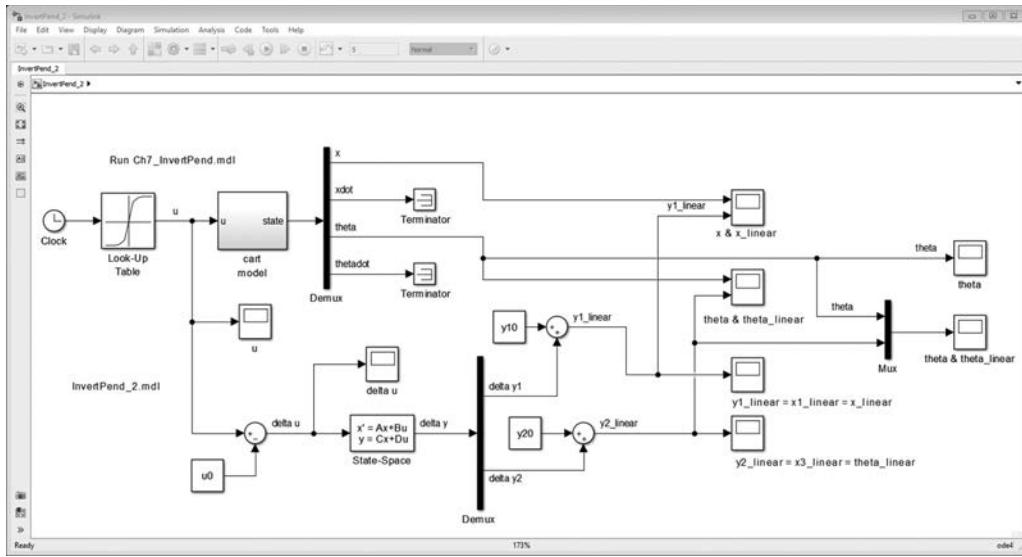


FIGURE 7.62 Simulink diagram for comparing nonlinear and linearized models.

On the other hand, when the state is $[\Delta x \Delta\theta \Delta\dot{x} \Delta\ddot{\theta}]^T$, Equation 7.162 implies

$$\begin{bmatrix} \Delta\dot{x}_1 \\ \Delta\dot{x}_2 \\ \Delta\dot{x}_3 \\ \Delta\dot{x}_4 \end{bmatrix} = \begin{bmatrix} \Delta\dot{x} \\ \Delta\dot{\theta} \\ \Delta\ddot{x} \\ \Delta\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -0.3267 & 0 & 1 \\ 0 & -13.5022 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\theta \\ \Delta\dot{x} \\ \Delta\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.333 \\ 0.444 \end{bmatrix} [\Delta u] \quad (7.164)$$

Once the linearized system matrices A_1 , B_1 , C_1 , and D_1 or A_2 , B_2 , C_2 , and D_2 are known, the inverted pendulum dynamics can be approximated using either set, and the response should compare favorably with the nonlinear system response provided the state and input deviations from the operating point are kept small. Figure 7.62 is the Simulink diagram for comparing the nonlinear system model and the linearized model using the set of matrices A_2 , B_2 , C_2 , and D_2 obtained from the “linmod” function.

Figure 7.63 shows the nonlinear and linearized response for $\theta(t)$ corresponding to a pulse input force of magnitude 2.5 N from 1 to 2 s. Agreement between the nonlinear and linearized responses is very good. Note the small deviation in $\theta(t)$ from $\theta^0 = \pi$ rad resulting from the particular input.

Figure 7.64 exemplifies what happens when the state vector $x(t)$ deviates by a significant amount from x^0 . The magnitude of the applied force pulse input is increased to 25 N. The nonlinear system model and linearized approximation no longer exhibit the same level of agreement as before.

Due to the absence of damping, the (nonlinear and linear) models predict sustained oscillations. Hence, the coefficient matrices A_1 and A_2 must possess a pair of purely imaginary characteristic roots, easily confirmed by checking the eigenvalues of each. The statements “eig (A1)” and “eig (A2)” both return two real eigenvalues 0,0, and two imaginary eigenvalues $\pm j3.674537$.

A closer look at Equations 7.163 and 7.164 reveals a simpler formulation of the governing equations. From Equation 7.164,

$$\Delta\ddot{x} = -0.3267 \Delta\theta + 0.333\Delta u \quad (7.165)$$

$$\Delta\ddot{\theta} + 13.5022 \Delta\theta = 0.444\Delta u \quad (7.166)$$

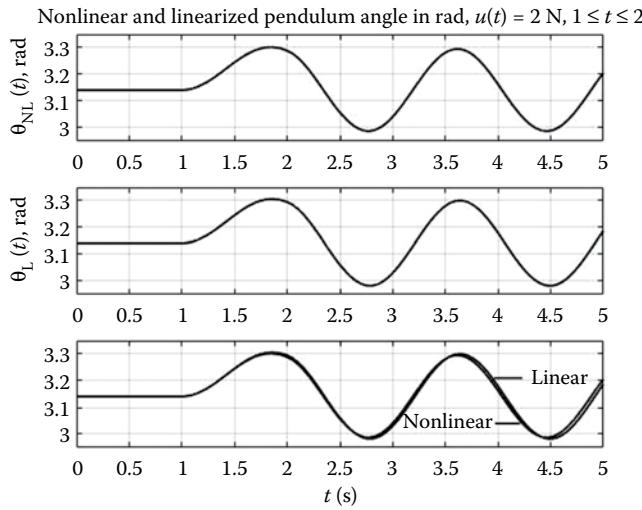


FIGURE 7.63 Nonlinear and linearized model response $\theta(t)$ for $u(t) = 2.5 \text{ N}$, $1 \leq t \leq 2$.

The natural modes from Equation 7.166 are $s_1, s_2 = (-13.5022)^{1/2} = \pm j3.6745$, and the frequency of undamped oscillations is $\omega_n = 3.6745 \text{ rad/s}$ (see Figures 7.63 and 7.64). Furthermore, the two remaining characteristic roots are, from Equation 7.165, both zero. Laplace transforming Equations 7.165 and 7.166 and solving for $\Delta\theta(s)$ and $\Delta X(s)$ result in

$$\Delta\theta(s) = \frac{0.444}{s^2 + 13.5022} \Delta U(s) \quad (7.167)$$

$$\Delta X(s) = 0.333 \left[\frac{s^2 + 13.0666}{s^2(s^2 + 13.5022)} \right] \Delta U(s) \quad (7.168)$$

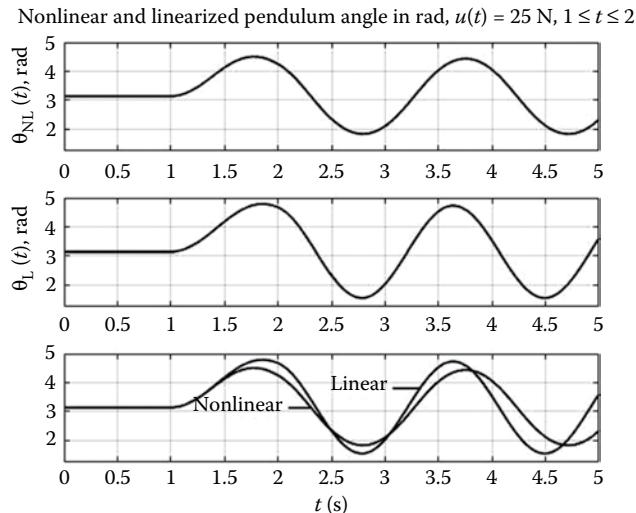


FIGURE 7.64 Nonlinear and linearized model response $\theta(t)$ for $u(t) = 25 \text{ N}$, $1 \leq t \leq 2$.

The inverted pendulum is often used as an example of an inherently (open-loop) unstable system, and numerous linear controls texts demonstrate techniques for designing linear controllers to balance the pendulum in the upright position ($\theta = 0$). The steady-state operating point $[x^0, \dot{x}^0, \theta^0, \dot{\theta}^0; u^0] = (0, 0, 0, 0; 0)$ is unstable, easily verified by changing “ $x30 = \theta^0$ ” to zero in M-file “*Ch7_InvertPend.m*” and observing the eigenvalues of the linearized system matrix A_1 or A_2 . Of course, basic intuition suggests as much, that is, “What happens to the pendulum when it is displaced from the upright equilibrium position?” Exercise 7.24 looks at this case in more detail.

You can implement your own “*linmod*” function to numerically compute the linearized system matrices A , B , C , and D . To illustrate, suppose we wish to estimate a_{43} in Equation 7.163. The exact value of -13.5022 was computed from the analytical expression for the partial derivative $(\partial f_4 / \partial x_3)(x^0, \underline{u}^0)$ using Equations 7.153 and 7.154. A simple central difference formula to approximate $(\partial f_4 / \partial x_3)(x^0, \underline{u}^0)$ is

$$\frac{\partial f_4}{\partial x_3}(x^0, \underline{u}^0) \approx \frac{f_4(x_1^0, x_2^0, x_3^0 + \Delta, x_4^0, u^0) - f_4(x_1^0, x_2^0, x_3^0 - \Delta, x_4^0, u^0)}{2\Delta} \quad (7.169)$$

From Equation 7.144, the numerator terms are

$$f_4(x_1^0, x_2^0, x_3^0 + \Delta, x_4^0, u^0) = \frac{-(ml/2)(x_4^0)^2 \sin 2(x_3^0 + \Delta) + (m+M)g \sin(x_3^0 + \Delta) - u \cos(x_3^0 + \Delta)}{l[M + m \sin^2(x_3^0 + \Delta)]} \quad (7.170)$$

$$f_4(x_1^0, x_2^0, x_3^0 - \Delta, x_4^0, u^0) = \frac{-(ml/2)(x_4^0)^2 \sin 2(x_3^0 - \Delta) + (m+M)g \sin(x_3^0 - \Delta) - u \cos(x_3^0 - \Delta)}{l[M + m \sin^2(x_3^0 - \Delta)]} \quad (7.171)$$

The operating point is $(x^0; u^0) = [0 \ 0 \ \pi \ 0; 0]^T$. After substituting in the numerical values for m , M , g and choosing $\Delta = 0.01$, we have

$$\begin{aligned} \frac{\partial f_4}{\partial x_3}(x^0, \underline{u}^0) &\approx \frac{(0.1+3)(9.8)}{(0.75)(2)(0.01)} \left[\frac{\sin(\pi + 0.01)}{3 + 0.1 \sin^2(\pi + 0.01)} - \frac{\sin(\pi - 0.01)}{3 + 0.1 \sin^2(\pi - 0.01)} \right] \\ &\approx -13.5020 \end{aligned} \quad (7.172)$$

which is very close to the analytically obtained value -13.5022 . Another example of linearization involving nonlinear tanks is presented in Section 8.4.

7.4.4 MULTIPLE LINEARIZED MODELS FOR A SINGLE SYSTEM

When the inputs to a nonlinear system vary by a considerable amount, a single linearized model may no longer be sufficient to describe the excursions of the state vector about an individual operating point. It becomes necessary to linearize the system dynamics in terms of deviation variables about different operating points. The linearized models are applicable to specific regions in state space. While the initial state may have been at equilibrium, be mindful that the initial conditions of the deviation variables are no longer zero as the state transitions between different linearized regions in state space. The situation is illustrated in Figure 7.65 for an autonomous, second-order system with different linearized models in each of the four quadrants of state space.

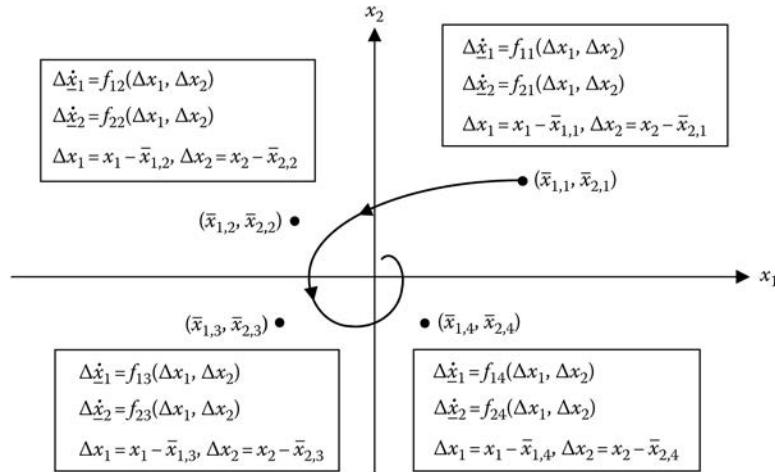


FIGURE 7.65 State trajectory of an autonomous, nonlinear, second-order system linearized about four different operating points.

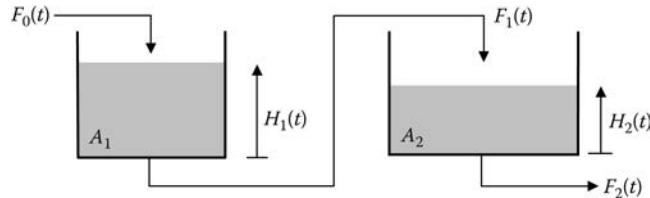


FIGURE 7.66 Second-order system consisting of two nonlinear first-order tanks.

An example of how to accommodate multiple operating points is illustrated using the nonlinear second-order system in [Figure 7.66](#). The mathematical model describing the coupled dynamics of the two tanks is given in Equations 7.173 through 7.176.

$$A_1 \frac{dH_1}{dt} + F_1 = F_0 \quad (7.173)$$

$$F_1 = c_1 H_1^{1/2} \quad (7.174)$$

$$A_2 \frac{dH_2}{dt} + F_2 = F_1 \quad (7.175)$$

$$F_2 = c_2 H_2^{1/2} \quad (7.176)$$

Solving for the state derivative functions gives

$$\frac{dH_1}{dt} = f_1(H_1, H_2, F_0) = \frac{1}{A_1}(F_0 - c_1 H_1^{1/2}) \quad (7.177)$$

$$\frac{dH_2}{dt} = f_2(H_1, H_2, F_0) = \frac{1}{A_2}(c_1 H_1^{1/2} - c_2 H_2^{1/2}) \quad (7.178)$$

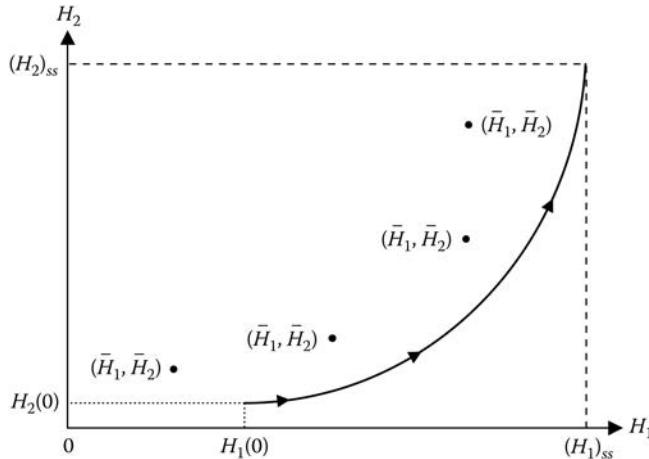


FIGURE 7.67 Several operating points and state trajectory for tanks subject to $F_0(t) = \tilde{F}_0$, $t \geq 0$.

Suppose the flow into the first tank is constant, that is, $F_0(t) = \tilde{F}_0$, $t \geq 0$. Steady-state levels are obtained by setting both derivatives equal to zero. The result is

$$(H_1)_{ss} = \left(\frac{\tilde{F}_0}{c_1} \right)^2, \quad (H_2)_{ss} = \left(\frac{\tilde{F}_0}{c_2} \right)^2 \quad (7.179)$$

A typical state trajectory starting from $\{H_1(0), H_2(0)\}$ and ending at $\{(H_1)_{ss}, (H_2)_{ss}\}$ is shown in Figure 7.67. Four different operating points designated (\bar{H}_1, \bar{H}_2) in the (H_1, H_2) state space are also shown.

Equations 7.177 and 7.178 are initially linearized about the operating point (\bar{H}_1, \bar{H}_2) nearest to the initial state $\{H_1(0), H_2(0)\}$. The equations are relinearized as necessary, that is, when the state trajectory transitions from a neighborhood about one operating point to another region about a different operating point.

For nonlinear systems with inputs, linearization requires a nominal input for each operating point. The following example presents the results of using a single operating point compared with using multiple operating points for linearizing the two-tank system in Figure 7.66 with constant inflow.

EXAMPLE 7.3

For the two-tank system shown in Figure 7.66, baseline parameter values are

$$A_1 = 25 \text{ ft}^2, A_2 = 15 \text{ ft}^2, c_1 = 3 \text{ ft}^3/\text{min}/\text{ft}^{1/2}, c_2 = 4 \text{ ft}^3/\text{min}/\text{ft}^{1/2}$$

- Find the steady-state levels $(H_1)_{ss}$ and $(H_2)_{ss}$ when $\tilde{F}_0 = 12 \text{ ft}^3/\text{min}$.
- Choose a steady-state operating point (H_1, H_2) where $H_1 = 0.5(H_1)_{ss}$. Find the tank 2 level H_2 at the operating point and the nominal inflow \tilde{F}_0 at the operating point.
- Introduce deviation variables and linearize the model differential equations.
- Solve the linearized equations for the case where both tanks are initially empty and $\tilde{F}_0 = 12 \text{ ft}^3/\text{min}$. Compare the level responses of both tanks to the solutions obtained from the nonlinear equations.
- Plot state trajectories for the linearized and nonlinear systems.
- Establish four steady-state operating points corresponding to tank 1 levels of $0.125(H_1)_{ss}$, $0.375(H_1)_{ss}$, $0.625(H_1)_{ss}$, and $0.875(H_1)_{ss}$.
- Repeat parts (d) and (e).

a. From Equation 7.179, the steady-state levels are

$$(H_1)_{ss} = \left(\frac{\tilde{F}_0}{C_1} \right)^2 = \left(\frac{12}{3} \right)^2 = 16 \text{ ft}, \quad (H_2)_{ss} = \left(\frac{\tilde{F}_0}{C_2} \right)^2 = \left(\frac{12}{4} \right)^2 = 9 \text{ ft} \quad (7.180)$$

b. The required inflow \bar{F}_0 to maintain tank 1 level at $\bar{H}_1 = 0.5(H_1)_{ss} = 8 \text{ ft}$ is equal to the steady-state outflow from tank 1, that is,

$$\bar{F}_0 = \bar{F}_1 = c_1 \bar{H}_1^{1/2} = 3(8)^{1/2} = 8.4853 \text{ ft}^3/\text{min} \quad (7.181)$$

For steady-state conditions, tank 2 level must be

$$\bar{H}_2 = \left(\frac{\bar{F}_2}{C_2} \right)^2 = \left(\frac{\bar{F}_1}{C_2} \right)^2 = \left(\frac{3(8)^{1/2}}{4} \right)^2 = 4.5 \text{ ft} \quad (7.182)$$

The steady-state operating point is $(\bar{H}_1, \bar{H}_2) = (8 \text{ ft}, 4.5 \text{ ft})$ and $\bar{F}_0 = 8.4853 \text{ ft}^3/\text{min}$.

c. Introducing deviation variables $\Delta H_1 = H_1 - \bar{H}_1$, $\Delta H_2 = H_2 - \bar{H}_2$ for the tank levels and $\Delta F_0 = F_0 - \bar{F}_0 = \tilde{F}_0 - \bar{F}_0$ for the inflow to tank 1 produces a system of linearized differential equations

$$\Delta \dot{H}_1 = a_{11} \Delta H_1 + a_{12} \Delta H_2 + b_1 \Delta F_0 \quad (7.183)$$

$$\Delta \dot{H}_2 = a_{21} \Delta H_1 + a_{22} \Delta H_2 + b_2 \Delta F_0 \quad (7.184)$$

$$a_{11} = \frac{\partial f_1(H_1, H_2, F_0)}{\partial H_1} \Bigg|_{H_1=\bar{H}_1, H_2=\bar{H}_2, F_0=\bar{F}_0} = \frac{-c_1}{2A_1 \bar{H}_1^{1/2}} \quad (7.185)$$

$$a_{12} = \frac{\partial f_1(H_1, H_2, F_0)}{\partial H_2} \Bigg|_{H_1=\bar{H}_1, H_2=\bar{H}_2, F_0=\bar{F}_0} = 0 \quad (7.186)$$

$$a_{21} = \frac{\partial f_2(H_1, H_2, F_0)}{\partial H_1} \Bigg|_{H_1=\bar{H}_1, H_2=\bar{H}_2, F_0=\bar{F}_0} = \frac{c_1}{2A_2 \bar{H}_1^{1/2}} \quad (7.187)$$

$$a_{22} = \frac{\partial f_2(H_1, H_2, F_0)}{\partial H_2} \Bigg|_{H_1=\bar{H}_1, H_2=\bar{H}_2, F_0=\bar{F}_0} = \frac{-c_2}{2A_2 \bar{H}_2^{1/2}} \quad (7.188)$$

$$b_1 = \frac{\partial f_2(H_1, H_2, F_0)}{\partial F_0} \Bigg|_{H_1=\bar{H}_1, H_2=\bar{H}_2, F_0=\bar{F}_0} = \frac{1}{A_1} \quad (7.189)$$

$$b_2 = \frac{\partial f_2(H_1, H_2, F_0)}{\partial F_0} \Bigg|_{H_1=\bar{H}_1, H_2=\bar{H}_2, F_0=\bar{F}_0} = 0 \quad (7.190)$$

Substituting values for A_1 , A_2 , c_1 , c_2 , \bar{H}_1 , \bar{H}_2 in Equation 7.185 and Equations 7.187 through 7.189, the linearized tank model is

$$\Delta \dot{H}_1 = -0.212 \Delta H_1 + 0.04 \Delta H_2 \quad (7.191)$$

$$\Delta \dot{H}_2 = -0.0354 \Delta H_1 - 0.0629 \Delta H_2 \quad (7.192)$$

- d. The general solution to Equations 7.183 and 7.184 with $a_{12} = b_2 = 0$, initial conditions $\Delta H_1(0) = H_1(0) - \bar{H}_1$, $\Delta H_2(0) = H_2(0) - \bar{H}_2$, and $\Delta F_0 = \tilde{F}_0 - \bar{F}_0$ is (see Exercise 7.25)

$$\Delta H_1(t) = -\frac{b_1 \Delta F_0}{a_{11}} + \left[\Delta H_1(0) + \frac{b_1 \Delta F_0}{a_{11}} \right] e^{a_{11}t} \quad (7.193)$$

$$\begin{aligned} \Delta H_2(t) = & \frac{a_{21}b_1 \Delta F_0}{a_{11}a_{22}} + \frac{a_{21}}{(a_{11} - a_{22})} \left[\Delta H_1(0) + \frac{b_1 \Delta F_0}{a_{11}} \right] e^{a_{11}t} \\ & + \left[\Delta H_2(0) + \frac{a_{21}}{(a_{22} - a_{11})} \left\{ \Delta H_1(0) + \frac{b_1 \Delta F_0}{a_{11}} \right\} \right] e^{a_{22}t} \end{aligned} \quad (7.194)$$

With $H_1(0) = H_2(0) = 0$, $\Delta F_0 = \tilde{F}_0 - \bar{F}_0 = 12 - 8.4853 = 3.5147 \text{ ft}^3/\text{min}$, the linearized tank-level responses $H_1(t) = \bar{H}_1 + \Delta H_1(t)$ and $H_2(t) = \bar{H}_2 + \Delta H_2(t)$ become

$$H_1(t) = 14.6274 \left(1 - e^{-0.0212t} \right), t \geq 0 \quad (7.195)$$

$$H_2(t) = 8.2279 - 12.4195e^{-0.212t} + 4.1916e^{-0.0629t}, t \geq 0 \quad (7.196)$$

The nonlinear system responses can be approximated by resorting to simulation with a fixed-step numerical integrator and small integration step. The Simulink diagram is shown in Figure 7.68.

An RK-4 integrator with step size of 0.01 s was used to approximate the tank 1 and tank 2 nonlinear system level responses. The linearized system responses in Equations 7.195 and 7.196 are plotted along with the nonlinear system responses in Figure 7.69.

Note that the nonlinear system step responses approach the correct steady-state levels $(H_1)_{ss} = 16 \text{ ft}$ and $(H_2)_{ss} = 9 \text{ ft}$ predicted in Equation 7.180. Can you verify whether the tank levels for the linearized system shown in Figure 7.69, namely, $H_1(350) = 14.62 \text{ ft}$ and $H_2(350) = 8.22 \text{ ft}$, are correct?

- e. The state trajectories are shown in Figure 7.70.
f. A similar procedure to the one used in parts (a) through (e) establishes four distinct steady-state operating points (\bar{H}_1, \bar{H}_2) where \bar{H}_1 is one of the four values $0.125(H_1)_{ss} = 2 \text{ ft}$, $0.375(H_1)_{ss} = 6 \text{ ft}$, $0.675(H_1)_{ss} = 10 \text{ ft}$, $0.875(H_1)_{ss} = 14 \text{ ft}$. The corresponding values of \bar{H}_2 and \bar{F}_0 are shown in Table 7.7.

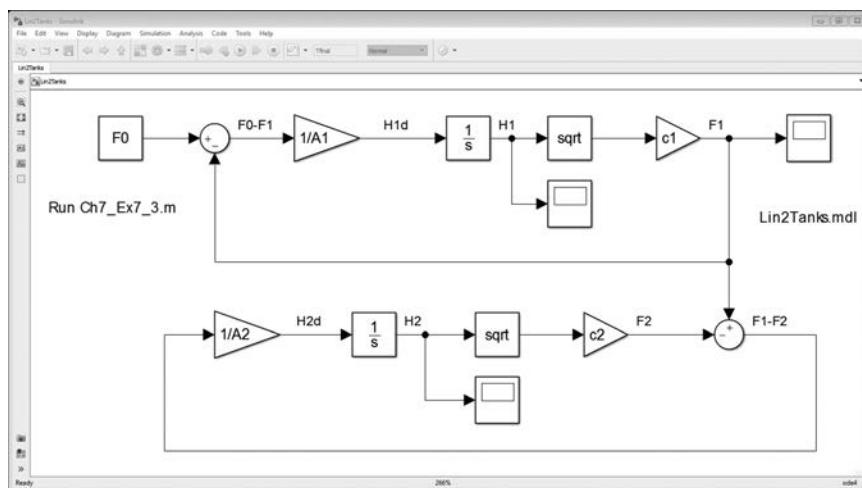


FIGURE 7.68 Simulink diagram for nonlinear two-tank system.

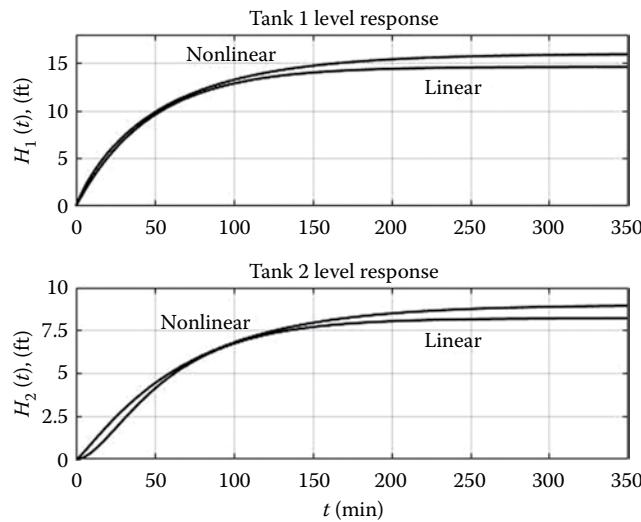


FIGURE 7.69 Comparison of linearized and nonlinear system tank-level responses.

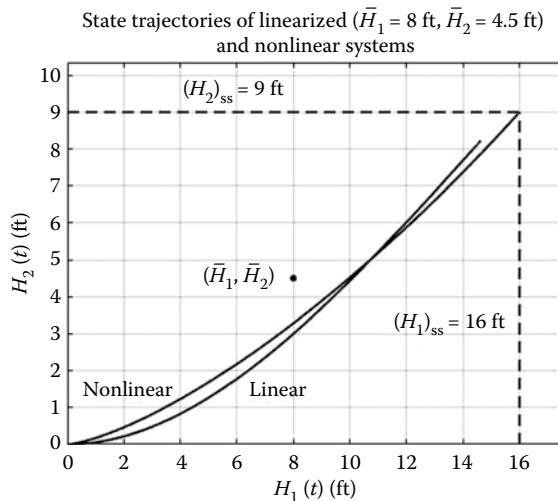


FIGURE 7.70 State trajectories of linearized ($\bar{H}_1 = 8$ ft, $\bar{H}_2 = 4.5$ ft) and nonlinear systems.

TABLE 7.7
Steady-State Operating Points (\bar{H}_1, \bar{H}_2) and Corresponding \bar{F}_0

Region	\bar{H}_1 (ft)	\bar{H}_2 (ft)	\bar{F}_0 (ft ³ /min)
I	2	1.125	4.2426
II	6	3.375	7.3485
III	10	5.625	9.4868
IV	14	7.875	11.2250

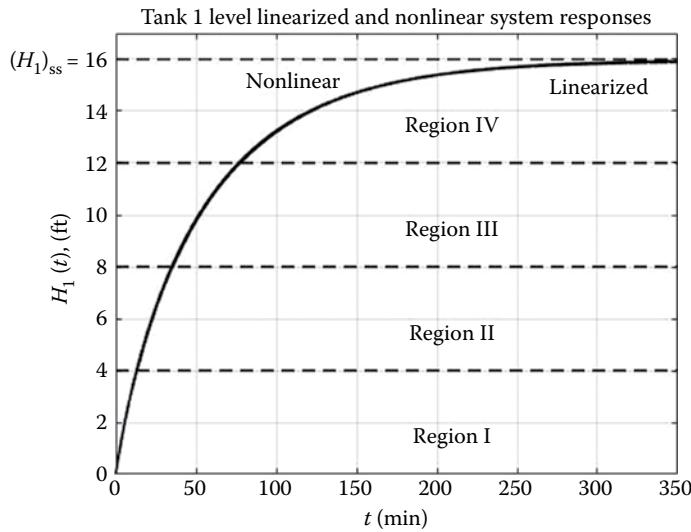


FIGURE 7.71 Tank 1 linearized system response using four operating points and nonlinear system response.

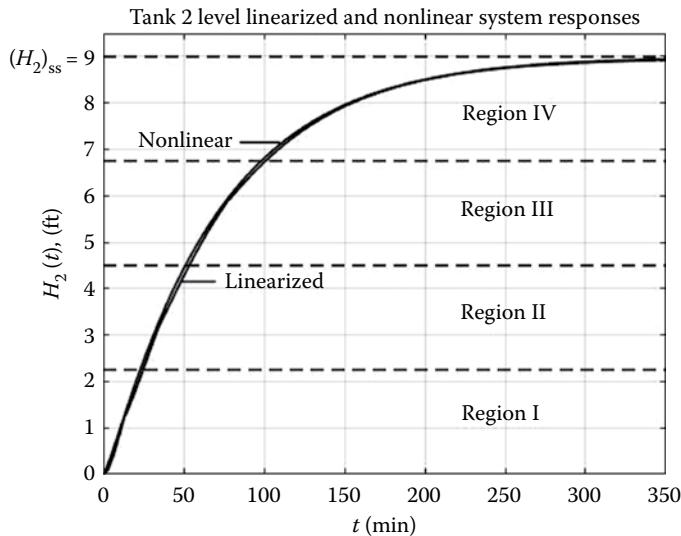


FIGURE 7.72 Tank 2 linearized system response using four operating points and nonlinear system response.

- g. The nonlinear and linearized system responses for both tanks are shown in Figures 7.71 and 7.72.

The static nonlinear operating curves for each tank are shown in Figure 7.73, and the state trajectories of the linearized and nonlinear systems are shown in Figure 7.74. The operating points listed in Table 7.7 are shown as well. Note the improved accuracy in the step response of the system linearized about multiple operating points compared to the case illustrated in Figure 7.70 where a single operating point (\bar{H}_1, \bar{H}_2) was used.

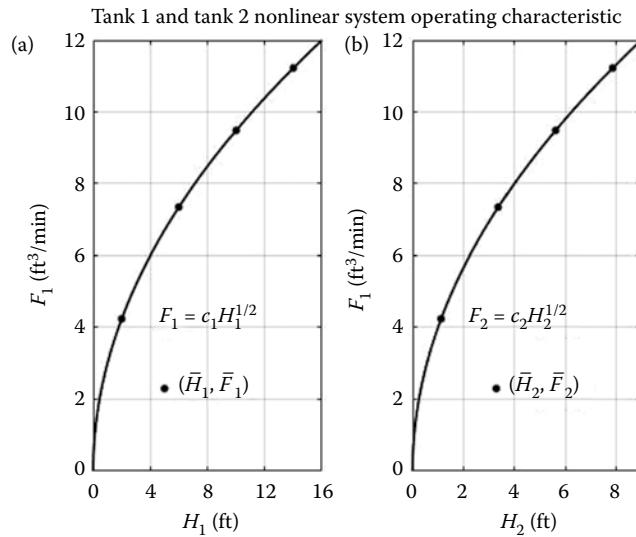


FIGURE 7.73 Static nonlinear operating curves for both tanks. (a) Tank 1 nonlinear system operating characteristic. (b) Tank 2 nonlinear system operating characteristic.

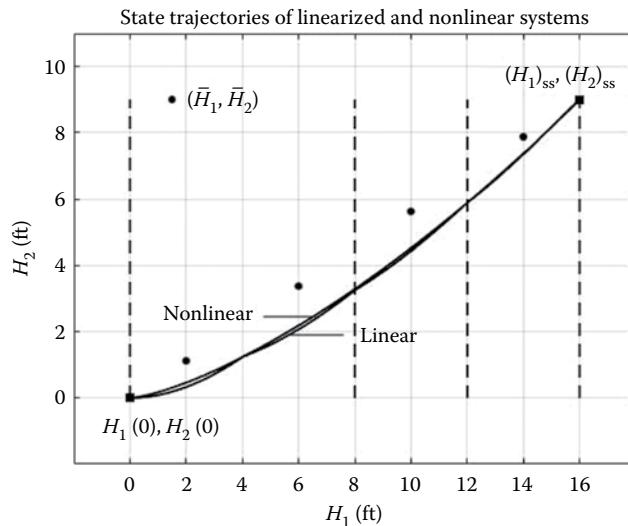


FIGURE 7.74 State trajectories for linearized and nonlinear systems.

EXERCISES

- 7.18 The nonlinear tank model in which the outflow is based on Equation 7.89 can be thought of as exhibiting a variable fluid resistance, that is, $R = f(H)$. When the tank is linearized about an operating pt (\bar{H}, \bar{F}_0) , the resistance $\bar{R} = f(\bar{H}) = \Delta H / \Delta F_0$, which is the reciprocal of the slope of the tangent drawn to the function $F_0 = cH^{1/2}$ at the operating point (see Figure 7.51). Hence, for small variations about the operating point, the tank behaves similar to a linear tank with resistance \bar{R} .

- a. Show that the linearized resistance \bar{R} about the point (\bar{H}, \bar{F}_0) is equal to $2\bar{H}_0/\bar{F}_0$.
- b. For the tank whose operating curve is shown in Figure 7.51, find the linearized resistance \bar{R} when the tank level fluctuates by a small amount about
(i) 5 ft (ii) 10 ft (iii) 15 ft (iv) 20 ft.
- c. Comment on the apparent fluid resistance of a nonlinear tank as the level rises.
- 7.19 A tank with nonlinear operating curve, shown in Figure 7.51, and cross-sectional area 25 ft² is initially filled to a height of 9 ft. There is no inflow.
- Employ Simulink (with suitable integrator and step size) to simulate the emptying of the tank. Graph $H_{\text{sim}}(t)$.
 - Linearize the differential equation model about the initial point $H(0) = 9$ ft, $F_0(0) = 1.5$ ft³/min, that is, choose the operating pt as $(\bar{H}, \bar{F}_0) = (9, 1.5)$, and find the linear differential equation describing the deviation $\Delta H(t) = H(t) - \bar{H}$.
 - Find the analytical solution for $\Delta H(t)$ and plot $H_{\text{lin}}(t) = \bar{H} + \Delta H(t)$ on the same graph as the simulated solution $H_{\text{sim}}(t)$. Comment on the results.
 - Find the analytical solution for the level, $H_{\text{anal}}(t)$, and compare it with the simulated response $H_{\text{sim}}(t)$ in part (a) and linearized response $H_{\text{lin}}(t)$ in part (c).
- 7.20 Starting with Equations 7.101 and 7.110, obtain Equation 7.111 for the linearized state derivatives.
- 7.21 The nonlinear tank shown in Figure E7.21a has an adjustable valve in the discharge line. The valve opening is given by the normalized variable θ ($0 \leq \theta \leq 1$) where $\theta = 0$ is a closed valve and $\theta = 1$ represents a fully open valve. The outflow is obtained from $F_0 = F_0(\theta, H) = c(\theta)H^{1/2}$.

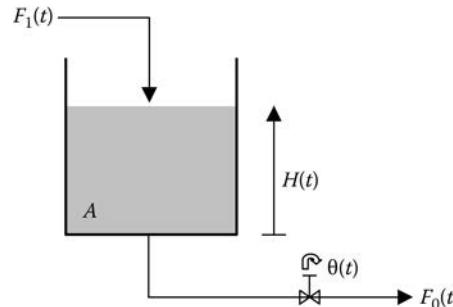
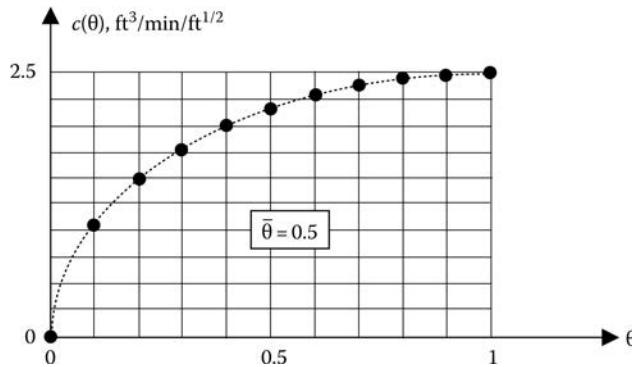


FIGURE E7.21A

An expression for $\Delta F_0(t)$ in the linearized differential equation model of the tank $A(d/dt)\Delta H(t) + \Delta F_0(t) = \Delta F_1(t)$ is obtained as follows:

$$\begin{aligned}
F_0 &= F_0(\bar{\theta}, \bar{H}_0) + \frac{\partial}{\partial \theta} F_0(\bar{\theta}, \bar{H}_0) \Delta \theta + \frac{\partial}{\partial H} F_0(\bar{\theta}, \bar{H}_0) \Delta H \\
&\Rightarrow \Delta F_0 = \frac{\partial}{\partial \theta} [c(\theta), H^{1/2}]_{\theta=\bar{\theta}, H=\bar{H}} \Delta \theta + \frac{\partial}{\partial H} [c(\theta), H^{1/2}]_{\theta=\bar{\theta}, H=\bar{H}} \Delta H \\
&= \bar{H}^{1/2} \left. \frac{d}{d\theta} c(\theta) \right|_{\theta=\bar{\theta}} \Delta \theta + c(\bar{\theta}) \left. \frac{d}{dH} H^{1/2} \right|_{H=\bar{H}} \Delta H \\
&= \bar{H}^{1/2} c'(\bar{\theta}) \Delta \theta + c(\bar{\theta}) \left[\frac{1}{2} \bar{H}^{-1/2} \right] \Delta H
\end{aligned}$$

Data points along the valve-operating characteristic $c(\theta)$ are shown in Figure E7.21b:

**FIGURE E7.21B**

- a. Find the linearized differential equation about the steady-state operating point where $\theta = 0.5$, $\bar{H} = 9 \text{ ft}$.
- b. Simulate the tank-level response when the inflow increases by 10% and valve opening decreases by 15% with respect to their operating point values. The initial conditions are $H(0) = H$, $\theta(0) = \theta$. Assume both changes are step inputs. The cross-sectional area of the tank is 50 ft^2 .
- 7.22 In Example 7.2, let $\beta = 0$ and vary α from -0.5 to 0.5 in steps of 0.05 .
- Plot the linearized level responses on the same graph.
 - Plot the nonlinear level responses on the same graph.
 - Repeat parts (a) and (b) for $\bar{H} = 4, 16$ and 25 ft .
- 7.23 The populations of two species coexisting in the same environment are governed by the predator-prey equations

$$\frac{dx}{dt} = x(a - bx - cy) + u_x$$

$$\frac{dy}{dt} = y(-k + \lambda x) + u_y$$

where $x = x(t)$ is the population of the prey at time “ t ,” $y = y(t)$ is the population of predators at time “ t ,” $u_x = u_x(t)$ is the net rate of new prey introduced at time “ t ,” $u_y = u_y(t)$ is the net rate of new predators entering the environment at time “ t ,” and a , b , c , λ , and k are parameters of the system.

- Find the nontrivial equilibrium points (\bar{x}, \bar{y}) in the x, y plane when the two inputs are $u_x = \bar{u}_x = 0$, $u_y = \bar{u}_y = 0$, $t \geq 0$. Leave your answers for \bar{x} and \bar{y} in terms of the system parameters.
 - Introduce deviation variables Δx , Δy , Δu_x , and Δu_y and choose the outputs as $\Delta y_1 = \Delta x$ and $\Delta y_2 = \Delta y$. Linearize the state equations about the operating point where $x = \bar{x}$, $y = \bar{y}$, $u_x = \bar{u}_x$, and $u_y = \bar{u}_y$ and find the linearized system matrices A , B , C , and D in terms of the system parameters.
 - Find the transfer functions $\Delta x(s)/\Delta u_x(s)$, $\Delta x(s)/\Delta u_y(s)$, $\Delta y(s)/\Delta u_x(s)$, and $\Delta y(s)/\Delta u_y(s)$.
 - Suppose the numerical values of the system parameters are $a = 12$, $b = 0$, $c = 2$, $k = 20$, and $\lambda = 4$. Further, let the inputs be $u_x = 1$, $t \geq 0$ and $u_y = 0$, $t \geq 0$. Simulate the nonlinear and linearized system responses starting from the point $x(0) = 0$, $y(0) = 0$ and compare results using plots of
 - $x(t)$ vs. t and $x_{lin}(t)$ vs. t on the same graph
 - $y(t)$ vs. t and $y_{lin}(t)$ vs. t on the same graph
 - $y(t)$ vs. $x(t)$ and $y_{lin}(t)$ vs. $x_{lin}(t)$ on the same graph
- Comment on the results.

- e. Repeat part (d) with $u_x = 0$, $t \geq 0$, and $u_y = -1$, $t \geq 0$.
f. Repeat parts (d) and (e) with $x(0) = 1$ and $y(0) = 1$.
- 7.24 The dynamics of an inverted pendulum with physical parameters $M = 4$ kg, $m = 0.15$ kg, and $l = 0.8$ m is to be linearized about a steady-state operating point $(\underline{x}^0; \underline{u}^0) = [x^0, \dot{x}^0, \theta^0, \dot{\theta}^0; u^0] = [0 \ 0 \ 0 \ 0; 0]^T$.
- Find the linearized system matrices A , B , C , and D
 - Analytically
 - Using “linmod”
 - By numerical approximation using a central difference approximation formula with suitably small Δ
 - Find the eigenvalues of the coefficient matrix A for the three methods in part (a). Comment on the results.
 - Use the A , B , C , and D matrices resulting from the analytical approach and simulate $\theta(t)$, $t \geq 0$ in response to the pulse input $u(t) = 0.01 N$, $1 \leq t \leq 2$.
 - Simulate the nonlinear system response for $\theta(t)$, $t \geq 0$ due to the same input in part (c). Compare the linearized and nonlinear responses.
- 7.25 For the two-tank system in [Figure 7.66](#),
- Show that the solution of the linearized differential equations in Equations 7.183 and 7.184 is given in Equations 7.193 and 7.194.
 - Check the solution at $t = 0$ and $t = \infty$.
 - The system in Example 7.3 is linearized about a steady-state operating point where $\bar{H}_1 = 1$ ft. Find expressions for \bar{H}_2 and \bar{F}_0 in terms of \bar{H}_1 and the system parameters c_1 , c_2 , A_1 , and A_2 and then evaluate them numerically.
 - Plot the linearized system response for constant inputs of $\tilde{F}_0 = 2, 4, 8$ ft³/min and both tanks initially empty.
 - Simulate the nonlinear system dynamics for the same constant input values, and plot the responses on the same graph used for the linearized system responses. Comment on the results.
- 7.26 Repeat Example 7.3 for the case where the tanks interact, that is, the flow out of the first tank F_1 enters tank 2 at the bottom and is modeled by

$$F_1 = c_{12}(H_1 - H_2)^{1/2}$$

where $c_{12} = 2$ ft³/min/ft^{1/2}.

- 7.27 The nonlinear pendulum in [Figure E7.27](#) is modeled by $J\ddot{\theta} + c\dot{\theta} + mgr \sin \theta = 0$.

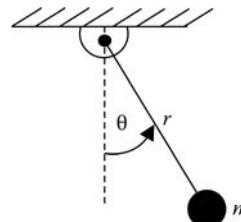


FIGURE E7.27

- The state components are $x_1 = \theta$ and $x_2 = \dot{\theta}$. Find the state derivative functions $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ in the state equations $\dot{x}_1 = f_1(x_1, x_2)$ and $\dot{x}_2 = f_2(x_1, x_2)$.
- Linearize the state equations about the equilibrium point $x_1 = 0$ rad, $x_2 = 0$ rad/s.

- c. The initial conditions are $x_1(0) = 0$ rad, $x_2(0) = 0.1$ rad/s. Find expressions for the linearized responses $x_1(t)$, $x_2(t)$ when the system parameters are

$$m = 0.25 \text{ slugs}, r = 2\text{ft}, c = 0.1 \text{ ft lb/rad/s}, J = mr^2 = 1 \text{ ft 1bs}^2$$

- d. Obtain the nonlinear system response by simulation, and plot the linearized and nonlinear system responses on the same graph.
e. Repeat parts (c) and (d) when the initial conditions are $x_1(0) = 0.25$ rad, $x_2(0) = 0$ rad/s.

7.5 ADDING BLOCKS TO THE SIMULINK LIBRARY BROWSER

7.5.1 INTRODUCTION

In order to keep development costs down, previously verified and validated models are reused in the development of new simulations. A verified model means the model was built right, whereas a validated model means the right model was built. As an example, in Section 5.12, various Kalman filters (continuous, discrete, and steady-state continuous) were developed in Simulink. These models were verified and validated by comparing them to known results (outputs and plots) from MATLAB scripts. It would be beneficial if these models were made available for use by other members of the simulation development team. What follows is the process by which models are added to a library and made available for modeling through the Simulink Library Browser.

Recall from Section 5.12 the case study of Kalman filtering led to the development of three different models: the continuous-time Kalman filter (CTKF), the discrete-time Kalman filter (DTKF), and the steady-state continuous-time Kalman filter (SSCKTF) whose top-level blocks are repeated in [Figures 7.75–7.77](#) for convenience.

In each of [Figures 7.75–7.77](#), the Kalman filter algorithms labeled CTKF Estimates, DTKF Estimates, and SSCKTF Estimates, respectively, have been selected to identify which particular blocks will be added to the library. In order to make these filters available to developers as individual drag and drop blocks within the Simulink Library browser, follow the procedure outlined next.

Start Simulink and click on Create Library as shown in [Figure 7.78](#).

This action opens an untitled library window shown in [Figure 7.79](#).

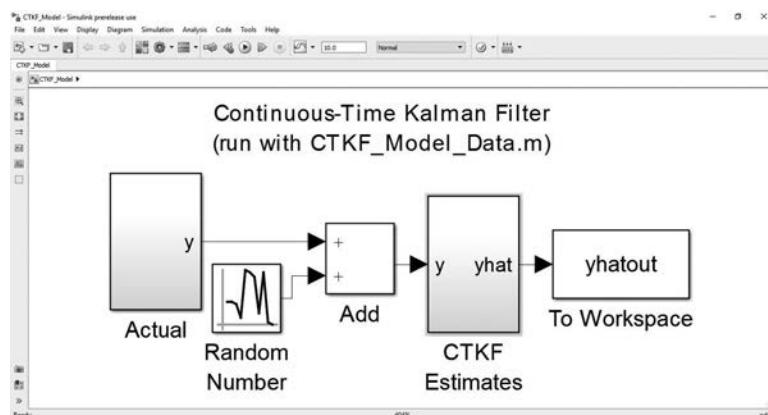


FIGURE 7.75 Continuous-time Kalman filter.

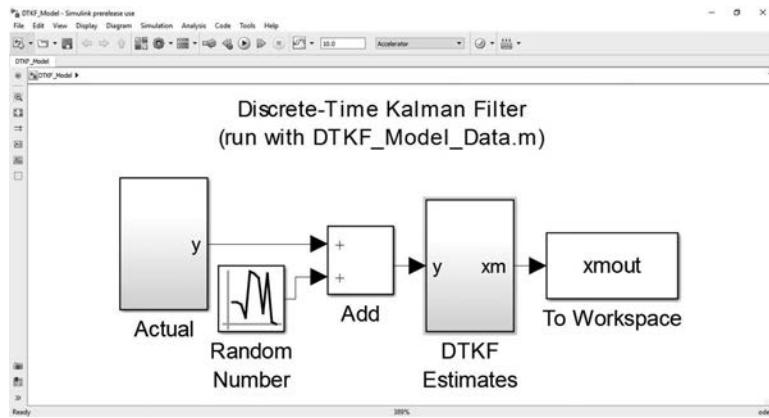


FIGURE 7.76 Discrete-time Kalman filter.

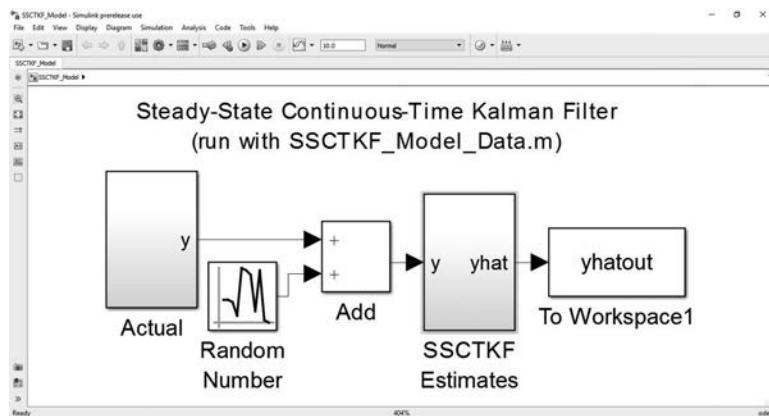


FIGURE 7.77 Steady-state continuous-time Kalman filter.

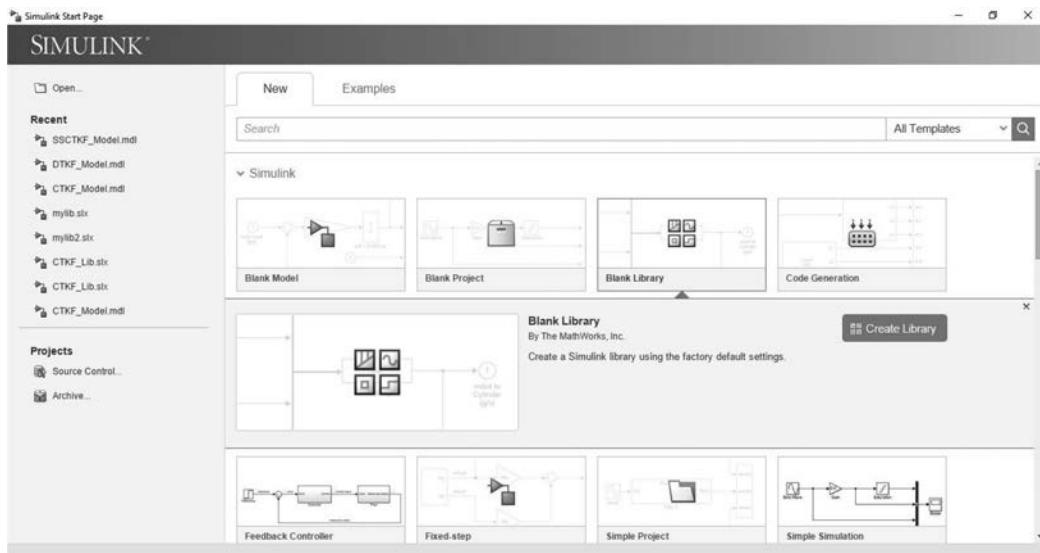


FIGURE 7.78 Creating a library.

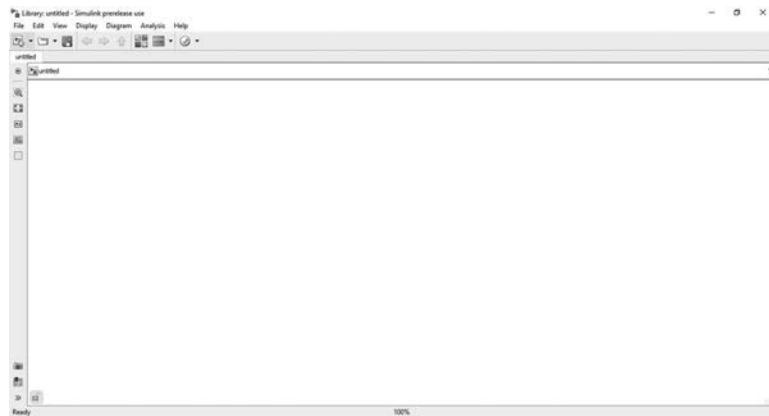


FIGURE 7.79 Library: untitled.

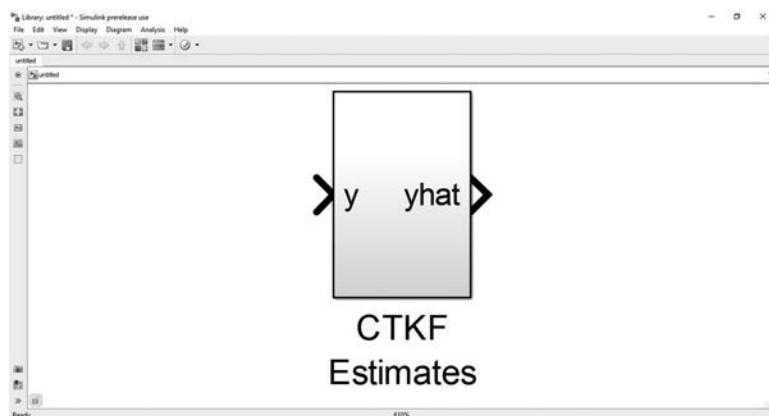


FIGURE 7.80 Drag and drop of CTKF estimates.

Simply drag and drop the CTKF estimates block into the untitled library window. The result of this action is shown in [Figure 7.80](#).

Repeating this procedure for the DTKF Estimates block and the SSCTKF block results in [Figures 7.81](#) and [7.82](#).

From the MATLAB command window, type `set_param(gcs, 'enableLBRepository', 'on')`, then save the blocks into the library as shown in [Figure 7.83](#).

In the Save As dialog box, enter the name of the library, chosen here as “kflib” (to represent Kalman filter library) in [Figure 7.84](#).

Once the library is saved, the name will change from “Library: untitled*” to “Library: kflib” as shown in [Figure 7.85](#).

The next step in the process is the creation of the S-block M-file to load the library when Simulink is started. To view a template, type “edit slblocks” in the MATLAB command window. The M-file template is given as follows where executable lines are identified by bold text.

For the Kalman filter library, the simplified S-block M-file was edited as shown in [Figure 7.86](#). This file must be saved as “slblocks.m” in the same folder as the library file in order for Simulink to acknowledge existence of the “kflib” library at startup.

When Simulink is started, [Figure 7.87](#) shows the Simulink Library Browser with an exploding directory named “My Kalman Filters” containing the three Kalman filters “CTKF Estimates,”

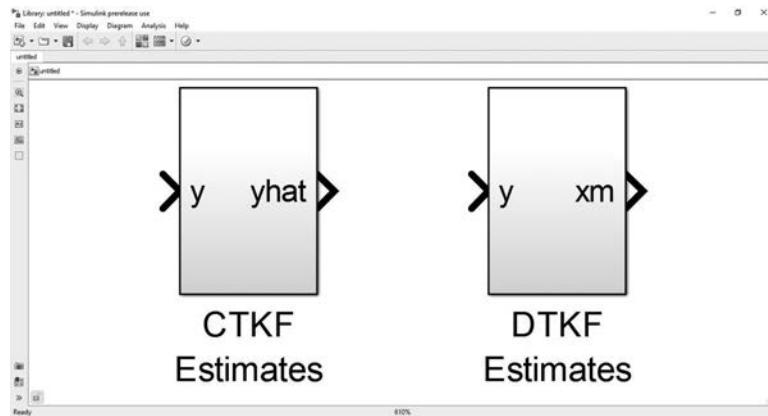


FIGURE 7.81 Drag and drop of DTKF estimates.

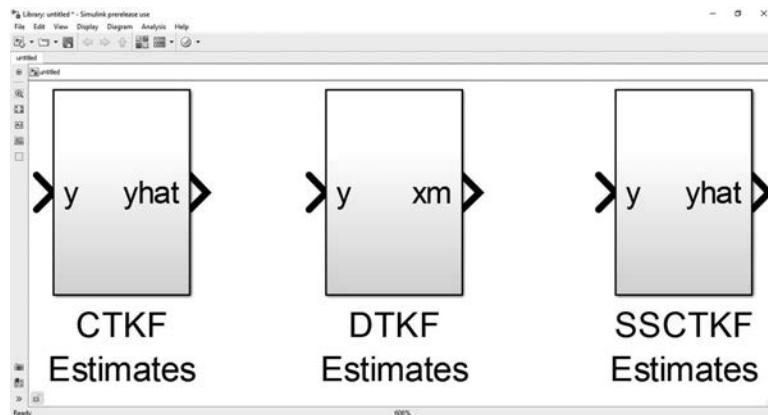


FIGURE 7.82 Drag and drop of SSCTKF estimates.

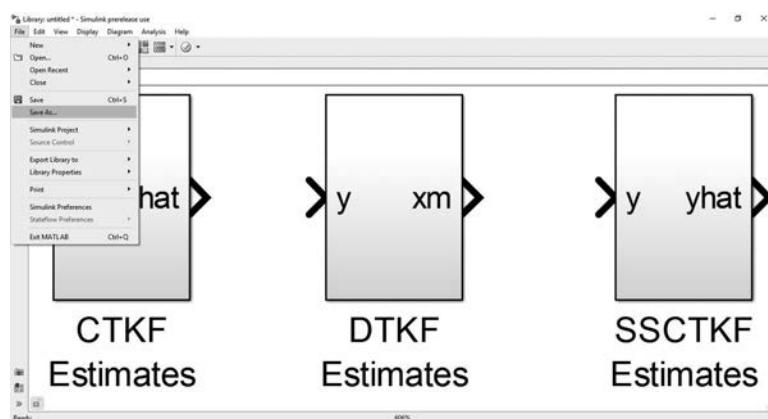


FIGURE 7.83 Saving the blocks into the library.

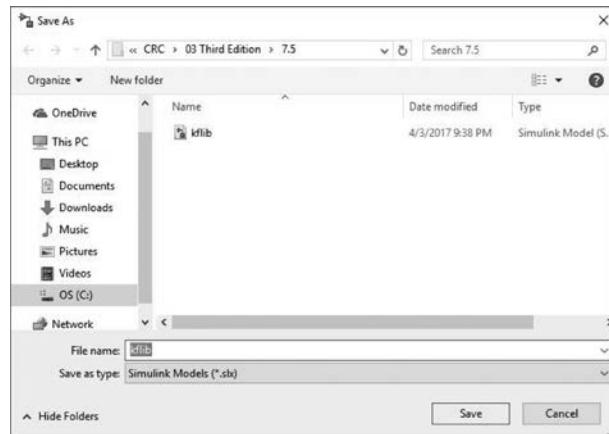


FIGURE 7.84 Saving the Kalman filter library, kflib.

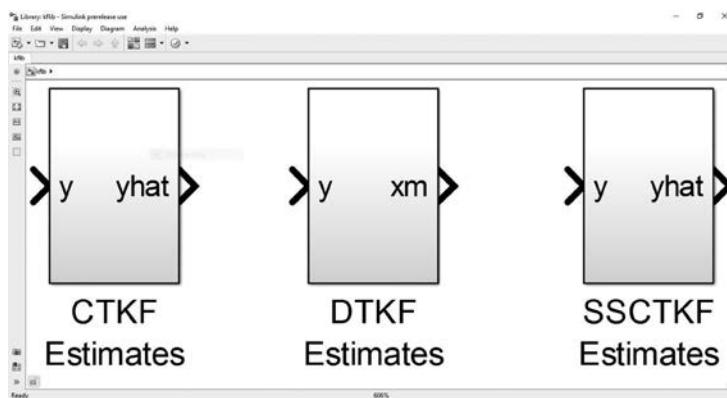


FIGURE 7.85 Library: kflib.

```

function blkStruct = slblocks
    % This function specifies that the library should appear in the
    % Library Browser and be cached in the browser repository

    Browser.Library = 'kflib';
    % 'kflib' is the name of the library
    %

    Browser.Name = 'My Kalman Filter Library';
    % 'My Kalman Filter Library' is the library name that appears
    % in the Library Browser

    blkStruct.Browser = Browser;

```

FIGURE 7.86 S-block M-file "slblocks.m."

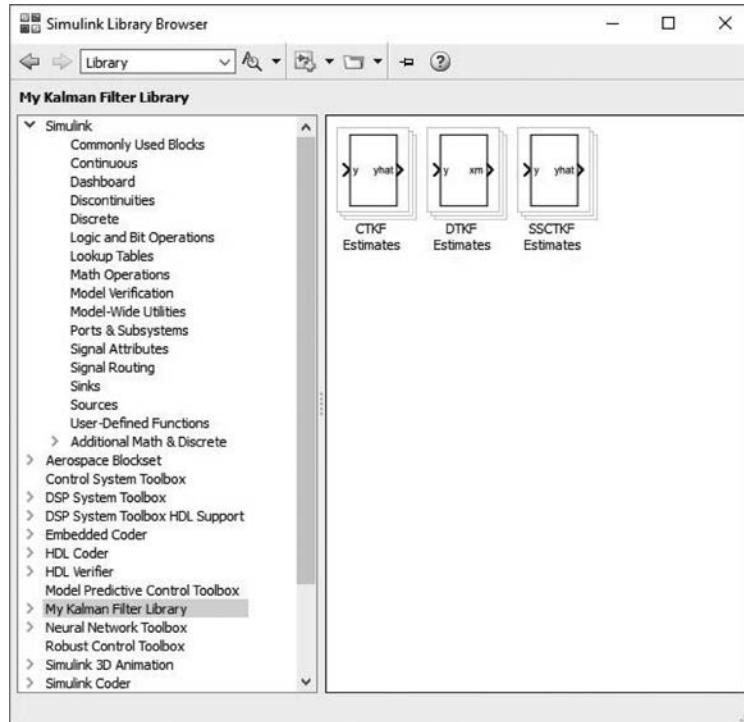


FIGURE 7.87 Simulink Library Browser with my Kalman filters.

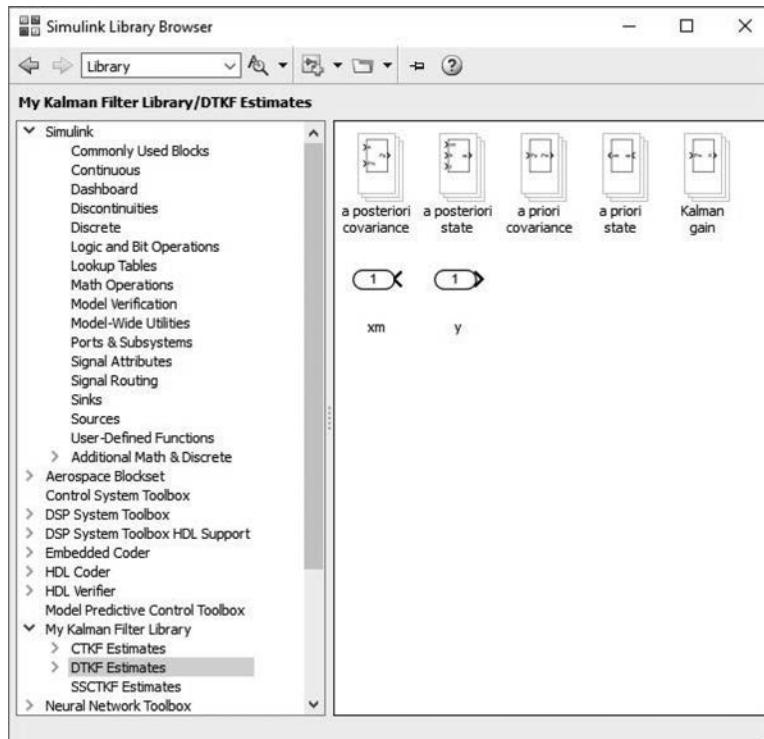


FIGURE 7.88 DTKF estimates subsystems.

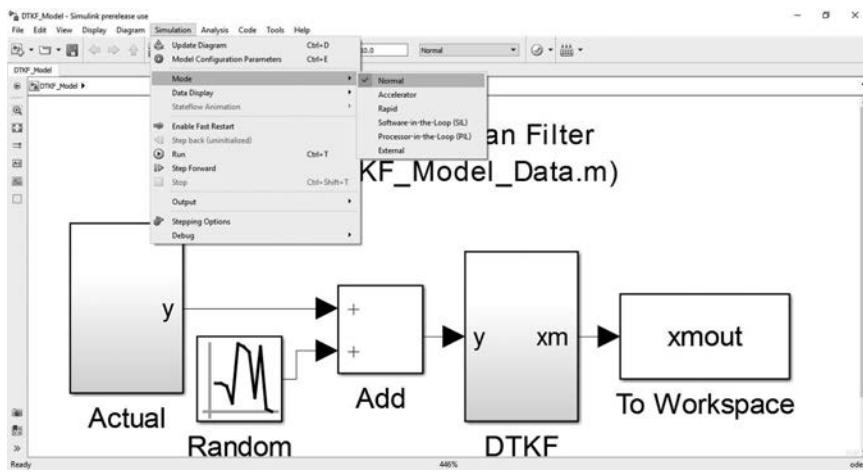


FIGURE 7.89 Normal simulation.

“DTKF Estimates,” and “SSCTKF Estimates,” which are now available to drag and drop into a Simulink model.

Simulink Library Browser.

By double-clicking on “DTKF Estimates” in the right window of the Simulink Library Browser, the subsystems of the algorithm (Kalman gain, a posteriori covariance, a posteriori state, a priori covariance, and a priori state) are displayed in the window as shown in [Figure 7.88](#). These are also available for dragging and dropping for developing Simulink models.

7.5.2 SUMMARY

This section demonstrated how to create a Simulink library and add it to the Simulink Library Browser, thereby making custom models available to other members of a development team.

EXERCISE

- 7.28 In Simulink, create a simple model for the equation of a line $y = mx + b$ where x is the input signal, m is a gain block on the input signal, b is a constant block added to the output of the gain block, and y is the output signal. Once the model is built, create a library named “linelib” and add it to the Simulink Library Browser by editing the slblocks.m file accordingly.

7.6 SIMULATION ACCELERATION

7.6.1 INTRODUCTION

The default simulation option in Simulink is Normal mode. It is set by clicking Simulation Normal in Simulink as shown in [Figure 7.89](#) for the discrete-time Kalman filter model from Section 5.12. In this mode, the simulation is executed as a single (interpreted) process within the MATLAB/Simulink environment. Normal mode supports debugging, M-files, scopes/viewers, run-time diagnostics, parameter tuning, and algebraic loops. However, depending on the level of fidelity built into the Simulink model, or depending on how many replications of the Simulink model are run, the simulation could consume a lot of the user’s time.

Simulink offers two *compiled* options: Accelerator mode ([Figure 7.90](#)) and Rapid Acceleration mode ([Figure 7.91](#)).

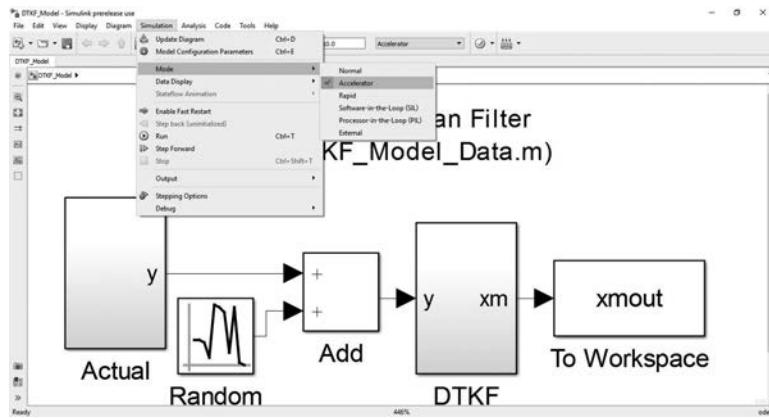


FIGURE 7.90 Accelerator mode simulation.

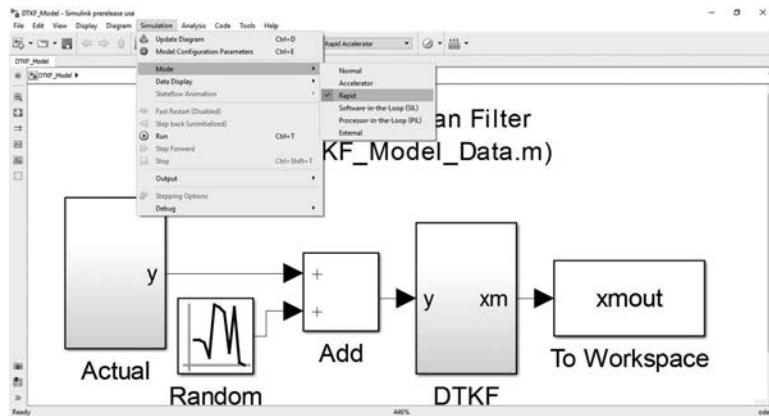


FIGURE 7.91 Rapidly accelerated simulation.

In Accelerator mode, the simulation executes as a single (compiled) process within the MATLAB/Simulink environment. This mode supports debugging, M-files, and scopes/viewers, and allows the user to tune parameters. In order to run the simulation in Accelerator mode, simply select Simulation → Accelerator in Simulink as shown in Figure 7.90 and execute the model.

In Rapid Accelerator mode, the simulation executes as two separate processes: MATLAB/Simulink running as one process while another compiled process runs in parallel. This mode only supports scopes/viewers and parameter tuning. In order to run the simulation in Rapid Accelerator mode, simply select Simulation → Rapid Accelerator in Simulink as shown in Figure 7.91 and execute the model.

After selecting Accelerator mode for the discrete-time Kalman filter model, the MATLAB Command Window displays the following message just before running the simulation.

```
### Building the Accelerator target for model: DTKF_Model
### Successfully built the Accelerator target for model: DTKF_Model
```

This message (with italics added) indicates that Accelerator mode was selected and that a compiled version was created.

After selecting Rapid Accelerator mode for the discrete-time Kalman filter model, the MATLAB Command Window displays the following message just before running the simulation.

```
### Building the rapid accelerator target for model: DTKF_Model
### Successfully built the rapid accelerator target model: DTKF_Model
```

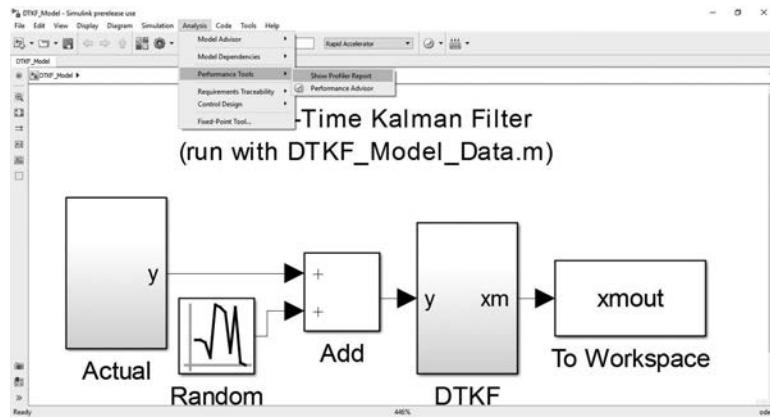


FIGURE 7.92 Simulink’s profiler.

This message (with italics added) indicates that Rapid Accelerator mode was selected and that a compiled version was created. Note that while both Accelerator mode and Rapid Accelerator mode use aspects of MATLAB’s Real-Time Workshop, the user does not need Real-Time Workshop to accelerate simulations. However, the user does need Real-Time Workshop to generate source code for other purposes.

One final comment is that the Rapid Accelerator mode lends itself toward running Monte Carlo simulations. Please see Section 5.10 for more information.

7.6.2 PROFILER

Sometimes, the user would like to know which sections of the simulation are consuming the most time. Simulink provides a tool called the Profiler for such analysis. To turn on the Profiler, click Tools → Profiler in Simulink (Figure 7.92) and rerun the simulation.

Output from the Profiler for the discrete-time Kalman filter simulation is shown in Figure 7.93. By examining this information, the user can take action (e.g., change an algorithm) to help reduce the amount of time the simulation is spending in any one area.

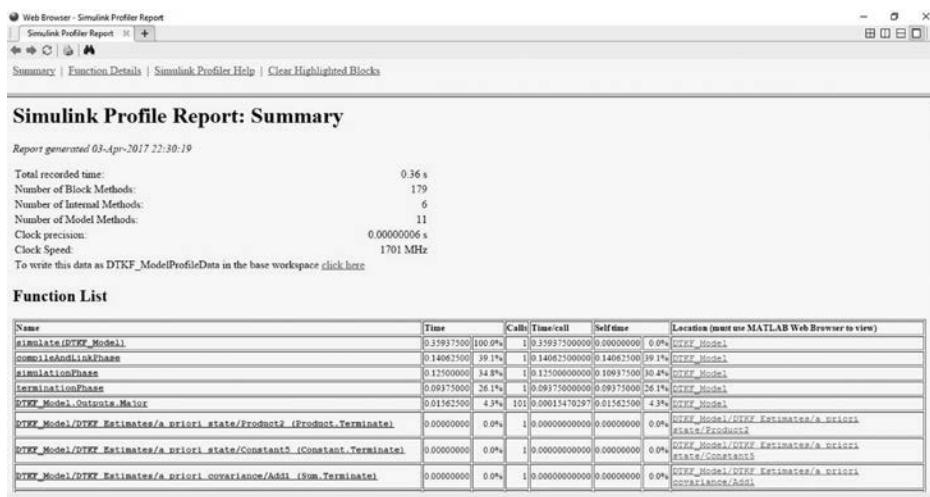


FIGURE 7.93 Profiler output (partial).

7.6.3 SUMMARY

This section demonstrated how to accelerate simulations using the Accelerator mode and the Rapid Accelerator mode. While both modes are compiled and are generally faster than the Normal (interpreted) mode, which one to use depends on the type of tool support needed. Both compiled modes support scopes/viewers and parameter tuning, but only Accelerator mode adds debugging and M-file support.

This section also briefly mentioned the Profiler—a tool to assist users in examining which areas of the simulation require the most amount of execution time. With this knowledge, the user can augment algorithms to increase simulation performance. This can pay dividends, particularly if the user is running many replications of a Monte Carlo simulation with a high level of fidelity.

EXERCISE

- 7.29 Open the discrete-time Kalman filter from Section 5.12.
- a. Run the model in Normal mode.
 - b. Run the model in Accelerator mode.
 - c. Run the model in Rapid Accelerator mode.
 - d. Turn on the Profiler and run the model to see where the model spends most of its time.

7.7 BLACK SWANS

7.7.1 INTRODUCTION

A black swan is an improbable event with colossal consequences. It is a metaphor for believing something is impossible until the belief is disproven. For example, all swans were assumed to be white, and black swans were thought to be non-existent until discovered in Western Australia. In *The Black Swan*, Taleb (2010, p. xxii) defines three black swan attributes: (1) It's an outlier, outside the realm of expectation because nothing in the past points to its possibility; (2) It brings extreme impact; and (3) We concoct explanations for it after the fact, making it seem predictable. In short, the three attributes are: “rarity, impact, and retrospective apparent predictability” (Taleb 2010). Also, by symmetry, non-occurrence of a seemingly certain event is also a black swan. Furthermore, lack of evidence of black swans doesn't mean they do not exist.

While swans of unusual character are labeled black, Benoit Mandelbrot (Wright 2007) claims we can predict something of their behavior – and in doing so, they are no longer black, but can be thought of as gray. They only seem black if we fail to acknowledge their potential existence or we fail to look. Modeling and simulation tools applied to examine potential black swans are inherently stochastic – they are based on probabilistic inputs and outputs and are viewed from a statistical perspective.

An appropriately visualized model architecture may identify the succession of events leading to the colossal black swan which may be confirmed by examination of the Percent Point Function. Additional help with exposing black swans is available through stochastic optimization, where random variables appear in the formulation of the optimization problem thus producing a random objective function for which random iterates are employed to solve the problem. When equipped with these tools and their insights, we can reduce the surprise of a black swan, rendering it gray, and thus be prepared for black swan resiliency.

7.7.2 MODELING RARE EVENTS

With a general understanding of black and gray swans, a model is created such that rare events may be simulated. We need a distribution that shows non-zero probabilities for data lying far from the mean on either side. To make our points, we use the financial markets because most of us can

relate to risk versus reward from a financial perspective. We will see that if thin tail distributions are used, risk is modeled too conservatively; whereas a fat tail distribution exposes a greater degree of risk and the potential for a black swan. The analogy can be generalized to a portfolio of lines of business, where different business opportunities may be assessed for return on investment, given its corresponding risk.

A recent example of a black swan event is the financial collapse of 2008. While we're not interested in the cause of the collapse, *per se*, we are interested in one of the many lessons learned, i.e. observance of the fat tail distribution as a more accurate representation of the collapse, and thereby categorize it as a black swan event. While implementing a normal distribution in a Monte Carlo simulation is far superior to simply using average values of risk and return (Savage 2009), the “thin tail” of the normal distribution assigns negligible probability to data far from the mean. Harry Markowitz (Markowitz 1952; Markowitz 1979; Markowitz 1999) consistently warned that distribution selection was tricky and urged that when moving from theory to practice, some caution was warranted. Benoit Mandelbrot (Mandelbrot, 1963) found price changes in some markets (especially cotton futures) were well described by Lévy stable distributions. Eugene Fama (Fama 1963) performed similar research to what is presented here and further demonstrated the merits of “fat tail” distributions in stocks. Paul Kaplan (Kaplan 2012) shows a log-stable distribution (see Appendix) captures the non-zero probability of occurrence for rare events far from the mean. The log-stable is a generalization of the log-normal distribution commonly used to model investment return. It assumes the logarithm of one plus the decimal form of risk and return following what Mandelbrot referred to as a stable Paretian distribution (Wright, 2007).

The top pane of [Figure 7.94](#) shows monthly returns of the S&P 500 stock index from January 1926 to March 2009 as represented by the frequency histogram. Historical returns over this time period include maximum monthly losses of 26% in November 1929, 24% in April 1932, 20% in October 2008, and 19% in December 1931; while the maximum gain was +50% in August 1932. The data is fitted with a normal distribution (dashed line) and a log-stable distribution (solid line). In the bottom pane, a closer examination displays the characteristics of the normal distribution's thin tail (dashed) versus the log-stable distribution's fat tail (solid). The normal distribution shows a negligible probability of losses beyond 16% (-0.16). While the theoretical tail of the normal distribution extends to infinity, it is clear from this exploded view that the probability of a 16% loss is practically zero. Using the normal distribution curve fit parameters, the actual probability is only 1.4%. Contrast this with the log-stable distribution which more accurately shows losses beyond 26% are indeed possible. In fact, using the log-stable curve fit parameters, the probability of a 16% loss is 9.6%—almost seven times more likely to occur.

From this analysis, we conclude that the log-stable distribution is superior to the normal distribution for modeling rare events of this type. While we have shown this to be true for the S&P 500, if rare events have a non-zero probability of occurrence in any practical application, the log-stable distribution should certainly be considered as the apparatus of choice.

7.7.3 MEASUREMENT OF PORTFOLIO RISK

In order to show the impact of portfolio risk, we model an aggressive asset allocation with the percentages shown in [Table 7.8](#), where each asset class is represented by a corresponding Exchange Traded Fund (ETF) ticker symbol, e.g., small cap stocks are represented by the ETF ticker symbol IWM, etc. Historical monthly return data for each ETF was obtained from the Investools/TD Ameritrade database.

Furthermore, we set up two portfolios according to this asset allocation: one called the Normal Portfolio where historical monthly returns of each asset class are fitted with normal distributions and the other, called the Log-Stable Portfolio, where the historical returns are fitted with log-stable distributions. The following plots were generated from 10,000 Monte Carlo trials.

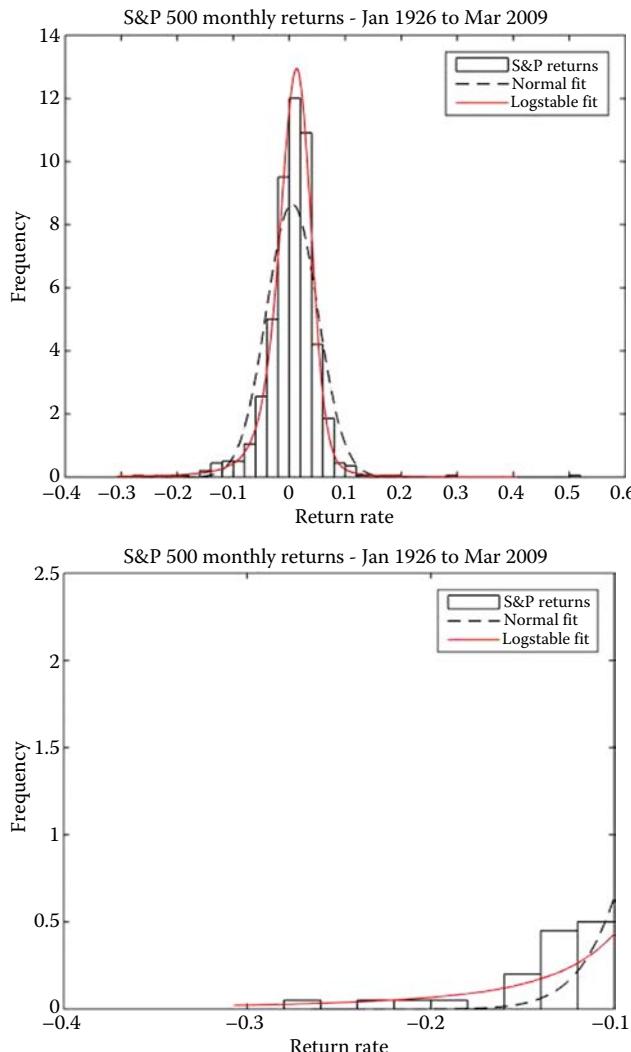


FIGURE 7.94 S&P 500 data from January 1926 to March 2009 fitted with Normal (top) and Log-Stable (bottom) Distributions.

TABLE 7.8
Portfolio Asset Allocations and ETFs

Asset Class	Allocation	ETF
Small Cap Stocks	20%	IWM
Mid Cap Stocks	15%	MDY
Large Cap Stocks	5%	SPY
Int'l Developed Stocks	5%	EFA
Int'l Emerging Stocks	10%	EEM
Corporate bonds	15%	LQD
Government Bonds	10%	AGZ
Real Estate	10%	IYR
Commodities	10%	DBC

From [Figure 7.95](#), it is difficult to tell the difference between the Normal Portfolio and the Log-Stable Portfolio.

However, upon zooming-in, we see (top pane of [Figure 7.96](#)) the maximum loss for the Normal Portfolio is 7% (-0.07) with a maximum gain of 7% (not shown). One may recall the normal distribution is characterized by the mean plus or minus the standard deviation and is therefore symmetric about the mean. The Log-Stable Portfolio (bottom pane of [Figure 7.96](#)) shows a maximum loss of 71% (-0.71) with a maximum gain of 16% (not shown). This left-skew (-71% versus $+16\%$) is based on the data and the characteristic parameters of the log-stable distribution: alpha, beta, gamma, and delta (see Appendix of this chapter).

For the S&P 500 data from January 1926 to March 2009, $\alpha = 1.5901$, $\beta = -0.5586$, $\gamma = 0.0219$, and $\delta = 0.0023$. Here, we see the negative value of beta as representing the left skew (see Appendix) corresponding to more risk than reward. If the data had been such that beta was positive, the distribution would have been right skewed with returns being greater than risk

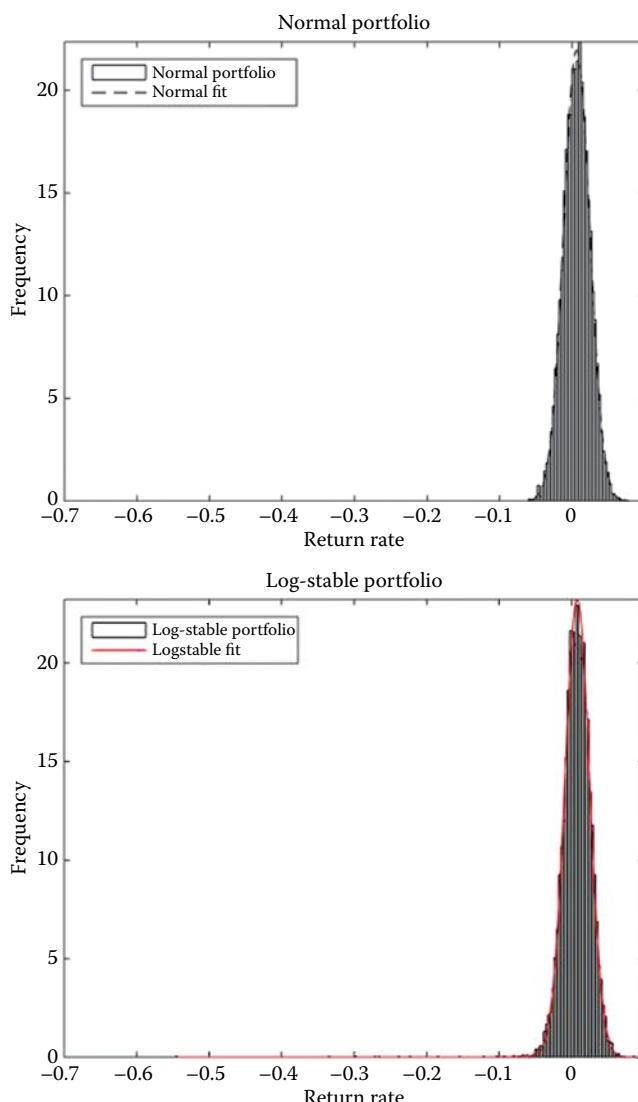


FIGURE 7.95 Portfolios of Asset Allocations Fitted with Normal (top) and Log-Stable (bottom) Distributions.

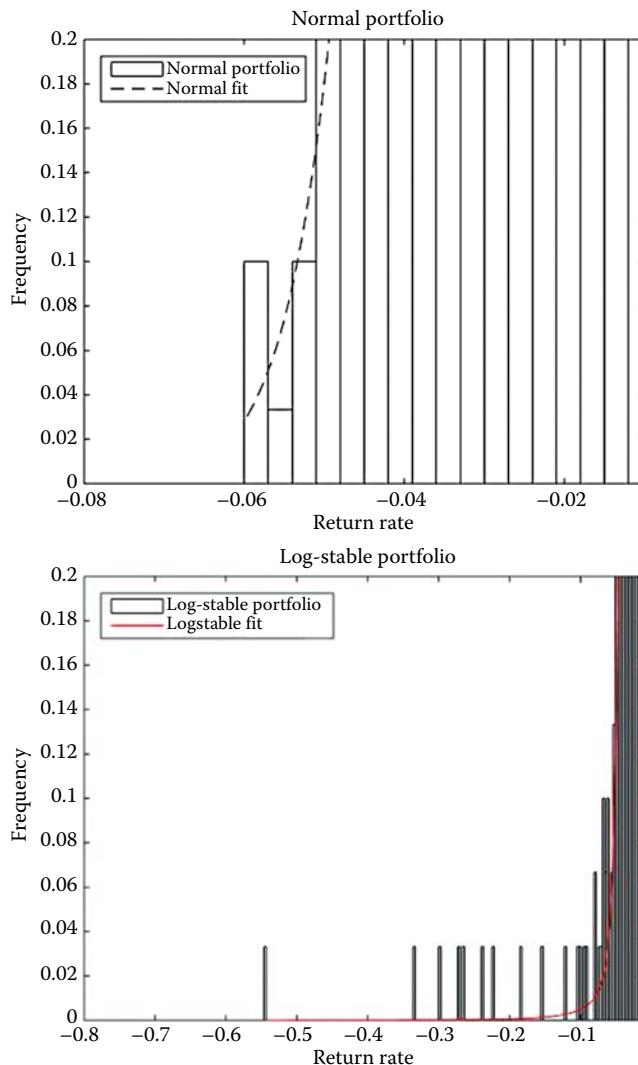


FIGURE 7.96 Zoom of Asset Allocations Fitted with Normal (top) and Log-Stable (bottom) Distributions.

(an elusive investment). Incidentally, fat tails can occur on either or both sides of the distribution, depending on the data being fitted to the log-stable distribution.

Importantly, the results illustrate how the log-stable distribution more accurately predicts high levels of risk, ten-fold. Both normal and log-stable distributions are fitted to the same historical return data. Yet, the normal distribution models risk at only 7%, while the log-stable more accurately characterizes the risk at 71%, for the portfolio. This sheds a little light on the 2008 financial collapse from a risk versus reward perspective. There was actually a higher degree of risk present than otherwise indicated by normal distributions.

Again, while a financial portfolio has been used to show how the log-stable distribution is superior to the normal distribution, if rare events (e.g., earthquake magnitudes, city populations, sizes of power outages, etc.) have a non-zero probability of occurrence (either left-skewed or right-skewed), fitting the event data to a log-stable distribution will model these characteristics, raising our awareness and allowing to prepare, organize, train and equip for black swan resiliency.

7.7.4 EXPOSING BLACK SWANS

This is all rudimentary with simple portfolios of historical returns separated into normal and log-stable distributions. What if your model of influential architecture contains thousands of inputs, including normally distributed data as well as (rare event) log-stable distributed data?

In this section, we model a portfolio with all asset classes fitted to a normal distribution except for one asset class in order to see which tools are useful for finding which input is causing the downside risk. The tools available are Tornado charts, Percent Point Functions, and stochastic optimization methods.

7.7.4.1 Percent Point Functions (PPFs)

A PPF shows the probability of a random number being less than or equal to a particular point on the plot. For example, in [Figure 7.97](#) (top pane), there is a 50% probability the return will be 1% or less and there is a 90% probability the return will be 3% or less. The latter statement could be interpreted as a 10% probability the return will be greater than 3%.

By examining [Figure 7.97](#) (top pane), it appears as if the Normal and Log-Stable PPFs are identical. However, upon closer inspection (bottom pane), we see the Normal Portfolio tail stops near -7%, while the Log-Stable Portfolio tail continues down to -71%. Similarly, but not shown, the positive tails for the Normal and Log-Stable PPF returns are 7 and 16%, respectively. Once more, we've shown the log-stable distribution reveals larger risk, ten-fold for the portfolios. Therefore, the method of "PPF tail inspection" is a viable method to see if potential rare events (black swans) might be lurking in the data.

Rather than comparing Normal and Log-Stable Portfolios, we now blend two portfolios – one with historical monthly returns for all asset classes fit to normal distributions except for government bond returns (AGZ) which are fit with the log-stable distribution; the other portfolio will fit only real estate returns (IYR) to the log-stable distribution. These two (separate, but mixed) portfolios are selected knowing ahead of time government bonds have the lowest (historical) spread between risk and reward (-2 to 4%), while real estate has the highest (historical) spread (-31 to 29%). The reason for this is to examine the tails to see if any black swans might be identified, independent of risk and reward spread.

In the case of the AGZ Mixed Portfolio ([Figure 7.98](#), top pane), we see it has significantly more risk (-27%) than the Normal Portfolio, which had 7% to the downside ([Figure 7.96](#), top pane). Even by modeling a less volatile asset class with the log-stable distribution, we are able to see additional risk through the lens of the PPF tail. Of course, for the IYR Mixed Portfolio ([Figure 7.98](#), bottom pane) the risk is more pronounced (-48%) due to its higher volatility and by virtue of being modeled with a log-stable distribution. The risk of IYR alone is driving the risk of the entire portfolio. In sum, the PPF tail exposes the possibility of the occurrence of a rare event.

7.7.4.2 Stochastic Optimization

Finally, we come to the method of stochastic optimization, where random variables appear in the formulation of the optimization problem thus producing a random objective function for which random iterates are employed to solve the problem. When optimizing (maximizing or minimizing) an objective function (portfolio), stochastic optimization assesses the range of each probabilistic input and selects whatever values are necessary to yield the desired result. For example, to find the minimum portfolio value, the minimum historical return of each asset class will be chosen.

For a small portfolio of only nine asset classes, it is rather simple to calculate deterministic minimum and maximum portfolio returns. [Table 7.9](#) (below) shows minimum and maximum historical returns for each ETF. Based on the asset allocation we've been using (repeated in the table), the minimum and maximum portfolio returns are -22 and 14%, respectively.

The results of running stochastic optimization on this small portfolio were identical with the deterministic case, i.e. minimum and maximum returns of -22 and 14%, respectively. Using

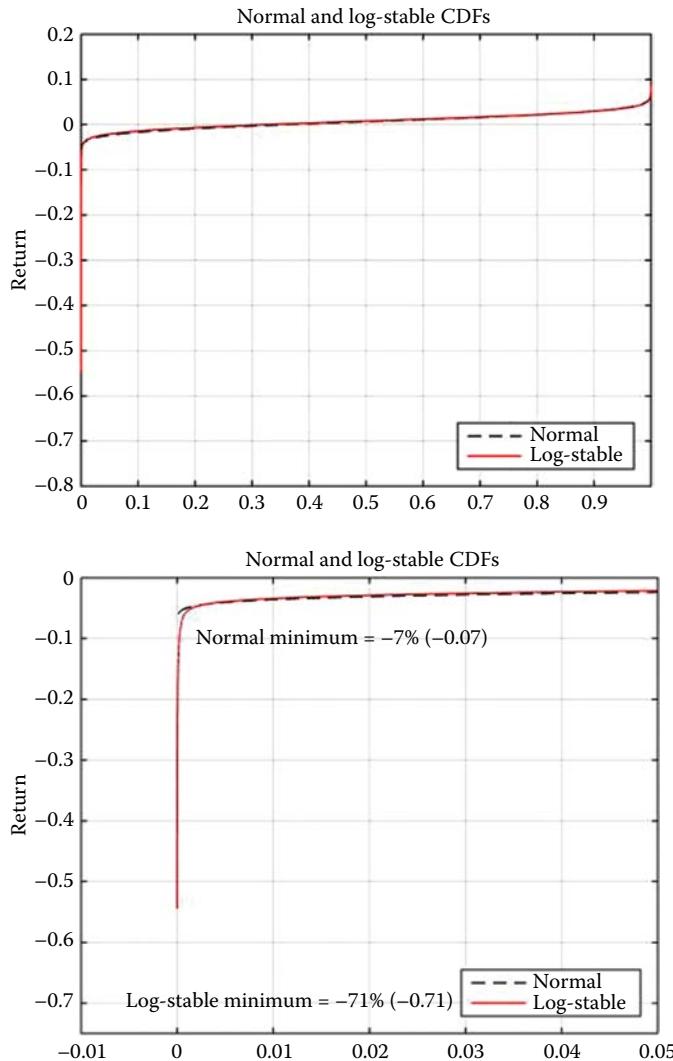


FIGURE 7.97 PPFs for Normal and Log-Stable Portfolios.

stochastic optimization for this sized problem is excessive. But, if a portfolio contains thousands of random inputs, including complex interconnections, it soon becomes intractable to perform these calculations with a spreadsheet, let alone by hand. The result of stochastic optimization is a list of all the inputs and the values that have been chosen so as to achieve either the minimum or maximum return.

The astute reader will wonder how the stochastic optimization risk and return range (–22 to 14%) relates to the prior results of the Normal (–7 to 7%) and Log-Stable (–71 to 16%) Portfolios. We've already discussed the normal distribution and how it naively characterizes risk. This explains why the risk and return range is lower than either of the other results. To explain the difference between the stochastic optimization and the Log-Stable Portfolio results, we recognize the historic lows and highs from Table 7.9 are bounded. For example, when stochastic optimization seeks a minimum, the algorithm selects minimal values of the inputs, which are the lower bounds. Likewise, for the maximum portfolio value, upper bounds are chosen. Stochastic optimization is dependent on the bounded values of the inputs. Compare this to the log-stable distribution which can return values beyond these limits, albeit with small (but non-zero) probability. Even though the historical data is

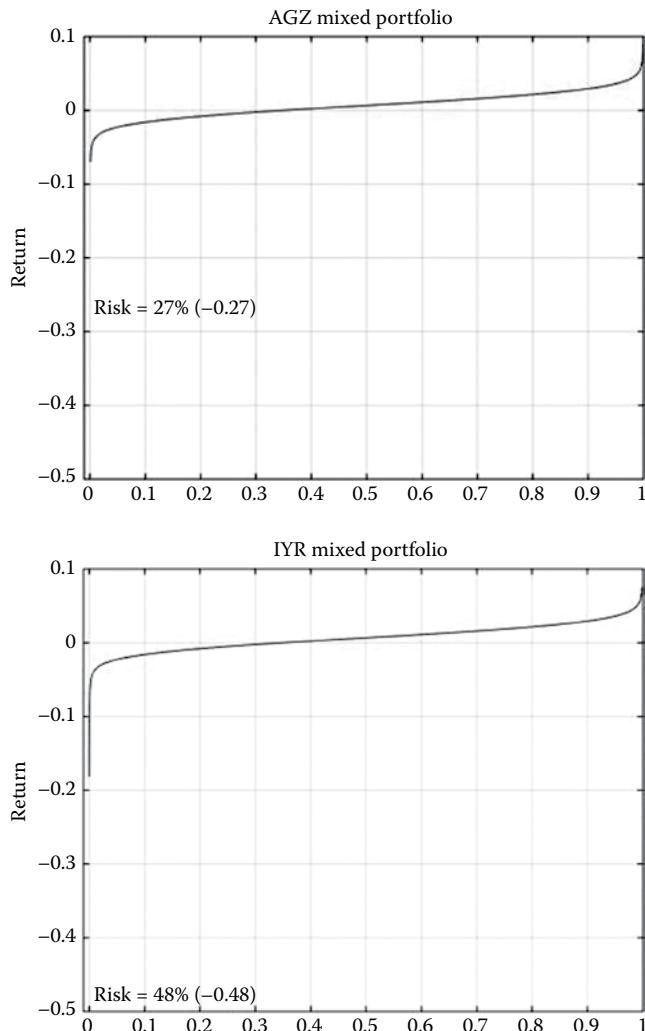


FIGURE 7.98 PPFs for AGZ (top) and IYR (bottom) Portfolios.

TABLE 7.9
Allocated Portfolio Minimum and Maximum Returns

ETF	Allocation	Minimum	Maximum
IWM	20%	-24%	14%
MDY	15%	-24%	15%
SPY	5%	-18%	10%
EFA	5%	-23%	12%
EEM	10%	-29%	16%
LQD	15%	-11%	13%
AGZ	10%	-2%	4%
IYR	10%	-38%	26%
DBC	10%	-29%	15%
Portfolio		-22%	14%

bounded, the log-stable distribution fits the data with parameters which allows random numbers to be drawn in excess of these bounds—again, with small, but non-zero probability. Think of it this way, while the largest earthquake on record is magnitude 9.5, the possibility exists for a larger earthquake to occur—we just haven't experienced it, yet.

7.7.5 SUMMARY

We briefly discussed black swans and their attributes, showing it's possible to identify potential black swans and in doing so, render them gray.

We showed the log-stable distribution is preferred to the normal distribution when it comes to modeling data that includes rare events lying far from the mean. The log-stable achieves this by assigning a non-zero probability of occurrence with its fat tail, whereas the normal distribution assigns a negligible probability due to its thin tail. We showed practical (financial) applications of log-stable modeling for both individual data sets (S&P 500) as well as a portfolio comprised of data sets (ETFs).

Finally, we discussed how PPFs and stochastic unconstrained optimization and their insights can reduce the surprise of a black swan, rendering it gray.

In the end, we have shown how modeling and simulation can be used to analyze and prepare or create a black swan and in practice, we've developed models and simulation tools that enable the analysis.

7.7.6 ACKNOWLEDGEMENTS

We wish to thank Dr. Paul Kaplan (Morningstar) and Dr. John Nolan (University of Virginia) for personal email correspondence. We wish to recognize Investools/TD Ameritrade as the database from which we were able to obtain historical monthly data for each asset class.

7.7.7 REFERENCES

- Fama, E., Mandelbrot and the stable paretian hypothesis. *Journal of Business*, 36(4), 420–429, 1963.
- Kaplan, P., *Frontiers of Modern Asset Allocation*. Hoboken, NJ: John Wiley & Sons, Inc., 2012.
- Kaplan, P., Using fat tails to model gray swans. Retrieved from <http://morningstardirect.morningstar.com/clientcomm/LogStableDistributions.pdf>, 2008.
- Mandelbrot, B., The variation of certain speculative prices. *Journal of Business*, 36(4), 394–419, 1963.
- Markowitz, H., Portfolio selection. *Journal of Finance*, 7(1), 77–91, 1952.
- Markowitz, H., Approximating expected utility by a function of mean and variance. *The Economic Review*, 69(3), 308–317, 1979.
- Markowitz, H., The early history of portfolio theory: 1600–1960. *Financial Analysts Journal*, 55(4), 5–16, 1999.
- Nolan, J., User manual for STABLE 5.1: MATLAB version. Retrieved from www.robustanalysis.com/MatlabUserManual.pdf, 2009b.
- Savage, S., *The Flaw of Averages*. Hoboken, NJ: John Wiley & Sons, Inc., 2009.
- Taleb, N., *The Black Swan: The Impact of the Highly Impossible*. NY: Random House Inc., 2010.
- Veillette, M., STBL: Alpha stable distributions for MATLAB. Retrieved from http://www.mathworks.com/matlabcentral/fileexchange/37514-stbl--alpha-stable-distributions-for-matlab/all_files, 2015.
- Wright, C., Tail tales. *CFA Institute Magazine*, 2007, March/April.

7.7.8 APPENDIX—MATHEMATICAL PROPERTIES OF THE LOG-STABLE DISTRIBUTION

The log-stable distribution is frequently used to model investment returns. Returns are expressed in decimal form, where negative returns represent losses and positive returns represent profit. We then normalize the returns by adding one and taking the natural log of the result. Once in this form, the

returns conform to a stable distribution. The probability density function (pdf) for a (fat-tail) stable distribution is

$$f(x; \alpha, \beta, \gamma, \delta) = \frac{1}{\gamma} g\left(\frac{x-\delta}{\gamma}; \alpha, \beta\right)$$

α represents the “fatness” of the tails and is in the range between 0 and 2, with 2 being a normal distribution. Also, if $\alpha < 1$, then the mean of distribution is infinite.

β represents the skewness of the distribution and lies within the range of -1 to 1 , where -1 signifies fully left-skewed and $+1$ signifies fully right-skewed. If $\beta = 0$, the distribution is symmetric.

γ represents the scale of the distribution and is positive. If $\alpha = 2$ (normal), then γ^2 is one-half the variance.

δ represents the location of the distribution. If $\alpha > 1$, then δ is the mean of distribution.

7.8 THE SIPMATH STANDARD

7.8.1 INTRODUCTION

This section discusses a means for efficiently representing uncertainty as probability distributions: Stochastic Information Packets (SIPs) are arrays of realizations of uncertainties, for example Monte Carlo simulation trials. Sets of SIPs, which maintain statistical dependence are known as Stochastic Library Units with Relationship Preserved (SLURPs). The SIPmath standard enables legacy and future simulation models to communicate with each other.

Strings of numbers representing uncertainty and probability distributions have been used at least since 1991 (Dembo, 1991). In 2005, the use of number strings (SIPs and SLURPs) was extended to drive interactive simulations for high-level decision-makers at Royal Dutch Shell (Savage, Scholtes, & Zweidler, 2006). Subsequently, the discipline of probability management was formalized. The approach is further described in *The Flaw of Averages, Why We Underestimate Risk in the Face of Uncertainty* (Savage, 2009) and *Calculating Uncertainty: Probability Management with SIP Math* (Thibault, 2013).

SIPs advance the modeling of uncertainty in four fundamental ways:

Actionable—SIPs may be used directly in calculations involving uncertainty on numerous platforms

Additive—SIPs allow uncertainties to be aggregated across platforms across the enterprise

Auditable—Uncertainties are represented as unambiguous data with provenance

Agnostic—Platform independence

Beyond modeling uncertainty, the SIP (as a vector array) makes data from one database/simulation easily accessible to other databases/simulations, thereby facilitating the movement of potentially large and unstructured data between distributed simulators. SIPs/SLURPs can accommodate both “big data” and data as small as a single number. Furthermore, conversion between file types (currently XLSX, CSV, XML, JSON) is facilitated with ease. SIP/SLURP formats have been successfully used with a wide range of software and simulation types, including R, MATLAB, Autobox, and other proprietary software.

The SIP standard is open, neutral, and not tied to any particular format or firm. It is sponsored by Probability Management (PM), a non-profit. There is no fee or license to use SIPs, and the standard is freely available at www.probabilitymanagement.org.

7.8.2 STANDARD SPECIFICATION

The purpose of the specification is to define standards for probability distributions as auditable and transportable data. The standard defined herein is for the Stochastic Information Packet (SIP) and

the Stochastic Library Unit with Relationships Preserved (SLURP). The standard defines a simple, adaptable data architecture that makes it easy to create and use SIP libraries by piggybacking on common data formats: CSV, XML, and XLSX (Excel Worksheets). The open SIPmath™ 2.0 Standard may be downloaded from the Probability Management web site <http://probabilitymanagement.org/library/SIP-Standard-Version2.pdf>.

While the standard was created to support simulation and analysis dealing with uncertainty (“Stochastic Processes”) any data can be archived using the specification. The features of storing a string or table of numbers, or even a single value with all of the descriptive information is valuable. The SIP provides a way to deliver units, the name of the variable, data provenance, and other information. Even when the data delivered in a SIP doesn’t “seem” stochastic, it provides a useful way to create open interfaces among simulation tools and organizations. Because information in one format can easily and reliably be translated to another format, the cost and barriers to information sharing are reduced.

The SIP provenance described below is the “data about the data.” This is one of the most valuable features of the specification standard. It provides a way to communicate important information about the origins, vintage, and details of the data. There are two fields in the specification for providing “data about the data,” one is the “about” field and the other is “provenance” field.

7.8.3 SIP DETAILS

The Stochastic Information Packet (SIP) represents a probability or frequency distribution as a data structure that holds an array of values and metadata. In the current standard, the values are realizations of the possible outcomes of an uncertain variable. The array for a probability distribution is composed so that the default probability of each element is $1/N$ where N is the number of elements in the array. The key benefit of using SIPs is that they are actionable, in that they may be used in calculations. If X is a random variable represented by $\text{SIP}(X)$, and $F(X)$ is a function of X , then $\text{SIP}(F(X)) = F(\text{SIP}(X))$. That is, the function, F , is applied sequentially to each element of $\text{SIP}(X)$. This means in effect that SIPs and the arithmetic, relational, and logical operators comprise a group [Tables 7.10](#) and [7.11](#).

7.8.4 SLURP DETAILS

A coherent set of SIPs that preserve statistical relationships between uncertainties is known as a Stochastic Library Unit with Relationships Preserved (SLURP). Two or more SIPs are coherent if the values of their corresponding samples are in some way interdependent. For calculations with these SIPs to be valid, the alignment of the samples must be preserved; if one of the SIPs is permuted, the others must be permuted by the same permutation index to preserve coherence. In this respect, the importance of the SLURP is that any SIP calculated with arithmetic, relational, or logical operations on SIPs in a given SLURP will also be coherent with that SLURP. Two attributes are required: name and coherent; one is optional: count [Table 7.12](#).

TABLE 7.10
SIP Standard Attributes

Name	Description
name	Required. A text string identifying the SIP, usually unique in context
count	Required. The number of samples
type	Required. The format type
ver	Required. The format version

TABLE 7.11
Common Optional Attributes

Name	Description
about	A description of the SIP or SLURP
avg	The average or mean of the SIP sample values before they're encoded into the string
csvr	The number of digits to the right of the decimal for CSV conversion
dataver	A number or date indicating the currency of the data in a SIP or SLURP
dims	The dimensions of a multidimensional SIP
hbin	The bin width of a histogram of the SIP
hmin	The minimum value in a histogram of the SIP
hnum	The number of bins in a histogram of the SIP
hvalN	The value in the Nth bin in a histogram of the SIP
max	The SIP maximum sample value
min	The SIP minimum sample value
offset	An offset factor to be applied to a SIP encoded value to get the sample value. The “b” in $ax + b$. Default is 0.
origin	An arbitrary text string should say something about the institution or project that produced a SIP or SLURP
provenance	Information about the source and authority of the data
Ptile	The (P/100) percentile
scale	A scale factor to be applied to a SIP encoded value to get the sample value. The “a” in $ax + b$. Default is 1.
units	A text string for the SIP data measurement units e.g., “Can\$” for Canadian dollars

TABLE 7.12
SLURP Standard Attributes

Name	Description
name	Can be any string, should be a unique identifier in context.
coherent	Must be either “true” or “false”. If false, the coherence of the included SIPs is not assured.
count	Optional. The number of SIPs in the SLURP.

7.8.5 SIPs/SLURPs AND MATLAB

To demonstrate how to export data from the MATLAB Workspace directly to an XML SLURP, we've adapted an open-source, econometric forecasting model developed by the Federal Reserve Bank of Philadelphia. The Philadelphia Research Intertemporal Stochastic Model (PRISM) is a research project available for download from <http://www.philadelphiafed.org/research-and-data/real-time-center/prism>

Its output is not an official forecast of the Philadelphia Federal Reserve. Furthermore, our adaptation is in no way sponsored by nor endorsed by the Philadelphia Federal Reserve.

First, we use the econometric forecasting model to generate MATLAB Workspace data representing the state variables: Gross Domestic Product (GDP), Consumption, Investment, Hours Worked, Inflation, and Fed Funds Rate. Each MATLAB Workspace stochastic state variable is represented by 1,000 sample paths with a forecast over several ten quarters. Multiple financial institutions can all access the same data to execute their own individual risk models. Because each model observes the same state of the world on each trial, the results of these models may, in principle, be aggregated into a SLURP of the entire financial sector.

In the vernacular of SIPmath, we will create a GDP SLURP of 1,000 sample paths reflecting the temporal stochastic forecast, comprised of 10 SIPs, one for each quarter. Of course, there would be

SLURPs for the remaining state variables as well, which would reflect cross-correlation between state variables. The union of all these SLURPs therefore form an overall SLURP. However, this example will remain focused on GDP.

If the reader has downloaded and installed the PRISM model, the following MATLAB code was added to the end of the dsgefct.m script:

```
% Define SLURP attributes
attributeSlurpStruct.name = 'GDP';
attributeSlurpStruct.coherent = 'true';
attributeSlurpStruct.count = 1000;
% Define SIP attributes
for j = 1:2
    attributeSipStruct(j).name = strcat('GDP_0', num2str(j));
    attributeSipStruct(j).count = 1000;
    attributeSipStruct(j).type = 'CSV';
    attributeSipStruct(j).ver = '1.0';
    attributeSipStruct(j).dataArray = yfcst2(:,j);
end
slurpXmlCreator(2,attributeSlurpStruct,attributeSipStruct);
```

The slurpXmlCreator function is found in the appendix. Upon execution of the sample code, an XML file named, GDP.xml is created. A portion of the XML file is displayed below. The actual stochastic data for GDP is suppressed to conserve space. Where it reads (1000 data points), there are 1000 comma separated values.

```
<?xml version="1.0" encoding="utf-8"?>
<SLURP name="GDP" coherent="true" count="1000">
<SIP name="GDP_01" count="1000" type="CSV" ver="1.0"> (1000 data points)
</SIP>
<SIP name="GDP_02" count="1000" type="CSV" ver="1.0"> (1000 data points)
</SIP>
...
<SIP name="GDP_10" count="1000" type="CSV" ver="1.0"> (1000 data points)
</SIP>
</SLURP>
```

To fully appreciate the power of SIPmath, the XML SLURP (GDP SIPs) from the PRISM model was imported into an Excel workbook with VBA code written to the specification. The simplicity of the standard was demonstrated by its success on first use, in spite the fact that the MATLAB encoding and VBA decoding algorithms were developed by two independent programmers communicating only through the documented specification. The 1,000 sample paths stored as SIPs, now in Excel, were then run through an interactive SIPmath model that generated two new sets of sample paths through lagged regressions representing a hypothetical predicted metric at each of two financial institutions. These two sets of 1,000 paths were then consolidated to represent the predicted sum of the metrics across all institutions. Consolidating the results of two simulations in this way is not usually possible.

7.8.6 SUMMARY

Interfacing multiple modeling environments together requires two things: (1) that the transfer protocol of data between the modeling environments is standardized, and (2) that the variables of interest are known and well defined. In order to address these requirements, the data should be transitioned to standardized SIP and/or SLURP format. In addition, it is highly recommended

that the definition of data classes expected to be transferred between modeling environments be jointly researched and specified.

7.8.7 APPENDIX

```
function slurpXmlCreator(nSips,attributeSlurpStruct,attributeSipStruct)
% This function generates an XML SLURP given metadata and data according
% to the SIPmath standard written by Marc Thibault of Probability
Management.
% http://probabilitymanagement.org/standards.html
% Randal Allen, 31 March 2017
% Contributor: Soham Chowdhury (dataArray to CSV)
% INPUTS
% nSips - the number of SIPs within each SLURP
% attributeSlurpStruct - the attribute structure of the SLURP (metadata)
% - Required: name, coherent
% - Optional: count, type, ver, about, origin
% attributeSipStruct - the attribute structure of each SIP (metadata and
data)
% - Required: name, count, type, and ver
% - Optional: min, max, avg, about, origin, units, scale, offset
% USAGE (how variables are assigned prior to function call)
% Define the SLURP attributes
% attributeSlurpStruct.name = 'slurp_name';
% attributeSlurpStruct.count = number_of_sips;
% attributeSlurpStruct.type = 'CSV';
% attributeSlurpStruct.ver = '1.2.4';
% attributeSlurpStruct.coherent = 'true';
% attributeSlurpStruct.about = 'info_about_slurp';
% attributeSlurpStruct.origin = 'more_provenance';
% Define the SIP attributes for each (j) of the SIPs (nSips)
% for j = 1:nSips
% attributeSipStruct(j).name = strcat('sip_name_',num2str(j));
% attributeSipStruct(j).count = number_of_sips;
% attributeSipStruct(j).type = 'CSV';
% attributeSipStruct(j).ver = '1.2.4';
% attributeSipStruct(j).about = 'info_about_sip';
% attributeSipStruct(j).min = min(data(j));
% attributeSipStruct(j).max = max(data(j));
% attributeSipStruct(j).avg = mean(data(j));
% attributeSipStruct(j).dataArray = data(j);
% end
% Call this function
% slurpXmlCreator(nSips,attributeSlurpStruct,attributeSipStruct);
% Create the top-level node for the SLURP
docNode = com.mathworks.xml.XMLUtils.createDocument('SLURP');
slurpName = docNode.getDocumentElement;
% Set the SLURP attributes
slurpName.setAttribute('name',attributeSlurpStruct.name);
slurpName.setAttribute('coherent',attributeSlurpStruct.coherent);
slurpName.setAttribute('count',num2str(attributeSlurpStruct.count));
slurpName.setAttribute('type',attributeSlurpStruct.type);
slurpName.setAttribute('ver',attributeSlurpStruct.ver);
slurpName.setAttribute('about',attributeSlurpStruct.about);
slurpName.setAttribute('origin',attributeSlurpStruct.origin);
```

```
% Set the attributes for each SIP of the SLURP
for j = 1:nSips
    sipName(j) = docNode.createElement('SIP');
    sipName(j).setAttribute('type',attributeSipStruct(j).type);
    sipName(j).setAttribute('count',num2str(attributeSipStruct(j).count));
    sipName(j).setAttribute('name',attributeSipStruct(j).name);
    sipName(j).setAttribute('ver',attributeSipStruct(j).ver);
    sipName(j).setAttribute('about',attributeSipStruct(j).about);
    sipName(j).setAttribute('min',num2str(attributeSipStruct(j).min));
    sipName(j).setAttribute('max',num2str(attributeSipStruct(j).max));
    sipName(j).setAttribute('avg',num2str(attributeSipStruct(j).avg));
    [nR, nC] = size(attributeSipStruct(j).dataArray);
    % Convert data array to a row vector for use with vec2str function
    % "round" added to suppress scientific notation
    if nR > nC %if a column vector
        dataArray = round(attributeSipStruct(j).dataArray,5)';
    else
        dataArray = round(attributeSipStruct(j).dataArray,5);
    end
    % Convert the dataArray into a comma-separated string of values
    csvDataString = vec2str(dataArray,[],[],0);
    sipName(j).setTextContent(csvDataString);
    slurpName.appendChild(sipName(j));
end
% Write the SLURP to an XML file.
xmlwrite(strcat(attributeSlurpStruct.name,'.xml'),docNode);
type(strcat(attributeSlurpStruct.name,'.xml'));
end
```

7.8.8 REFERENCES

- Dembo, R. S., Scenario optimization, *Annals of Operations Research*, 30(1), 63–80. <http://link.springer.com/article/10.1007%2FBF02204809>, 1991.
- Savage, S., S. Scholtes, and D. Zweidler, Probability management, *OR/MS Today*, February, 33(1). <http://www.lionhrtpub.com/orms/orms-2-06/frprobability.html>, 2006.
- Savage, S., *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*, Wiley 2009.
- Thibault, J. M., *Calculating Uncertainty: Probability Management with SIP Math*, 2013.
- Schorfheide, F., K. Sill, and M. Kryshko, DSGE Model-based forecasting of non-modeled variables, Research Department, Federal Reserve Bank of Philadelphia, 2008.

8 Advanced Numerical Integration

8.1 INTRODUCTION

Dynamic errors, an important aspect in digital simulation of dynamic systems, are introduced. Instead of focusing on truncation errors, the simulationist may be more concerned with errors in dynamic response, a yardstick of simulation accuracy involving comparisons of transient and sinusoidal responses of continuous-time and discrete-time models.

The subject of dynamic errors has been covered in great detail by Howe (1986). The commonly used numerical integrators are analyzed by considering the characteristic roots, magnitude, and phase properties of the “equivalent continuous-time system,” that is, the continuous-time system whose sampled values coincide with the discrete-time (simulated) system outputs. The connection between digital simulation and discrete-time systems is further illustrated by exploring the subject of stability in both arenas.

Stiff systems, initially introduced in [Chapter 6](#), are once again considered. Multirate integration schemes are presented as an alternative to the use of stiff integrators for the case where the overall system can be decomposed into several interconnected subsystems operating at different speeds.

Real-time simulation is a specialized application involving interactions between a digital simulation and real-time inputs from physical components or a human operator. The necessity of synchronizing with signals to and from external components places additional constraints on the simulation environment and numerical integrators. Real-time compatible numerical integrators are discussed along with numerical integrators not suitable for real-time implementation and an explanation of why they are not.

The chapter concludes with a look at some additional techniques for developing discrete-time models intended to approximate the dynamic behavior of linear time-invariant (LTI) continuous-time models.

8.2 DYNAMIC ERRORS (CHARACTERISTIC ROOTS, TRANSFER FUNCTION)

The use of numerical integrators to simulate the behavior of continuous-time systems introduces errors, that is, the transient and steady-state behavior of the discrete-time responses differs from that of the continuous-time outputs at the times where the simulated response is computed. Some insight with respect to the differences is possible by considering expressions for the truncation errors inherent in the various types of numerical integrators. We know that the local and global truncation errors are sensitive to the integration step size and the state derivative functions which define the continuous-time system model.

The differences in transient and steady-state sinusoidal responses are termed dynamic errors. Truncation errors, on the other hand, relate numerical solutions of differential equation models to various-order Taylor Series expansions of the continuous-time solutions. A mathematical framework for comparing dynamic errors resulting from numerical integration of linear continuous-time models is possible. Given that real-world system models are invariably nonlinear, the first step is therefore to linearize the system of nonlinear differential and algebraic equations about a steady-state operating point, similar to the procedures discussed in [Section 7.4](#).

The dynamic errors associated with the use of fixed-step numerical integrators applied to linear system models fall in one of two categories (Howe 1986). One type of error focuses on differences

between characteristic roots of the continuous-time system model and the apparent or equivalent continuous-time system. By equivalent continuous-time system, we mean the continuous-time system that generates sampled values identical with the discrete-time (simulated) system.

The second type of error relates to differences between the frequency response function of the continuous-time system and the discrete-time system used to approximate its behavior. Only linear first- and second-order systems will be considered because higher-order systems can be represented as linear combinations of these lower-order subsystems.

8.2.1 DISCRETE-TIME SYSTEMS AND THE EQUIVALENT CONTINUOUS-TIME SYSTEMS

Consider a first-order linear system modeled by

$$\frac{dx}{dt} = f(x, u) = \lambda x + u \quad (8.1)$$

The characteristic root is λ , the pole of the system transfer function

$$H(s) = \frac{X(s)}{U(s)} = \frac{1}{s - \lambda} \quad (8.2)$$

Digital simulation of the system requires solution of a difference equation obtained by numerical integration of the state derivative function $f(x, u)$. For explicit Euler integration, the z -domain transfer function of the resulting discrete-time system can be obtained by z -transforming the difference equation

$$x_A(n+1) = x_A(n) + T[\lambda x_A(n) + u(n)] \quad (8.3)$$

or equivalently from (see Section 4.7)

$$H(z) = H(s) \Big|_{s \leftarrow \frac{z-1}{T}} = \frac{1}{s - \lambda} \Big|_{s \leftarrow \frac{z-1}{T}} = \frac{1}{(z - 1/T) - \lambda} = \frac{T}{z - (1 + \lambda T)} \quad (8.4)$$

The discrete-time system pole is located at $z_1 = 1 + \lambda T$.

The equivalent continuous-time system is the system whose output $x(t)$, $t \geq 0$ is identical to the discrete-time output $x_A(nT)$ at times $t_n = nT$, $n = 0, 1, 2, \dots$. To illustrate, suppose the input to the system in Equation 8.1 is $u(t) = 1$, $t \geq 0$. The response is

$$x(t) = \frac{1}{\lambda} [e^{\lambda t} - 1], \quad t \geq 0, \quad (8.5)$$

The use of explicit Euler integration with step size T to approximate the continuous-time step response produces the discrete-time approximation $x_A(n)$, short for $x_A(nT)$, $n = 0, 1, 2, \dots$ obtained from

$$X_A(z) = H(z)U(z) \quad (8.6)$$

$$= \left[\frac{T}{z - (1 + \lambda T)} \right] \frac{z}{z - 1} \quad (8.7)$$

Partial fraction expansion of Equation 8.7 followed by inverse z -transformation of the resulting terms gives

$$x_A(n) = \frac{1}{\lambda} [(1 + \lambda T)^n - 1], \quad n = 0, 1, 2, \dots \quad (8.8)$$

Let the equivalent continuous-time system be described by

$$\frac{dx}{dt} = f(x, u) = \lambda^* x + Ku \quad (8.9)$$

where λ^* and K are the characteristic root and gain parameter of the equivalent first-order continuous-time system, respectively. The step response is

$$x^*(t) = \frac{K}{\lambda^*} [e^{\lambda^* t} - 1], \quad t \geq 0 \quad (8.10)$$

Sampling the equivalent continuous-time system response every T s gives

$$x^*(nT) = \frac{K}{\lambda^*} [e^{\lambda^* nT} - 1], \quad n = 0, 1, 2, \dots \quad (8.11)$$

Equating the discrete-time responses in Equations 8.8 and 8.11,

$$\frac{1}{\lambda} [(1 + \lambda T)^n - 1] = \frac{K}{\lambda^*} [e^{\lambda^* nT} - 1], \quad n = 0, 1, 2, \dots \quad (8.12)$$

Solving for K and λ^* ,

$$e^{\lambda^* nT} = (1 + \lambda T)^n \Rightarrow \lambda^* = \frac{1}{T} \ln(1 + \lambda T) \quad (8.13)$$

$$\frac{K}{\lambda^*} = \frac{1}{\lambda} \Rightarrow K = \frac{\lambda^*}{\lambda} = \frac{\ln(1 + \lambda T)}{\lambda T} \quad (8.14)$$

The step response of the first-order continuous-time system in Equation 8.1 with characteristic root $\lambda = -2$ is shown in [Figure 8.1](#). Also shown is the step response of the discrete-time system in Equation 8.3 corresponding to explicit Euler integration of the derivative function with step size $T = 0.05$ s. The step response of the equivalent continuous-time system in Equation 8.9 with λ^* and K computed from Equations 8.13 and 8.14 is also shown.

From Equation 8.4, the pole of the discrete-time system is $z_l = 1 + \lambda T$. Replacing $1 + \lambda T$ in Equation 8.13 with z_l leads to an expression relating the characteristic root of the equivalent continuous-time system and the pole of the discrete-time system. That is,

$$\lambda^* = \frac{1}{T} \ln z_l \quad (8.15)$$

Solving Equation 8.15 for the discrete-time system pole leads to

$$z_l = e^{\lambda^* T} \quad (8.16)$$

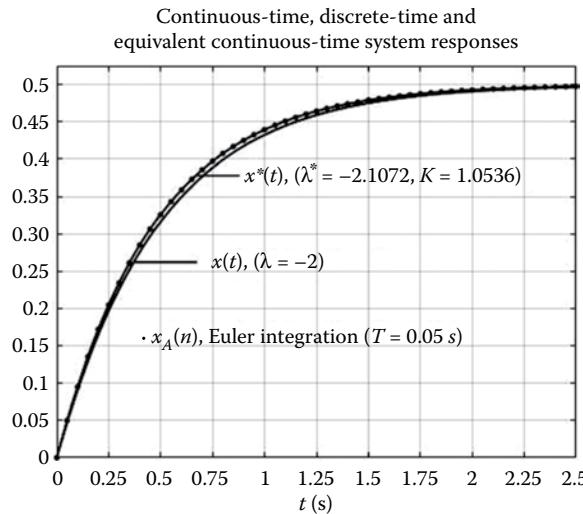


FIGURE 8.1 Step response of continuous-time, discrete-time, and equivalent continuous-time systems.

Uniform sampling of the equivalent continuous-time system response $x^*(t)$ every T s generates the discrete-time system signal $x_A(n)$ with pole z_1 given in Equation 8.16. In the general case, sampling continuous-time signals with real and complex poles produces discrete-time system signals with z -plane poles given by

$$z_i = e^{Ts_i}, \quad i = 1, 2, \dots, n \quad (8.17)$$

where s_i are the poles of the continuous-time system (Jacquot).

Equation 8.17 applies to LTI systems and their characteristic roots as well. The sampled output of a continuous-time system with characteristic root (s -plane pole) s_1 is identical to the output from a discrete-time system with characteristic root (z -plane pole) located at $z_1 = e^{Ts_1}$. Looking at it from the opposite direction, the continuous-time system equivalent to a discrete-time system with a pole z_1 has an s -plane pole at $s_1 = 1/T \times \ln z_1$.

According to Equation 8.17, a continuous-time integrator with a pole at $s = 0$ in the s -plane is the continuous-time system equivalent to a discrete-time system with a pole at $z = 1$. [Figure 8.2](#) illustrates the point by showing that a pure integrator generates a continuous-time signal $x(t)$ in response to the input $u(t)$, which matches the response of the discrete-time system with z -domain transfer function $H(z) = K/(z - 1)$ at the discrete times $t_n = nT, n = 0, 1, 2, \dots$

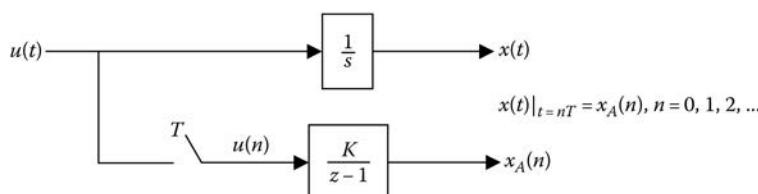


FIGURE 8.2 An integrator as the equivalent continuous-time system to a discrete-time system with pole at $z = 1$.

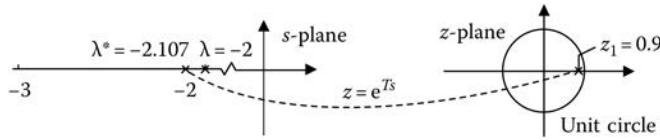


FIGURE 8.3 Mapping $z = e^{Ts}$ for finding the equivalent continuous-time system characteristic root $\lambda^* = -2.107$ when $z_1 = 0.9$, $T = 0.05$.

Suppose $u(t) = e^{-at}$, $t \geq 0$ is the input to the integrator. The output $x(t)$ is

$$x(t) = \int_0^t u(t)dt = \frac{1}{a}(1 - e^{-at}), \quad t \geq 0 \quad (8.18)$$

The discrete-time response is found from inverse z -transformation of

$$X(z) = \left(\frac{K}{z-1} \right) \frac{z}{z - e^{-aT}} = \frac{K}{1 - e^{-aT}} \left[\frac{z}{z-1} - \frac{z}{z - e^{-aT}} \right] \quad (8.19)$$

$$\Rightarrow x_A(n) = \frac{K}{1 - e^{-aT}} (1 - e^{-anT}), \quad n = 0, 1, 2, \dots \quad (8.20)$$

and it follows that $x(nT) = x_A(n)$, $n = 0, 1, 2, \dots$ provided

$$K = \frac{1 - e^{-aT}}{a} \quad (8.21)$$

Figure 8.3 shows the characteristic root of the continuous-time system in Equation 8.2 for the case when $\lambda = -2$. The pole of the discrete-time system resulting from explicit Euler integration with step size $T = 0.05$ is located at $z_1 = (1 + \lambda T) = 1 + (-2)(0.05) = 0.9$ in the z -plane. The characteristic root of the equivalent continuous-time system is $\lambda^* = 1/T \times \ln z_1 = 1/0.05 \times \ln 0.9 = -2.107$ in the s -plane.

8.2.2 CHARACTERISTIC ROOT ERRORS

The fractional error in characteristic root incurred using numerical integration for digital simulation of a first-order continuous-time system with characteristic root λ is defined as (Howe 1986)

$$e_\lambda = \frac{\lambda^* - \lambda}{\lambda} \quad (8.22)$$

For an underdamped second-order system with complex poles $\lambda_{1,2} = -\zeta\omega_n \pm j\omega_d$ where ζ and ω_n are the damping ratio and natural frequency, respectively, and $\omega_d = \sqrt{1 - \zeta^2}\omega_n$ is the damped natural frequency, the characteristic root errors are

$$e_\zeta = \zeta^* - \zeta, \quad e_{\omega_n} = \frac{\omega_n^* - \omega_n}{\omega_n}, \quad e_{\omega_d} = \frac{\omega_d^* - \omega_d}{\omega_d} \quad (8.23)$$

$\zeta^* \omega_n^*$ and ω_d^* are the damping ratio, natural frequency, and damped natural frequency of the equivalent continuous-time second-order system, respectively. The characteristic roots of the equivalent continuous-time system are

$$\lambda_{1,2}^* = -\zeta^* \omega_n^* \pm j\omega_d^* = -\zeta^* \omega_n^* \pm j\sqrt{1 - (\zeta^*)^2} \omega_n^* \quad (8.24)$$

High-order linear continuous-time systems can be represented as the sum of first-and second-order continuous-time systems. Hence, the characteristic root errors introduced in Equations 8.22 and 8.23 are sufficient to analyze transient response dynamic errors of higher-order systems comprising first- and second-order subsystems.

EXAMPLE 8.1

The first-order system in Equation 8.1 is simulated using trapezoidal integration.

- a. Find an expression for e_λ , the fractional error in characteristic root.
- b. Find an asymptotic formula for e_λ valid for $|\lambda T| \ll 1$.
- c. Over what range of values for λT is the asymptotic formula for e_λ accurate?
- a. The difference equation for trapezoidal integration is based on

$$x_A(n+1) = x_A(n) + \frac{T}{2} \{f[x_A(n), u(n)] + f[x_A(n+1), u(n+1)]\} \quad (8.25)$$

where $f[x_A(n), u(n)]$ and $f[x_A(n+1), u(n+1)]$ refer to the derivative function in Equation 8.1. Z-transforming the difference equation and then solving for the ratio $X(z)/U(z)$ results in the z-domain transfer function

$$H(z) = \frac{X(z)}{Y(z)} = T \left[\frac{z+1}{(2-\lambda T)z - (2+\lambda T)} \right] \quad (8.26)$$

The z-plane pole is

$$z_1 = \frac{2 + \lambda T}{2 - \lambda T} = \frac{1 + \lambda T/2}{1 - \lambda T/2} \quad (8.27)$$

From Equation 8.15, the characteristic root of the equivalent continuous-time system is

$$\lambda^* = \frac{1}{T} \ln z_1 = \frac{1}{T} \ln \left(\frac{1 + \lambda T/2}{1 - \lambda T/2} \right) \quad (8.28)$$

and the fractional error in characteristic root is

$$e_\lambda = \frac{\lambda^*}{\lambda} - 1 = \frac{1}{\lambda T} \ln \left(\frac{1 + \lambda T/2}{1 - \lambda T/2} \right) - 1 \quad (8.29)$$

- b. Equation 8.29 is expressed in the form

$$e_\lambda = \frac{1}{\lambda T} \left[\ln \left(1 + \frac{\lambda T}{2} \right) - \ln \left(1 - \frac{\lambda T}{2} \right) \right] - 1 \quad (8.30)$$

The asymptotic formula for e_λ is obtained by truncating the Taylor Series expansion

$$\ln(1+a) = a - \frac{a^2}{2} + \frac{a^3}{3} - \frac{a^4}{4} + \dots \quad (8.31)$$

after the cubic term where $a = \lambda T/2$ and $a = -\lambda T/2$ in Equation 8.30. After simplification, the result is

$$e_\lambda \approx \frac{1}{12}(\lambda T)^2, \quad |\lambda T| \ll 1 \quad (8.32)$$

- c. A plot of the exact and asymptotic formulas for e_λ is shown in Figure 8.4. From the graph, it appears that the exact and asymptotic formulas for e_λ are nearly identical for $-0.5 \leq \lambda T < 0$.

The first-order continuous-time system in Equation 8.1 is asymptotically stable provided $\lambda < 0$. The graphs of exact and asymptotic error in Figure 8.4 are for $\lambda T < 0$; hence, they apply strictly to asymptotically stable, first-order systems. Equations 8.29 and 8.32 are not valid for $\lambda T = 0$, that is, when the continuous-time system reduces to a marginally stable integrator with characteristic root $\lambda = 0$.

Consider the use of trapezoidal integration with step size T to simulate the autonomous first-order system $\dot{x} = f(x) = \lambda x$ with initial condition $x(0)$. The discrete-time signal $x_A(n)$ satisfies the difference equation

$$x_A(n+1) = \left(\frac{1 + \lambda T/2}{1 - \lambda T/2} \right) x_A(n), \quad n = 0, 1, 2, 3, \dots \quad (8.33)$$

with solution given by

$$x_A(n) = \left(\frac{1 + \lambda T/2}{1 - \lambda T/2} \right)^n x(0), \quad n = 0, 1, 2, 3, \dots \quad (8.34)$$

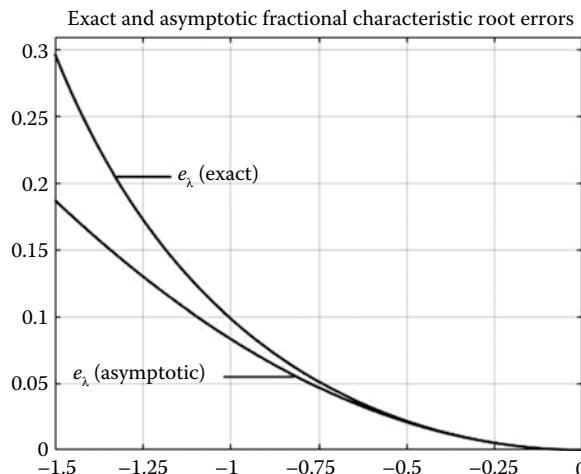


FIGURE 8.4 Exact and asymptotic fractional characteristic root errors for trapezoidal integration (with step size T) of first-order system $\dot{x} = \lambda x + u$.

TABLE 8.1
Effect of Parameter λT on Equivalent Characteristic Root and Fractional Characteristic Root Errors with Trapezoidal Integration

λ	T	λT	λ^*	e_λ (Exact)	e_λ (Asymptotic)
-0.5	0.015	-0.0075	-0.500002	4.68754×10^{-6}	4.68750×10^{-6}
-0.5	0.15	-0.075	-0.500234	4.69146×10^{-4}	4.68750×10^{-4}
-0.5	1.5	-0.75	-0.526538	5.12764×10^{-2}	4.68750×10^{-2}
-0.5	2.5	-1.25	-0.586535	1.73070×10^{-1}	1.30208×10^{-1}

Table 8.1 summarizes the results for a first-order system with characteristic root $\lambda = -0.5$ simulated using trapezoidal integration with four different step sizes. The results are consistent with the graphs in Figure 8.4.

Several different responses are shown in Figure 8.5. The top two plots show the response of the continuous-time system and the discrete-time response corresponding to the top two rows in Table 8.1. Due to the close agreement between λ and λ^* , the response of the equivalent continuous-time system is indistinguishable from the response of the actual system. Additionally, the discrete-time output (not all points shown) is in close agreement with the continuous-time response at times 0, T , $2T$, ...

In the last two cases ($\lambda T = -0.75$ and $\lambda T = -1.25$), the difference between λ and λ^* is significant, and the response of the equivalent continuous-time system is noticeably different from the actual system response, particularly for the case where $\lambda T = -1.25$. The simulated (discrete-time) response is off as well.

Characteristic root errors resulting from simulation of second-order systems using specific numerical integrators are obtained in a straightforward manner. To illustrate, consider an under-damped second-order continuous-time system with characteristic roots $\lambda_{1,2} = -\zeta\omega_n \pm j\sqrt{1-\zeta^2}\omega_n$. Similar to the approach used in Equation 8.4, replacing the Laplace variable s in the continuous-system transfer function with the reciprocal of the z-domain transfer function for Euler integration leads to the z-domain transfer function of the discrete-time system. This gives

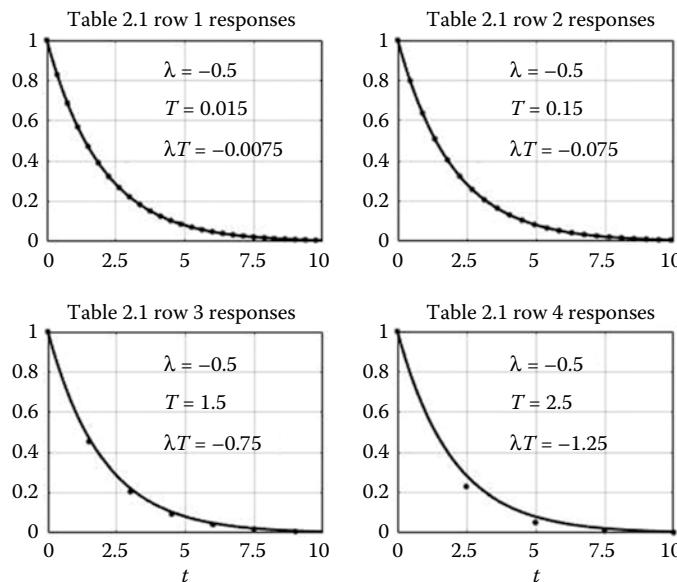


FIGURE 8.5 Responses of first-order continuous-time, discrete-time, and equivalent continuous-time systems for conditions in Table 8.1.

$$H(z) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \Big|_{s \leftarrow \frac{z-1}{T}} \quad (8.35)$$

$$= \frac{K(\omega_n T)^2}{z^2 - 2(1 - \zeta\omega_n T)z + 1 - 2\zeta\omega_n T + (\omega_n T)^2} \quad (8.36)$$

Setting the denominator to zero and solving for the poles of $H(z)$ give

$$z_{1,2} = 1 - \zeta\omega_n \pm j\sqrt{(1 - \zeta^2)\omega_n T} \quad (8.37)$$

From Equation 8.15, the characteristic roots of the equivalent continuous-time system are

$$s_{1,2}^* = \frac{1}{T} \ln z_{1,2} \quad (8.38)$$

Finding an expression for s_1^* is easier when the corresponding z -plane pole z_1 is written in polar form.

$$s_1^* = \frac{1}{T} \ln(R e^{j\theta}) = \frac{1}{T} \ln R + j \frac{\theta}{T} \quad (8.39)$$

where R and θ are obtained from Equation 8.37 (after simplification) as

$$R = \sqrt{1 - 2\zeta\omega_n T + (\omega_n T)^2} \quad (8.40)$$

$$\theta = \tan^{-1} \left(\frac{\sqrt{1 - \zeta^2}\omega_n T}{1 - \zeta\omega_n T} \right) \quad (8.41)$$

Substituting Equations 8.40 and 8.41 into Equation 8.39 gives $s_1^* = a^* + jb^*$ where the real and imaginary components a^* and b^* are given by

$$a^* = \frac{1}{T} \ln \left(\sqrt{1 - 2\zeta\omega_n T + (\omega_n T)^2} \right) \quad (8.42)$$

$$b^* = \frac{1}{T} \tan^{-1} \left(\frac{\sqrt{1 - \zeta^2}\omega_n T}{1 - \zeta\omega_n T} \right) \quad (8.43)$$

The continuous-time pole s_1 , the z -plane pole z_1 , and the equivalent continuous-time system pole s_1^* are shown in [Figure 8.6](#).

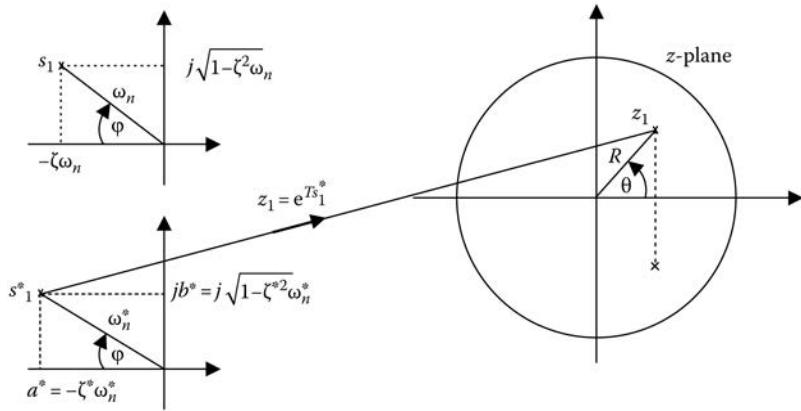


FIGURE 8.6 Relationship between second-order continuous-time system, discrete-time, and equivalent continuous-time system complex pole.

From Figure 8.6, it follows that

$$\omega_n^* = (a^{*2} + b^{*2})^{1/2} = \frac{1}{T} \left\{ \left[\ln \left(\sqrt{1 - 2\zeta\omega_n T + (\omega_n T)^2} \right) \right]^2 + \left[\tan^{-1} \left(\frac{\sqrt{1 - \zeta^2} \omega_n T}{1 - \zeta\omega_n T} \right) \right]^2 \right\} \quad (8.44)$$

$$\zeta^* = \cos \varphi = \frac{-a^*}{\omega_n^*} = \frac{-\ln \left(\sqrt{1 - 2\zeta\omega_n T + (\omega_n T)^2} \right)}{\omega_n^* T} \quad (8.45)$$

Asymptotic formulas for ω_n^* and ζ^* are given in (Howe 1986) as

$$\omega_n^* \approx \left[1 + \frac{\zeta\omega_n T}{2} \right] \omega_n, \quad \omega_n T \ll 1 \quad (8.46)$$

$$\zeta^* \approx \zeta - \left(\frac{1 - \zeta^2}{2} \right) \omega_n T, \quad \omega_n T \ll 1 \quad (8.47)$$

Exact and approximate (asymptotic) expressions for the fractional error in natural frequency of the equivalent continuous-time system

$$e_{\omega_n} = \left(\frac{\omega_n^*}{\omega_n} \right) - 1 \quad (8.48)$$

are obtained from Equation 8.44 for the exact result and Equation 8.46 for the asymptotic one. Figure 8.7 shows exact and asymptotic fractional errors for several second-order continuous-time system damping ratios using explicit Euler integration.

Substituting Equation 8.46 into Equation 8.48 results in the asymptotic fractional error as a linear function of $\omega_n T$, that is,

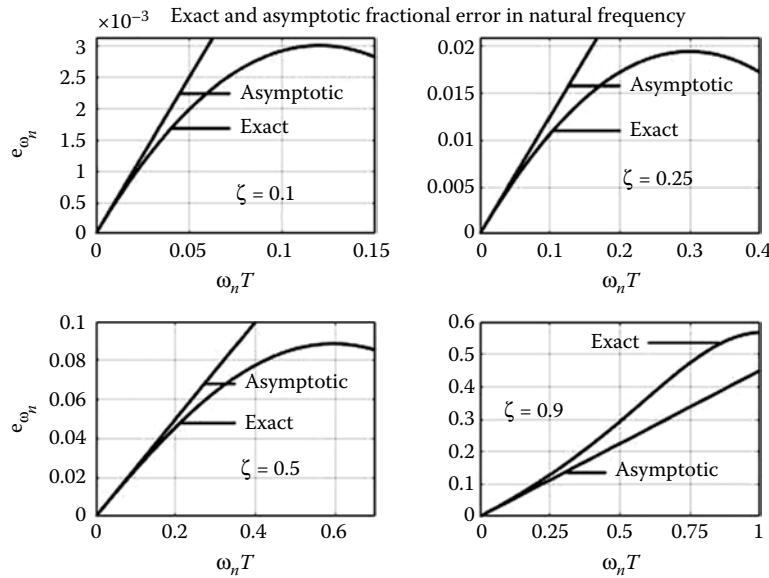


FIGURE 8.7 Exact and asymptotic fractional errors in natural frequency with explicit Euler integration.

$$e_{\omega_n} \approx 0.5\zeta(\omega_n T), \quad \omega_n T \ll 1 \quad (8.49)$$

From Equation 8.45, the damping ratio error e_ζ is expressible as

$$e_\zeta = \zeta^* - \zeta = \frac{-\ln(\sqrt{1-2\zeta\omega_n T + (\omega_n T)^2})}{\omega_n^* T} - \zeta \quad (8.50)$$

where ω_n^* is given in Equation 8.44. From Equation 8.47, the asymptotic approximation for e_ζ is

$$e_\zeta \approx 0.5(\zeta^2 - 1)\omega_n T, \quad \omega_n T \ll 1 \quad (8.51)$$

The asymptotic expressions for e_{ω_n} in Equation 8.49 and e_ζ in Equation 8.51 are of order $O(\omega_n T)$ when using Euler integration to simulate an underdamped second-order system.

A plot of the exact and asymptotic formulas for the equivalent system damping ratio ζ^* as a function of $\omega_n T$ when $\zeta = 0.1$ is shown in the top half of Figure 8.8. Agreement between the two plots is excellent over the interval $0 \leq \omega_n T \leq 0.5$.

The equivalent continuous-time system is marginally stable when its two characteristic roots (transfer function poles) are purely imaginary, that is, $\zeta^* = 0$ (see Figure 8.6). From Equation 8.51 with $\zeta = 0.1$ and $\zeta^* = 0$, the dimensionless parameter $\omega_n T$ is computed as

$$\begin{aligned} \zeta^* - \zeta &= 0 - 0.1 \approx 0.5[(0.1)^2 - 1]\omega_n T \\ \Rightarrow \omega_n T &= \frac{-0.1}{0.5(-0.99)} = 0.202 \end{aligned} \quad (8.52)$$

The damping ratio ζ^* of the equivalent continuous-time system is negative whenever $\omega_n T > 0.202$. The implication of $\zeta^* < 0$ is obvious from Figure 8.6, namely, the characteristic roots

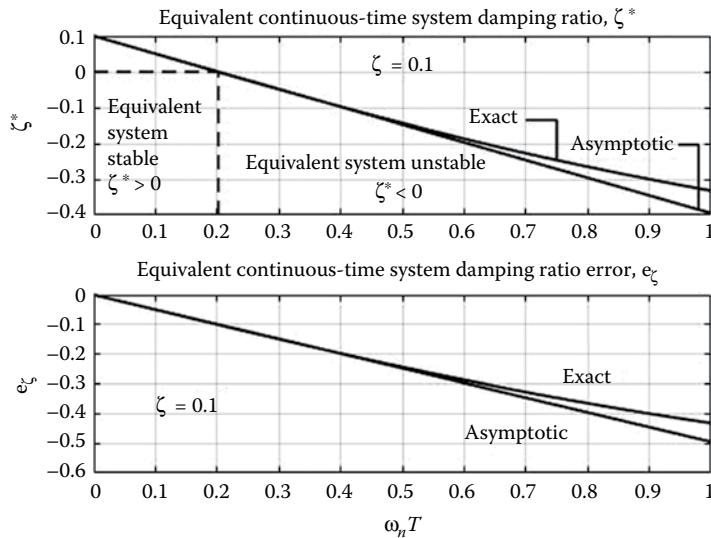


FIGURE 8.8 Equivalent system damping ratio (ζ^*) and error (e_ζ) vs. $\omega_n T$.

are located in the right half of the complex plane, and the equivalent system is unstable despite the fact that the actual continuous-time system is asymptotically stable with positive damping ratio $\zeta = 0.1$. The lower half of Figure 8.8 shows plots of the error $e_\zeta = \zeta^* - \zeta = \zeta^* - 0.1$ vs. $\omega_n T$ based on the exact and asymptotic formulas in Equations 8.50 and 8.51.

Figure 8.9 points out a serious shortcoming of using explicit Euler integration to simulate the response of a marginally stable ($\zeta = 0$) second-order system. The equivalent continuous-time system is unstable because $\zeta^* < 0$ regardless of how small $\omega_n T$ is chosen. The natural modes of the equivalent continuous-time system are oscillatory with increasing amplitude. The discrete-time system based on the use of explicit Euler integration is likewise unstable with a pair of complex poles outside the Unit Circle. This problem can be fixed by using trapezoidal integration instead of Euler integration (see Exercise 8.2).

Figures 8.7 through 8.9 are generated in M-file “Ch8_Fig2_7throughFig2_9.m.”

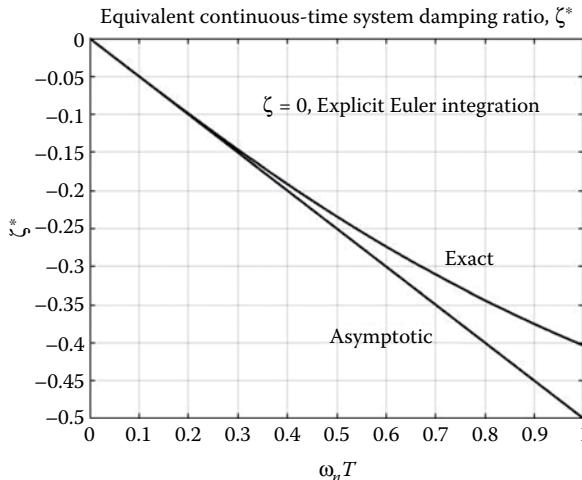


FIGURE 8.9 Equivalent system damping ratio using explicit Euler integration.

EXAMPLE 8.2

A second-order system with damping ratio $\zeta = 0.1$, natural frequency $\omega_n = 50$ rad/s, and steady-state gain $K = 1$ is initially in equilibrium. A unit step input is applied at $t = 0$. The step response is simulated using explicit Euler integration with step size T .

- Find the step response $x(t)$, $t \geq 0$.
- Find the equivalent system natural frequency ω_n^* and damping ratio ζ^* for $T = 0.001, 0.002, 0.004, 0.005$ s.
- Plot the continuous-time system response $x(t)$ and the discrete-time system response $x_A(n)$, $n = 0, 1, 2, \dots$ corresponding to the values of T in part (b).
- The unit step response of an underdamped second-order system is (see [Chapter 2](#))

$$x(t) = K \left[1 - e^{-\zeta \omega_n t} \left(\cos \omega_d t + \frac{\zeta \omega_n}{\omega_d} \sin \omega_d t \right) \right], \quad t \geq 0 \quad (8.53)$$

Substituting the given values for the system parameters, ζ , ω_n , and K and evaluating the damped natural frequency $\omega_d = \sqrt{1 - \zeta^2} \omega_n$ give

$$x(t) = 1 - e^{-5t} \left[\cos(50\sqrt{0.99}t) + \frac{1}{10\sqrt{0.99}} \sin(50\sqrt{0.99}t) \right], \quad t \geq 0 \quad (8.54)$$

- Using Equations 8.44 and 8.46 for the exact and asymptotic equivalent system natural frequencies along with Equations 8.45 and 8.47 for the exact and asymptotic equivalent system damping ratios, the results are tabulated in [Table 8.2](#). The damped natural frequency of the continuous-time system ω_d and the exact and asymptotic damped natural frequency approximation of the equivalent system are also shown.

Note that the equivalent continuous-time (as well as the discrete-time) system is on the verge of instability at $\omega_n T = 0.2$ in agreement with the top graph shown in [Figure 8.8](#).

- The continuous-time response is plotted on the same graph as the simulated response for the four distinct values of T in [Figure 8.10](#). Every third point of the discrete-time response is plotted in the top left graph. Every point is shown in the remaining plots.

The damped natural frequency of the four simulated step responses corresponding to $\omega_n T = 0.05, 0.01, 0.2, 0.25$ appears to be in close agreement with the continuous-time system response. However, even the discrete-time response in the top left graph where the integration step size is $T = 0.001$ s deviates considerably from the continuous-time response in the neighborhood of the peaks and low points. The oscillatory discrete-time response in the lower left graph in [Figure 8.10](#)

TABLE 8.2
Comparison of Actual System and Equivalent System Parameters

$\omega_n T$	ω_n	$\omega_n^* \text{ Exact}$	$\omega_n^* \text{ Approximate}$	ζ	$\zeta^* \text{ Exact}$	$\zeta^* \text{ Approximate}$	ω_d	$\omega_d^* \text{ Exact}$	$\omega_d^* \text{ Approximate}$
0.05	50	50.099	50.125	0.1	0.0751	0.0753	49.749	49.958	49.983
0.10	50	50.147	50.250	0.1	0.0501	0.0505	49.749	50.084	50.186
0.20	50	50.084	50.500	0.1	0.0000	0.0010	49.749	50.084	50.500
0.25	50	49.975	50.625	0.1	-0.0249	-0.0237	49.749	49.959	50.611

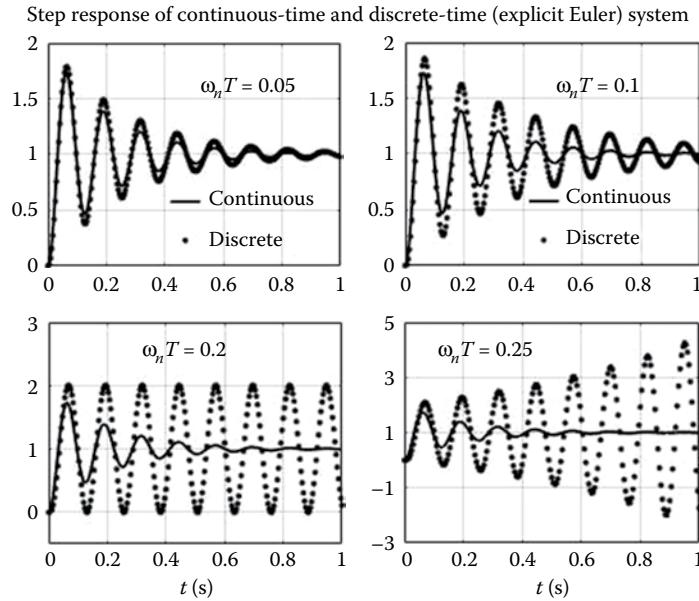


FIGURE 8.10 Continuous-time and discrete-time unit step responses of second-order system ($\zeta = 0.1$, $\omega_n = 50$ rad/s) using explicit Euler integration.

is consistent with [Figure 8.8](#), which shows the equivalent continuous-time system damping ratio is zero when $\omega_n T \approx 0.2$.

It is clear from this example that the use of explicit Euler integration to approximate the dynamics of an underdamped, stable, second-order system ($0 < \zeta < 1$) may result in an equivalent continuous-time system that is asymptotically stable ($0 < \zeta^* < 1$), marginally stable ($\zeta^* = 0$), or unstable ($\zeta^* < 0$). From [Figure 8.6](#), the equivalent continuous-time system is marginally stable when $a^* = 0$. Setting the argument of the natural log term in the expression for a^* in [Equation 8.42](#) to 1 and solving for $\omega_n T$ gives

$$(\omega_n T)_{\max} = 2\zeta \Rightarrow T_{\max} = \frac{2\zeta}{\omega_n} \quad (0 < \zeta < 1) \quad (8.55)$$

where $(\omega_n T)_{\max}$ and T_{\max} are the values of $(\omega_n T)$ and T , which result in marginally stable, discrete-time, and equivalent continuous-time systems. A plot of $(\omega_n T)_{\max} = 2\zeta$ is shown in [Figure 8.11](#) along with $\omega_n T$ ranging from 0.01 up to $(\omega_n T)_{\max}$ when $\zeta = 0.707$.

Consider the second-order system

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = K\omega_n^2u \quad (8.56)$$

with parameters $\zeta = 0.707$, $\omega_n = 10$ rad/s, and $K = 1$. Differentiating the unit step response in [Equation 8.53](#) gives the unit impulse response (Ogata 1998). Alternatively, the impulse response can be obtained by inverse Laplace transformation of the system transfer function $H(s) = X(s)/U(s)$. Either way, the result is

$$h(t) = K \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_d t, \quad \omega_d = \sqrt{1-\zeta^2} \omega_n \quad (8.57)$$

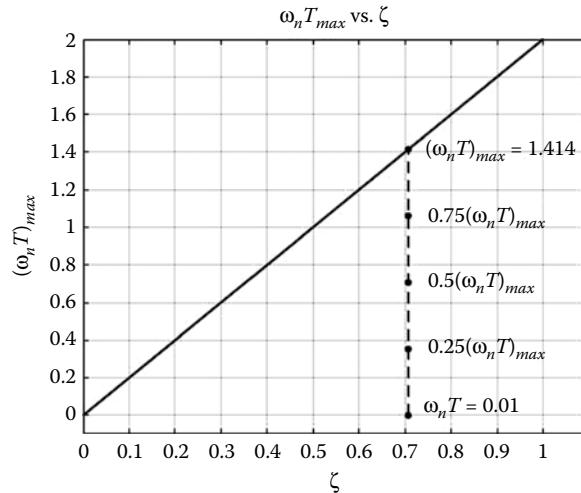


FIGURE 8.11 Plot of $(\omega_n T)$ vs. ζ resulting in marginally stable, second-order equivalent continuous-time system using explicit Euler integration.

Suppose we attempt to simulate the impulse response of the system in Equation 8.56 using explicit Euler integration. The difference equation for explicit Euler integration of the second-order system in Equation 8.56 was developed in Section 4.7 and is repeated in Equation 8.58.

$$x_{k+2} - 2(1 - \zeta \omega_n T)x_{k+1} + [1 - 2\zeta \omega_n T + (\omega_n T)^2]x_k = K(\omega_n T)^2 u_k, \quad k = -1, 0, 1, 2, \dots \quad (8.58)$$

The unit impulse response of the second-order system in Equation 8.56 is identical to the response of the unforced system with initial conditions $x(0) = 0$, $\dot{x}(0) = \omega_n^2$ (see Exercise 8.6). Therefore, the impulse response can be simulated by solving the difference equation in Equation 8.58 with $u_k = -1, 0, 1, 2, \dots$ along with the appropriate initial conditions, namely, $x(0) = 0$ and $x(-1) = -\omega_n^2 T$. The simulated impulse responses for the values of $\omega_n T$ in Figure 8.11 are shown in Figures 8.12 and 8.13. Not all the data points for the discrete-time response when $\omega_n T = 0.01$ are shown.

Figure 8.12 illustrates the necessity of choosing the time step to achieve an accurate transient response. Indeed, all four simulated responses in Figure 8.12 are stable and converge to the correct steady state, but only one is reasonably accurate. Figure 8.13 represents the case where the discrete-time system (and the equivalent continuous-time system) are marginally stable with oscillatory natural modes.

It may have occurred to you that the impulse response of the second-order system in Equation 8.56 could be simulated by finding the impulse response h_k , $k = 0, 1, 2, \dots$ of the discrete-time system described by Equation 8.58, either analytically or by recursive solution of the difference equation with $u_k = \delta_k = 1$, $k = 0, 1, 2, \dots$. Think twice before doing so because $h_k \neq h(t)|_{t=kT}$.

8.2.3 TRANSFER FUNCTION ERRORS

A second class of dynamic error involves the frequency response functions of the continuous-time system and the discrete-time system used to simulate it. The fractional error in the (discrete-time system) transfer function is

$$e_H = \frac{H(e^{j\omega T}) - H(j\omega)}{H(j\omega)} \quad (8.59)$$

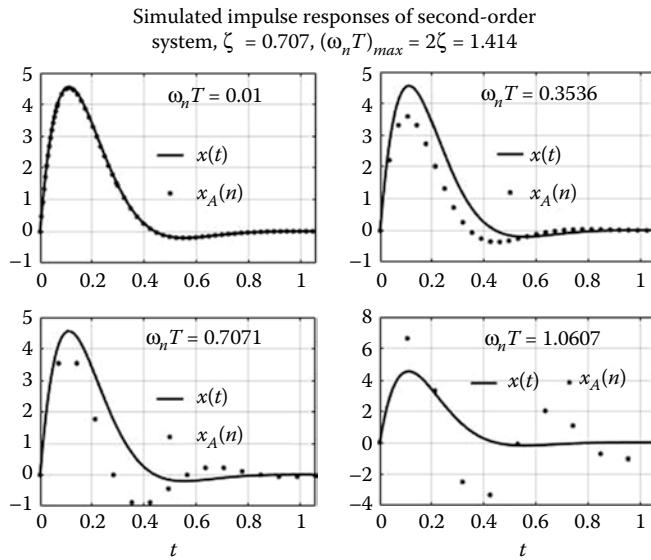


FIGURE 8.12 Continuous-time and simulated second-order system impulse responses using explicit Euler integration with different values for parameter $\omega_n T$.

where it is important to remember that

$$H(j\omega) = H(s)|_{s \leftarrow j\omega}, \quad H(z) = H(s)|_{s \leftarrow 1/H_I(z)}, \quad H(e^{j\omega T}) = H(z)|_{z \leftarrow e^{j\omega T}} \quad (8.60)$$

The fractional error in transfer function is a complex-valued, frequency-dependent function, which can be expressed in terms of a real and imaginary component, that is,

$$e_H = \frac{H(e^{j\omega T})}{H(j\omega)} - 1 = e_M + j e_A \quad (8.61)$$

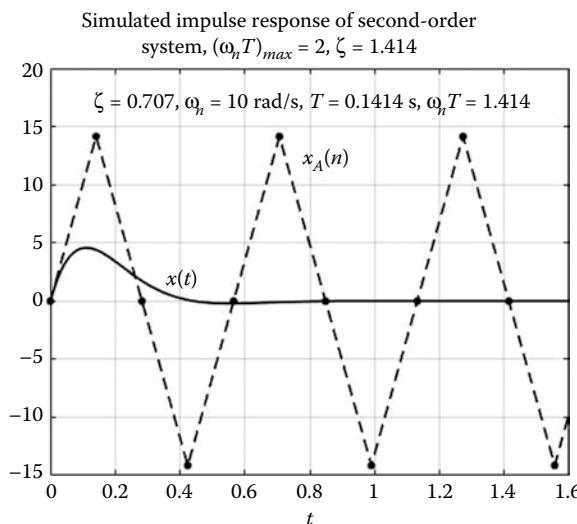


FIGURE 8.13 Simulated impulse response of second-order system with marginally stable Euler integrator.

In polar form, the frequency response functions are expressed as

$$H(j\omega) = |H(j\omega)|e^{j\phi}, \quad \text{where } \phi = \text{Arg}[H(j\omega)] \quad (8.62)$$

$$H(e^{j\omega T}) = |H(e^{j\omega T})|e^{j\phi^*}, \quad \text{where } \phi^* = \text{Arg}[H(e^{j\omega T})] \quad (8.63)$$

Substitution of Equations 8.62 and 8.63 into Equation 8.61 yields

$$e_H = \frac{|H(e^{j\omega T})|e^{j\phi^*}}{|H(j\omega)|e^{j\phi}} - 1 \quad (8.64)$$

$$= \frac{|H(e^{j\omega T})|}{|H(j\omega)|} e^{j(\phi^* - \phi)} - 1 \quad (8.65)$$

Approximating $e^{j(\phi^* - \phi)}$ in a first-order Taylor Series expansion, that is,

$$e^{j(\phi^* - \phi)} \approx 1 + j(\phi^* - \phi) \quad (8.66)$$

$$\Rightarrow e_H \approx \frac{|H(e^{j\omega T})|}{|H(j\omega)|} [1 + j(\phi^* - \phi) - 1] \quad (8.67)$$

$$\Rightarrow e_H \approx \frac{|H(e^{j\omega T})|}{|H(j\omega)|} + \frac{|H(e^{j\omega T})|}{|H(j\omega)|} j(\phi^* - \phi) - 1 \quad (8.68)$$

When the simulation is reasonably accurate, $H(e^{j\omega T}) \approx H(j\omega)$ over a range of frequencies and the term $(|H(e^{j\omega T})|)/(|H(j\omega)|)j(\phi^* - \phi)$ can be approximated by $j(\phi^* - \phi)$ (Howe 1986).

The final expression for e_H is therefore

$$e_H \approx \frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 + j(\phi^* - \phi) \quad (8.69)$$

Comparison of Equations 8.61 and 8.69 reveals

$$e_M = \text{Re}(e_H) = \text{Re} \left\{ \left[\frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 \right] \right\} \approx \frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 \quad (8.70)$$

$$e_A = \text{Im}(e_H) = \text{Im} \left\{ \left[\frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 \right] \right\} \approx \phi^* - \phi \quad (8.71)$$

From Equation 8.70, e_M , the real part of e_H (the fractional error in discrete-time transfer function), is approximately equal to the fractional error in the discrete-time transfer function gain. Furthermore, e_A , the imaginary part of e_H , is approximately equal to the phase error of $H(e^{j\omega T})$.

Consider the case of a continuous-time integrator approximated by explicit Euler integration with step size T . Setting $\lambda = 0$ in Equation 8.4 or referring to Equation 4.465, the z -domain transfer function is

$$H(z) = \frac{T}{z-1} \quad (8.72)$$

Substituting expressions for $H(e^{j\omega T})$ and $H(j\omega)$ in the definition of e_H gives

$$e_H = \frac{T/(e^{j\omega T}-1)}{1/j\omega} - 1 \quad (8.73)$$

$$= \frac{j\omega T - e^{j\omega T} + 1}{e^{j\omega T} - 1} \quad (8.74)$$

$$= \frac{1 - \cos \omega T + j(\omega T - \sin \omega T)}{\cos \omega T - 1 + j \sin \omega T} \quad (8.75)$$

Rationalizing Equation 8.75, that is, multiplying numerator and denominator by $\cos \omega T - 1 - j \sin \omega T$, and simplifying lead to

$$e_H = e_M + j e_A = \frac{\omega T \sin \omega T}{2(1 - \cos \omega T)} - 1 + j \left(\frac{-\omega T}{2} \right) \quad (8.76)$$

From Equation 8.70, an approximation for the fractional gain error in $H(e^{j\omega T})$ is

$$\frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 \approx e_M = \frac{\omega T \sin \omega T}{2(1 - \cos \omega T)} - 1 \quad (8.77)$$

and from Equation 8.71, the approximation for the phase error in $H(e^{j\omega T})$ is

$$\text{Arg}[H(e^{j\omega T})] - \text{Arg}[H(j\omega)] \approx e_A = -\frac{\omega T}{2} \quad (8.78)$$

Exact expressions for the fractional gain error and phase error for the explicit Euler integrator are (see Exercise 8.7)

$$\text{Fractional gain error} = \frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 = \frac{\omega T}{[2(1 - \cos \omega T)]^{1/2}} - 1 \quad (8.79)$$

$$\text{Phase error} = \text{Arg}[H(e^{j\omega T})] - \text{Arg}[H(j\omega)] = -\tan^{-1} \left(\frac{\sin \omega T}{\cos \omega T - 1} \right) - \left(-\frac{\pi}{2} \right) \quad (8.80)$$

[Figure 8.14](#) contains graphs of the exact and approximate expressions for the fractional error in gain for $0 \leq \omega T \leq 1$ rad. Note that e_M is a good approximation to the fractional gain error in $H(e^{j\omega T})$

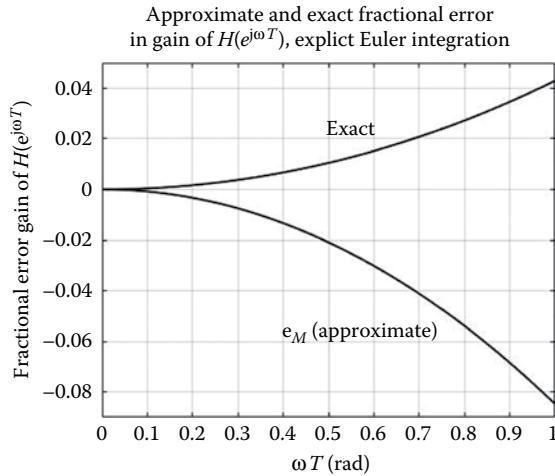


FIGURE 8.14 Approximate and exact fractional error in discrete-time transfer function gain using explicit Euler integration.

provided $\omega T \ll 1$. An asymptotic approximation for e_M , which holds for $\omega T \ll 1$, can be obtained by replacing $\sin \omega T$ and $\cos \omega T$ in Equation 8.77 with the first two nonzero terms in the Taylor Series expansions,

$$\sin \omega T \approx \omega T - \frac{(\omega T)^3}{3!}, \quad \cos \omega T \approx 1 - \frac{(\omega T)^2}{2!} \quad (8.81)$$

eventually leading to $e_M \approx 0$, $\omega T \ll 1$ confirmed by the graph of e_M in Figure 8.14.

The phase angle plots for the continuous-time integrator and explicit Euler integrator are shown in Figure 8.15. The top graph shows the constant phase angle $-\pi/2$ rad for the continuous-time

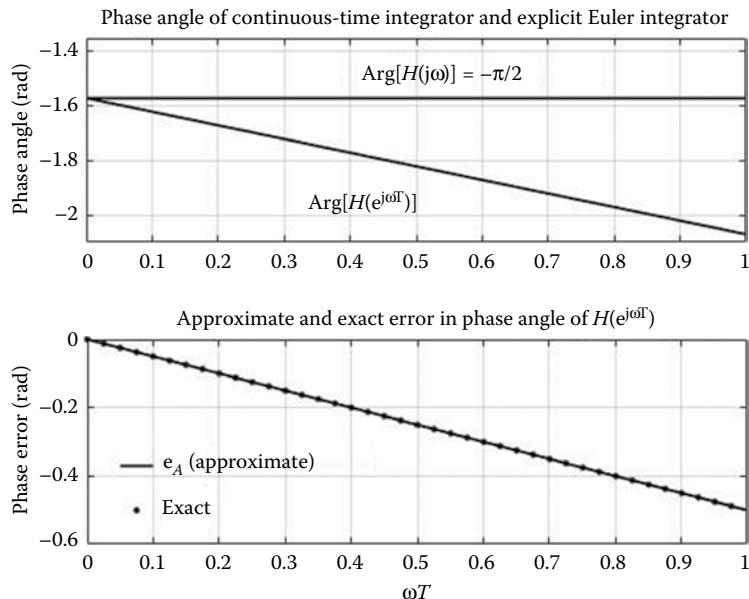


FIGURE 8.15 Phase angle plots for continuous-time and explicit Euler integrator.

integrator along with the phase angle of the discrete-time transfer function given in Equation 8.72. The lower graph shows e_A in Equation 8.78 and equally spaced points computed from the exact expression for the phase error in Equation 8.80. The linear approximation e_A is virtually identical to the exact expression for the phase error.

An asymptotic expression for $H(e^{j\omega T})$ can be derived starting with Equation 8.72.

$$H(e^{j\omega T}) = \frac{T}{e^{j\omega T} - 1} = \frac{T}{[1 + j\omega T + ((j\omega T)^2/2!) + ((j\omega T)^3/3!) + \dots] - 1} \quad (8.82)$$

Truncating the power series for $e^{j\omega T}$ after the quadratic term gives

$$H(e^{j\omega T}) \approx \frac{T}{j\omega T + (j\omega T)^2/2}, \quad \omega T \ll 1 \quad (8.83)$$

$$\approx \frac{1}{j\omega} \frac{T}{(1 + j\omega T/2)}, \quad \omega T \ll 1 \quad (8.84)$$

The frequency response function of the continuous-time integrator is $H(j\omega) = 1/j\omega$. The second term

$$\frac{1}{1 + j\omega T/2} = \frac{1}{[1 + (\omega T/2)^2]^{1/2} e^{j\tan^{-1}(\omega T/2)}} \quad (8.85)$$

$$\approx e^{-j\omega T/2}, \quad \omega T \ll 1 \quad (8.86)$$

$$\Rightarrow H(e^{j\omega T}) \approx H(j\omega) e^{-j\omega T/2}, \quad \omega T \ll 1 \quad (8.87)$$

implying that the asymptotic behavior ($\omega T \ll 1$) of the explicit Euler integrator is that of a pure continuous-time integrator with an additional delay of $-\omega T/2$ rad.

To illustrate Equation 8.87, a sine wave $u(t)$ at a frequency of ω rad/s is input to an integrator shown in Figure 8.16. The signal $u(t)$ is sampled every T s, and the resulting discrete-time signal is input to an explicit Euler integrator updating at 1/T Hz.

The top half of Figure 8.17 shows the sinusoidal input $u(t) = \sin 2t$, the explicit Euler output $x_A(n)$, $n = 0, 5, 10, \dots$ when the step size $T = 0.025$ s, and the continuous-time output $x(t)$. The parameter $\omega T = 2(0.025) = 0.05$ rad is small enough for the asymptotic formula in Equation 8.87 to accurately predict the characteristics of the discrete-time output $x_A(n)$. According to Equation 8.87, the steady-state amplitudes of $x_A(n)$ and $x(t)$ are equal for all input frequencies provided $\omega T \ll 1$. The amplitude is

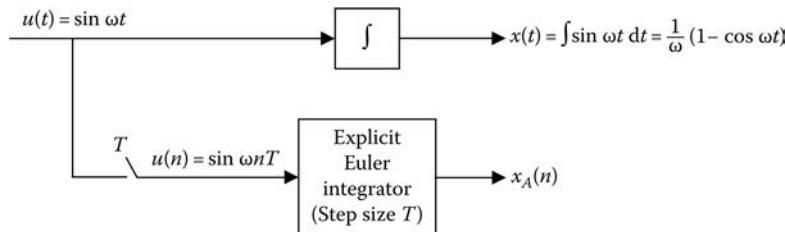


FIGURE 8.16 Continuous- and discrete-time integration of a sinusoidal input.

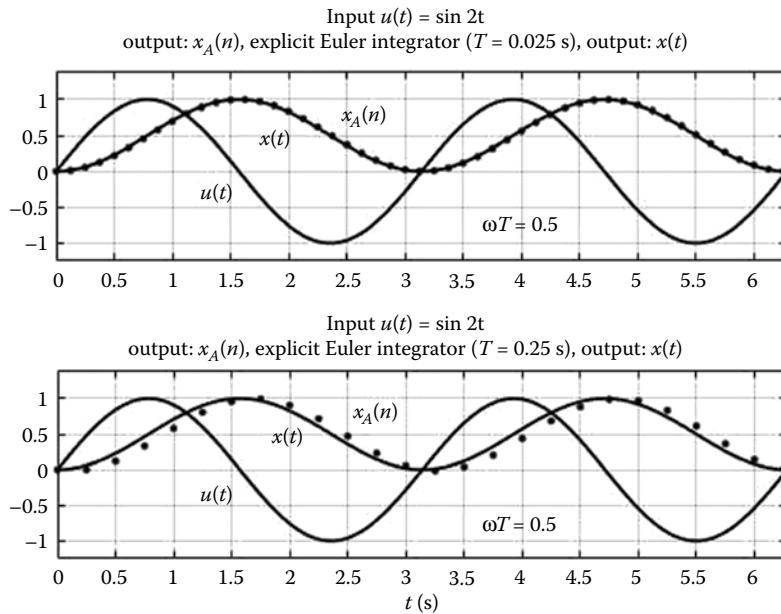


FIGURE 8.17 Explicit Euler and continuous-time integrator outputs ($\omega T = 0.05$ rad).

$$|x_A(n)| = |x(t)| = |H(j\omega)| \cdot |u(t)| = \left| \frac{1}{j\omega} \right| \cdot 1 = \frac{1}{\omega} = \frac{1}{2} \quad (8.88)$$

easily verified by looking at the plots of $x_A(n)$ and $x(t)$ in the top half of Figure 8.17.

The asymptotic approximation for $H(e^{j\omega T})$ in Equation 8.87 also predicts a delay of $\omega T/2$ rad in $x_A(n)$ relative to the continuous-time output $x(t)$. The time delay can be estimated by zooming in on the responses in the top half of Figure 8.17.

Alternatively, the time delay can be determined by finding the time between occurrences of equal values of $x_A(n)$ and $x(t)$. For example, at $t = 4$ s, the discrete-time variable $n = 4/0.025 = 160$ and from M-file “Ch8_Fig8_17.m,” $x_A(160) = 0.5603$. Setting $x(t_0) = x_A(160) = 0.5603$ and solving for t_0 .

$$x(t_0) = \frac{1}{\omega} (1 - \cos \omega t_0) = x_A(160) = 0.5603 \quad (8.89)$$

$$t_0 = \frac{1}{2} \cos^{-1}[1 - 2(0.5603)] + \frac{2\pi}{\omega} = 3.9874 \text{ s} \quad (8.90)$$

The simulated response $x_A(n)$ is lagging the output $x(t)$ by $4 - 3.9874 = 0.0126$ s, in close agreement with the predicted value of $T/2 = 0.0125$ s.

The lower half of Figure 8.17 illustrates the case where $\omega = 2$ rad/s, $T = 0.25$ s, and the asymptotic approximation in Equation 8.87 based on $\omega T \ll 1$ no longer applies. From Equation 8.79, the fractional gain error in $H(e^{j\omega T})$ when $\omega T = 0.5$ rad is 0.010493 (see Figure 8.14). Solving for $|H(e^{j\omega T})|$ in Equation 8.79,

TABLE 8.3
Measured Peak-to-Peak Swing in $x_A(n)$
for Different Time Periods

Duration of Simulation (s)	$\text{Max}(x_A) - \text{Min}(x_A)$
$P = 2\pi/\omega = \pi$	1.010481
$25P = 25\pi$	1.010491
$50P = 50\pi$	1.010491
$100P = 100\pi$	1.010492

$$\begin{aligned} |H(e^{j\omega T})| &= (1 + \text{fractional gain error}) |H(j\omega)| \\ &= (1 + 0.010493) \left| \frac{1}{j2} \right| = 0.505247 \end{aligned} \quad (8.91)$$

Since $|u(t)| = 1$, the predicted peak-to-peak swing in $x_A(n)$ is $2 \times 0.505247 = 1.010494$. The discrete-time response $x_A(n)$ was generated for different lengths of time (instead of two periods as in Figure 8.17) to capture the peak-to-peak swing in $x_A(n)$ using the MATLAB statement “max (x_A) – min (x_A)” in “Ch8_Fig8_17.m.” The results are tabulated in Table 8.3.

The time delay was computed in the same manner used for the case when $\omega T = 0.05$ rad. The results are $n = 4/0.25 = 16$, $x_A(16) = 0.4371$, $x(3.8639) = 0.4371$ and the time delay is equal to $4 - 3.8639 = 0.1361$ s. The asymptotic formula in Equation 8.87 for $H(e^{j\omega T})$ underestimates the time delay, that is, $T/2 = 0.125$ s.

8.2.4 ASYMPTOTIC FORMULAS FOR MULTISTEP INTEGRATION METHODS

The same steps used to obtain the asymptotic formula in Equation 8.84 for the explicit Euler integrator are applicable to the multistep integration formulas introduced in Section 6.4. For example, simulating the response $x(t)$ of a continuous-time integrator subject to input $u(t)$, using a second-order explicit Adams–Bashforth (AB-2) numerical integrator, reduces to solve the difference equation

$$x_A(n+1) = x_A(n) + \frac{T}{2}[3u(n) - u(n-1)] \quad (8.92)$$

z -Transforming Equation 8.92 and solving for the z -domain transfer function give

$$\frac{X(z)}{U(z)} = H_I(z) = \frac{T}{2} \left[\frac{3 - z^{-1}}{z - 1} \right] \quad (8.93)$$

where the subscript I in $H_I(z)$ reminds us that we are dealing with the z -domain transfer function approximation of a continuous-time integrator. Replacing z by $e^{j\omega T}$ in Equation 8.93 produces the discrete-time system frequency response function

$$H_I(e^{j\omega T}) = \frac{T}{2} \left[\frac{3 - e^{-j\omega T}}{e^{-j\omega T} - 1} \right] \quad (8.94)$$

Approximating the complex exponentials $e^{j\omega T}$ and $e^{-j\omega T}$ by power series up to the third-order term generates the asymptotic formula (see Exercise 8.10)

$$H_I(e^{j\omega T}) \approx \frac{1}{j\omega} \left[\frac{1}{1 - (5/12)(\omega T)^2} \right], \quad \omega T \ll 1 \quad (8.95)$$

According to the asymptotic approximation in Equation 8.95, the frequency response of an AB-2 integrator is identical in phase to that of an ideal continuous-time integrator while the gain is off by the factor in parenthesis in Equation 8.95. Hence, the phase error in the asymptotic approximation of $H_I(e^{j\omega T})$ is zero and the fractional gain error is

$$\frac{|H_I(e^{j\omega T})|}{|H(j\omega)|} - 1 \approx \left[\frac{1}{1 - (5/12)(\omega T)^2} \right] - 1 \quad (8.96)$$

$$\approx \frac{(5/12)(\omega T)^2}{1 - (5/12)(\omega T)^2} \quad (8.97)$$

$$\approx (5/12)(\omega T)^2, \quad |\omega T| \ll 1 \quad (8.98)$$

Equations 8.84 and 8.95 are special cases of a general formula (Howe 1986, 1995)

$$H_I(e^{j\omega T}) \approx \frac{1}{j\omega} \left[\frac{1}{1 + e_I(j\omega T)^k} \right], \quad \omega T \ll 1 \quad (8.99)$$

which holds for the multistep integrators in Section 6.4, namely, explicit Adams–Bashforth, implicit Adams–Moulton, and predictor–correctors. Numerical values for the error coefficient e_I depend on the order k and type of integrator. A table of values for low-order numerical integrators is given in [Table 8.4](#). The error coefficients are identical to the constants in the local truncation error term for each integrator (see [Table 6.9](#)).

Frequency responses for a continuous-time integrator and AB-1 (explicit Euler) through AB-4 numerical integrators are shown in [Figure 8.18a](#) through d. Also shown are the frequency responses for the same AB integrators based on the asymptotic formula in Equation 8.99 where $e_I, k = 1, 2, 3, 4$ are given in [Table 8.4](#). The plots are generated in M-file “*Ch8_Fig8_18abcd.m*.”

[Figure 8.18a](#) shows close agreement between the exact and asymptotic Euler magnitude functions up to $\omega T = \omega(1) \approx 0.5$ rad. Beyond that, the two plots begin to deviate from the continuous-time integrator magnitude function with the exact Euler the better approximation. Hence, for $\omega T \ll 1$, the Euler integrator introduces essentially zero gain error. The exact and asymptotic Euler phase plots also agree up to approximately $\omega T = 0.5$ rad. However, [Figure 8.18](#) shows the Euler integrator introducing phase error with respect to the continuous-time integrator beginning around $\omega T = 0.04$ rad. Significant phase error in the neighborhood of 30° is present for $\omega T = 1$ rad.

The AB-2 integrator and its asymptotic approximation are both quite accurate in the range of frequencies for which $\omega T < 0.4$ rad. Beyond that, the asymptotic approximation of the magnitude begins to deviate from both the continuous-time and exact AB-2 magnitude functions. From Equation 8.99 with $k = 2$, the asymptotic curve approaches infinity at the point where

$$1 + e_I(j\omega T)^k = 1 + e_I(j)^2(\omega T)^2 = 1 - \frac{5}{12}\omega^2 = 0 \quad (8.100)$$

$$\Rightarrow \omega = 1.5492 \text{ rad/s}$$

TABLE 8.4

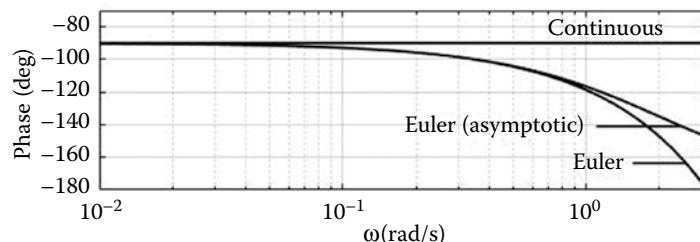
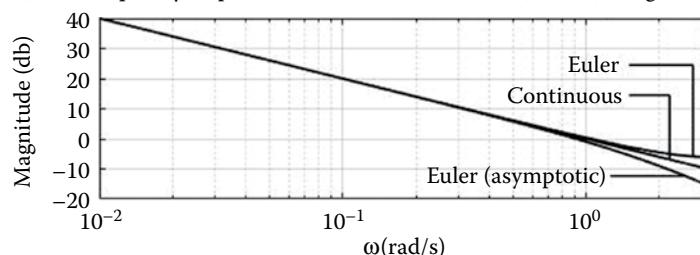
Error Coefficients in Asymptotic Formula in Equation 8.99 for k th Order, z-Domain Frequency Response Functions of Numerical Integrators

Numerical Integrator	Equation	Order k	Error Coefficient e_k
AB-1 (explicit Euler)		1	1/2
AB-2	6.180	2	5/12
AB-3	6.186	3	3/8
AB-4	6.187	4	251/720
AB-5	6.188	5	475/1440
AM-2 (trapezoidal)	6.191	2	-1/12
AM-3	6.192	3	-1/24
AM-4	6.193	4	-19/720
AM-5	6.194	5	-27/1440
AB-2 predictor	6.204	2 ^a	-1/12 ^a
AM-2 corrector	6.205		
AB-3 predictor	6.206	3 ^b	-1/24 ^b
AM-3 corrector	6.207		
AB-4 predictor	6.208	4 ^c	-19/720 ^c
AM-4 corrector	6.209		

^a AB-2/AM-2.

^b AB-3/AM-3.

^c AB-4/AM-4.

(a) Frequency response of continuous and Euler ($T = 1$ s) integrators**FIGURE 8.18** Exact and asymptotic frequency response of (a) AB-1 (Euler) integrator.

(Continued)

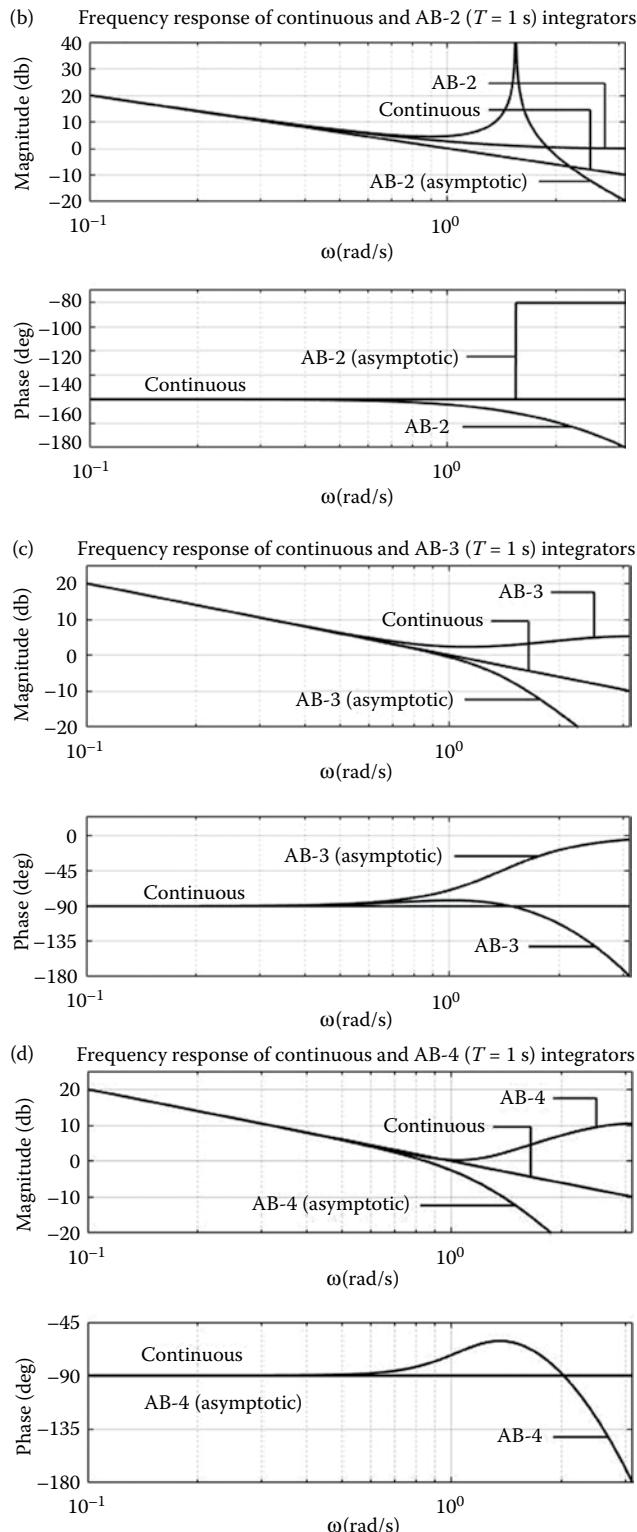


FIGURE 8.18 (Continued) Exact and asymptotic frequency response of (b) AB-2 integrator, (c) AB-3 integrator, and (d) AB-4 integrator.

The asymptotic phase plot is exact up to $\omega = 1.5492$ rad/s where it increases from -90° to 90° due to the change in sign of the denominator. It follows from Equation 8.99 that the asymptotic phase plots for $k = 2, 6, 10, \dots$ are similar to the one in Figure 8.18b and the asymptotic plots are exact, that is, $\arg\{H(e^{j\omega T})\} = -180$ deg for $k = 4, 8, 12, \dots$

8.2.5 SIMULATION OF LINEAR SYSTEM WITH TRANSFER FUNCTION $H(s)$

In addition to a simple continuous-time integrator, it is possible to approximate the discrete-time frequency response of higher-order linear systems simulated using numerical integrators like the ones represented in Table 8.4. We learned in Section 4.7 that $H(z)$ resulting from digital simulation of a continuous-time system with transfer function $H(s)$ is obtained by substituting $1/H_I(z)$ for s , where $H_I(z)$ is the z -domain transfer function of the numerical integrator.

Consider the first-order system governed by

$$\frac{dx}{dt} = \lambda x + Ku \quad (8.101)$$

similar to Equation 8.1 except for the gain K on the right-hand side of the equation. Simulation of the system using a numerical integrator with transfer function $H_I(z)$ results in a discrete-time system with frequency response function

$$H(e^{j\omega T}) = H(s) \Big|_{s \leftarrow 1/H_I(e^{j\omega T})} = \frac{K}{s - \lambda} \Big|_{s \leftarrow 1/H_I(e^{j\omega T})} \quad (8.102)$$

$$= \frac{KH_I(e^{j\omega T})}{1 - \lambda H_I(e^{j\omega T})} \quad (8.103)$$

An approximate expression for $H(e^{j\omega T})$ is obtained using the asymptotic approximation for $H(e^{j\omega T})$ in Equation 8.99.

$$H(e^{j\omega T}) \approx \frac{K(1/j\omega[1 + e_I(j\omega T)^k])}{1 - \lambda(1/j\omega[1 + e_I(j\omega T)^k])}, \quad \omega T \ll 1 \quad (8.104)$$

$$\approx \frac{K}{j\omega[1 + e_I(j\omega T)^k] - \lambda}, \quad \omega T \ll 1 \quad (8.105)$$

Using trapezoidal integration, $k = 2$, $e_I = -1/12$ from Table 8.4,

$$H(e^{j\omega T}) \approx \frac{K}{j\omega[1 + (1/12)(\omega T)^2] - \lambda}, \quad \omega T \ll 1 \quad (8.106)$$

The exact expression for $H(e^{j\omega T})$ is obtained from

$$H(z) = H(s) \Big|_{s \leftarrow 1/H_I(z)} = \frac{K}{s - \lambda} \Big|_{s \leftarrow \frac{2(z-1)}{T(z+1)}} \quad (8.107)$$

$$\Rightarrow H(e^{j\omega T}) = \frac{K}{((2/T)(z-1)/(z+1)) - \lambda} \Big|_{z=e^{j\omega T}} \quad (8.108)$$

$$= \frac{KT(e^{j\omega T} + 1)}{(2 - \lambda T)e^{j\omega T} - (2 + \lambda T)} \quad (8.109)$$

The use of trapezoidal integration for digital simulation of linear continuous-time systems is referred to as Tustin's method.

EXAMPLE 8.3

The capacitor in the circuit shown in Figure 8.19 is initially uncharged when the switch closes.

- Find the discrete-time transfer function $H(z) = V_c(z)/E_0(z)$ using Tustin's method.
- Find the asymptotic form of the discrete-time frequency response function.
- Find the exact expression for the discrete-time frequency response function.
- Graph the frequency response function of the continuous-time system and the discrete-time frequency response functions obtained in parts (b) and (c) if the time constant of the circuit is 25 μs and the integration step size is 1 μs .
- Compute the gain and phase errors based on the asymptotic expression for $H(e^{j\omega T})$ when the input is a sinusoidal input at 1×10^5 Hz.
- The differential equation of the circuit is

$$\tau \frac{d}{dt} v_c(t) + v_c(t) = e_0(t), \quad (\tau = RC) \quad (8.110)$$

Laplace transforming Equation 8.110 leads to the transfer function

$$H(s) = \frac{V_c(s)}{E_0(s)} = \frac{1}{\tau s + 1} = \frac{1/\tau}{s + 1/\tau} \quad (8.111)$$

$$\Rightarrow H(z) = \frac{V_c(z)}{E_0(z)} = \frac{1}{\tau s + 1} \Big|_{s=\frac{2(z-1)}{T(z+1)}} \quad (8.112)$$

$$= \frac{z+1}{[1+2(\tau/T)]z + (1-2\tau/T)} \quad (8.113)$$

- The transfer function for the first-order system in Equation 8.101 is identical to $H(s)$ in Equation 8.111 when $\lambda = -1/\tau$ and $K = -1/\tau$. Making those substitutions in Equation 8.106 gives the asymptotic formula for $H(e^{j\omega t})$.

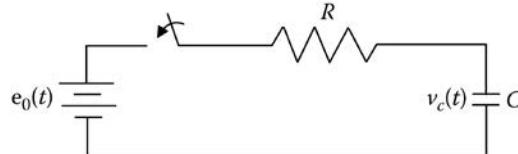


FIGURE 8.19 RC circuit for digital simulation.

$$H(e^{j\omega T}) \approx \frac{1/\tau}{j\omega[1 + (1/12)(\omega T)^2] - (-1/\tau)}, \quad \omega T \ll 1 \quad (8.114)$$

$$\approx \frac{1}{j\omega\tau[1 + (1/12)(\omega T)^2] + 1}, \quad \omega T \ll 1 \quad (8.115)$$

c. The exact expression for $H(e^{j\omega T})$ is from Equation 8.109,

$$H(e^{j\omega T}) = \left. \frac{(T/\tau)(e^{j\omega T} + 1)}{(2 - \lambda T)e^{j\omega T} - (2 + \lambda T)} \right|_{\lambda = -1/\tau} \quad (8.116)$$

$$= \frac{(T/\tau)(e^{j\omega T} + 1)}{[2 + (T/\tau)] e^{j\omega T} - [2 - (T/\tau)]} \quad (8.117)$$

d. The MATLAB M-file “Ch8_Ex8_3.m” computes the magnitude and phase of

$$H(j\omega) = \left. \frac{1}{\tau s + 1} \right|_{s \leftarrow j\omega} = \frac{1}{1 + j\omega\tau} \quad (8.118)$$

The magnitude and phase of $H(e^{j\omega T})$ using the asymptotic and exact formulas in Equations 8.115 and 8.117 are computed. The gain and phase plots are shown in Figure 8.20.

e. At $\omega = 1 \times 10^5$ Hz $\times 2\pi$ rad/cycle $= 2\pi \times 10^5$ rad/s,

$$H(j2\pi \times 10^5) = \frac{1}{1 + j2\pi \times 10^5(25 \times 10^{-6})} = \frac{1}{1 + j5\pi} = 0.0635 e^{j(-1.5072)} \quad (8.119)$$

$$H(e^{j2\pi \times 10^5 \times 10^{-6}}) \approx \frac{1}{j2\pi \times 10^5(25 \times 10^{-6})[1 + (1/12)(2\pi \times 10^5 \times 10^{-6})^2] + 1} \quad (8.120)$$

$$\approx 0.0615 e^{j(-1.5092)} \quad (8.121)$$

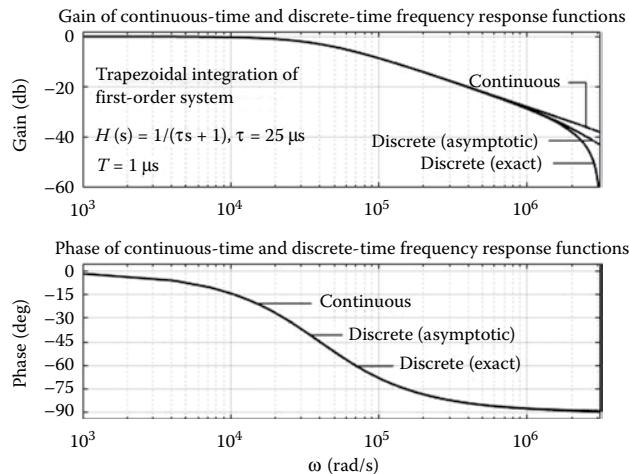


FIGURE 8.20 Continuous-and discrete-time (exact and asymptotic) bode plots for first-order system using trapezoidal integration.

The fractional gain error and phase errors in the discrete-time frequency response function based on the asymptotic approximation for $H(e^{j\omega T})$ in Equation 8.115 at $\omega = 1 \times 10^5$ Hz are

$$\begin{aligned}\text{Fractional gain error} &\approx \frac{|H(e^{j2\pi \times 10^5 \times 10^{-6}})|}{|H(j2\pi \times 10^5)|} - 1 \\ &\approx \frac{0.0635}{0.0615} - 1 = -0.0317\end{aligned}\quad (8.122)$$

$$\begin{aligned}\text{Phase error} &\approx \text{Arg}[H(e^{j2\pi \times 10^5 \times 10^{-6}})] - \text{Arg}[H(j2\pi \times 10^5)] \\ &\approx -1.5092 - (-1.5072) \\ &\approx -0.002 \text{ rad}(-0.1157 \text{ deg})\end{aligned}\quad (8.123)$$

[Figure 8.20](#) shows the asymptotic formula for approximating $H(e^{j\omega T})$ is accurate up to approximately $\omega T = 10^6 \text{ rad/s} \times 10^{-6} \text{ s} = 1 \text{ rad}$. The exact and asymptotic discrete-time frequency response functions are close to the continuous-time frequency response function up until frequencies approaching the Nyquist frequency $\pi/T = 10^6 \pi \text{ rad/s}$.

For an underdamped second-order system with damping ratio ζ and natural frequency ω_n , the asymptotic approximation of the discrete-time frequency response function using a k th-order numerical integrator with error coefficient e_l is obtained in the same manner employed for the first-order system, namely,

$$H(e^{j\omega T}) = \left. \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right|_{s \leftarrow 1/H_l(e^{j\omega T})} \quad (8.124)$$

Substituting the asymptotic expression for $H(e^{j\omega T})$ in Equation 8.99 into Equation 8.124 results in (after simplification) (Howe 1986)

$$H(e^{j\omega T}) \approx \frac{1}{\omega_n^2 - \omega^2[1 + 2e_l(j\omega T)^k] + j2\zeta\omega_n\omega[1 + e_l(j\omega T)^k]}, \quad \omega T \ll 1 \quad (8.125)$$

and the fractional error in $H(e^{j\omega T})$ is approximated by the asymptotic formula

$$e_H = \frac{H(e^{j\omega T})}{H(j\omega)} - 1 \approx \frac{2e_l(j\omega T)^k[(\omega/\omega_n)^2 - j\zeta(\omega/\omega_n)]}{1 - (\omega/\omega_n)^2 + j2\zeta(\omega/\omega_n)}, \quad \omega T \ll 1 \quad (8.126)$$

Rationalizing Equation 8.126 leads to expressions for e_M and e_A , the real and imaginary components of e_H , which provide suitable approximations for the fractional gain error and phase error of $H(e^{j\omega T})$, respectively. The expressions are of the form

$$e_M = f_M(\zeta, \omega/\omega_n, k, e_l)(\omega T)^k, \quad e_A = f_A(\zeta, \omega/\omega_n, k, e_l)(\omega T)^k, \quad \omega T \ll 1 \quad (8.127)$$

The functions $f_M(\zeta, \omega/\omega_n, k, e_l)$ and $f_A(\zeta, \omega/\omega_n, k, e_l)$ are further addressed in Exercise 8.13. The notable feature in Equation 8.127 is the dependence of both error measures on the term $(\omega T)^k$, emphasizing the importance of choosing the step size T and the integrator order k .

EXERCISES

- 8.1 Repeat Example 8.1 for the case where explicit Euler is used in place of trapezoidal integration.
- 8.2 A second-order system with damping ratio $\zeta = 0$ and natural frequency ω_n is simulated using trapezoidal integration with step size T .
 - a. Plot the equivalent continuous-time system damping ratio ζ^* as a function of the parameter $\omega_n T$ for $0 \leq \omega_n T \leq 1$.

- b. Plot the equivalent continuous-time system natural frequency ω_n^* as a function of the continuous-time system natural frequency ω_n for $0 \leq \omega_n \leq 10$ rad/s when $T = 0.1$ s.
 - c. Repeat parts (a) and (b) for $\zeta = 0.1$ and 1.
 - d. What effect does changing the value of T have on the equivalent continuous-time system damping ratio and natural frequency?
- 8.3 Consider the overdamped continuous-time second-order system with transfer function

$$H(s) = \frac{1}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

shown in Figure E8.3.

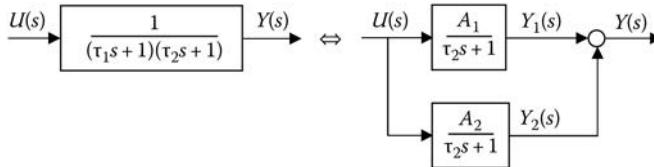


FIGURE E8.3

- a. Decompose $H(s)$ into the sum of two first-order transfer functions $H(s) = H_1(s) + H_2(s)$ where $H_1(s) = A_1/(\tau_1 s + 1)$ and $H_2(s) = A_2/(\tau_2 s + 1)$ (see above figure) and express the constants A_1 and A_2 in terms of the time constants τ_1 and τ_2 .
- b. The equivalent realizations of the same second-order system are simulated using explicit Euler integration with step size T . Find the fractional error in the frequency response functions $H(e^{j\omega T})$, $H_1(e^{j\omega T})$, and $H_2(e^{j\omega T})$. Leave your answers in terms of τ_1 , τ_2 , and T .
- c. Resolve the fractional errors into real and imaginary components, that is,

$$\begin{aligned} e_H &= \frac{H(e^{j\omega T})}{H(j\omega)} - 1 = e_M + j e_A, \quad e_{H_1} = \frac{H_1(e^{j\omega T})}{H_1(j\omega)} - 1 = e_{M1} + j e_{A1} \\ e_{H_2} &= \frac{H_2(e^{j\omega T})}{H_2(j\omega)} - 1 = e_{M2} + j e_{A2} \end{aligned}$$

For $\tau_1 = 1$ s, $\tau_2 = 10$ s, and $T = 0.05$ s, plot e_M , e_{M1} , e_{M2} vs. ωT on a single graph and e_A , e_{A1} , e_{A2} , vs. ωT on a different graph. Comment on the results.

- d. Find exact expressions for the fractional gain error in $H(e^{j\omega T})$, $H_1(e^{j\omega T})$, and $H_2(e^{j\omega T})$. Plot the fractional gain error in $H(e^{j\omega T})$ vs. ωT and e_M vs. ωT on the same graph. Repeat for $H_1(e^{j\omega T})$ and e_{M1} and then for $H_2(e^{j\omega T})$ and e_{M2} .
- e. Find exact expressions for the phase error in $H(e^{j\omega T})$, $H_1(e^{j\omega T})$, and $H_2(e^{j\omega T})$. Plot the phase error in $H(e^{j\omega T})$ vs. ωT and e_A vs. ωT on the same graph. Repeat for $H_1(e^{j\omega T})$ and e_{A1} and then for $H_2(e^{j\omega T})$ and e_{A2} .
- f. Simulate the two configurations shown in Figure E8.3 when $u(t) = \sin 50t$, $t \geq 0$ using an explicit Euler integrator with step size $T = 0.01$ s. Plot the continuous-time input and output and the simulated response on the same graph for each configuration. Do the results agree with the graphs obtained in parts (d) and (e)?

- 8.4 For AB-2 integration,

- a. Find the discrete-time frequency response function $H_1(e^{j\omega T})$.
- b. Find expressions for the exact and asymptotic fractional gain and phase errors.
- c. Plot the results over a suitable range of values for ωT .

- 8.5 Generate a new table and figure similar to [Table 8.2](#) and [Figure 8.9](#) where a second-order system with damping ratio ζ and natural frequency ω_n is simulated using numerical integration for the following cases:

Z	ω_n (rad/s)	Numerical Integrator
0	50	Explicit Euler
0	50	Implicit Euler
0	50	Trapezoidal
0.1	50	Implicit Euler
0.1	50	Trapezoidal
0.707	1	Explicit Euler
0.707	1	Implicit Euler
0.707	1	Explicit Euler
2	0.01	Explicit Euler
2	0.01	Implicit Euler
2	0.01	Trapezoidal

- 8.6 For a second-order system described by

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = K\omega_n^2u$$

- a. Show that the unit impulse response is identical to the response of the autonomous system ($u = 0, t \geq 0$) with initial conditions $x(0) = 0, \dot{x}(0) = K\omega_n^2$.
 - b. Show that the initial conditions for the difference equation of the discrete-time system resulting from the use of explicit Euler integration are $x(0) = 0$ and $x(-1) = -K\omega_n^2T$.
 - c. Suppose the parameter values are $\zeta = 0.5$, $\omega_n = 10$, and $K = 1$. Simulate the continuous-time step and impulse responses using explicit Euler integration with $\omega_n T = 0.05$. Compare the simulated and analytical solutions.
- 8.7 Derive the exact expressions for the fractional gain error and phase error in the discrete-time transfer function $H(e^{j\omega T})$ using explicit Euler integration given in Equations 8.79 and 8.80.
- 8.8 Show that the asymptotic expression for the fractional error in the discrete-time transfer function $H(e^{j\omega T})$ resulting from explicit Euler integration of the first-order system $\dot{x} = \lambda x + u$ is given by

$$e_H = \frac{H(e^{j\omega T})}{H(j\omega)} - 1 \approx \frac{\omega\lambda}{2(\omega^2 + \lambda^2)}\omega T - j\frac{\omega^2}{2(\omega^2 + \lambda^2)}\omega T, \quad \omega T \ll 1$$

What does the system reduce to when $\lambda = 0$? Comment on what happens to the real and imaginary components.

- 8.9 Verify the curves plotted in [Figures 8.14](#) and [8.15](#) for the fractional gain and phase errors based on explicit Euler integration by using the MATLAB functions “real,” “imag,” “abs,” and “angle,” that is,

$$\text{Fractional gain error} \approx e_M = \text{Re}(e_H) = R_e \left\{ \frac{H(e^{j\omega T})}{H(j\omega)} - 1 \right\} = \text{Re} \left\{ \frac{j\omega T}{e^{j\omega T} - 1} - 1 \right\}$$

$$\text{Fractional gain error} = \frac{|H(e^{j\omega T})|}{|H(j\omega)|} - 1 = \left| \frac{\omega T}{e^{j\omega T} - 1} \right| - 1$$

$$\text{Phase error} \approx e_A = \text{Im}(e_H) = \text{Im} \left\{ \frac{H(e^{j\omega T})}{H(j\omega)} - 1 \right\} = \text{Im} \left\{ \frac{j\omega T}{e^{j\omega T} - 1} - 1 \right\}$$

$$\text{Phase error} = \text{Arg}[H(e^{j\omega T})] - \text{Arg}[H(j\omega)] = -\text{Arg}[e^{j\omega T} - 1] - \left(-\frac{\pi}{2} \right)$$

- 8.10 Derive the asymptotic formula for $H_1(e^{j\omega T})$ in Equation 8.95 starting with the exact expression for the discrete-time frequency response function in Equation 8.94.
- 8.11 Using trapezoidal integration to simulate the first-order system in Equation 8.1,
 a. Find the fractional error in transfer function e_H .
Hint: Start with Equation 8.109.
 b. Find the real and imaginary parts of e_H , that is, $e_H = e_M + j e_A$.
 c. Compare e_M and e_A with the exact expressions for the fractional gain and phase errors.
- 8.12 For simulation of the first-order system $\dot{x} = \lambda x + u$ using a k th-order numerical integrator with error coefficient e_I
 a. Show that the asymptotic expression for the fractional error in transfer function is given by

$$e_H = \frac{H(e^{j\omega T})}{H(j\omega)} - 1 = \frac{j\omega e_I (j\omega T)^k}{j\omega - \lambda}, \quad \omega T \ll 1$$

- b. Derive expressions for e_M and e_A when the order k is odd and different expressions when k is even.
- 8.13 Derive the asymptotic expressions for $H(e^{j\omega T})$ in Equation 8.125 and e_H in Equation 8.126. Find the functions $f_M(\zeta, \omega/\omega_n, k, e_I)$ and $f_A(\zeta, \omega/\omega_n, k, e_I)$ in Equation 8.127 when the numerical integrator order k is odd and even.
- 8.14 Show that the characteristic root error resulting from simulation of a first-order continuous-time system with characteristic root λ is approximated by

$$e_\lambda = \frac{\lambda^*}{\lambda} - 1 \approx -e_I(\lambda T)^k, \quad |\lambda T| \ll 1$$

where e_I and k are the error coefficient and order of the numerical integrator, respectively.

8.3 STABILITY OF NUMERICAL INTEGRATORS

We have seen a number of examples where digital simulation of a stable continuous-time system with a bounded input (or even no input with nonzero initial conditions) produced a sequence of numbers that grow without bound as time increases. The unstable conditions can be attributed to a combination of the numerical integrator and integration step size (for fixed-step integrators). Stability of fixed-step numerical integrators is reflected in the natural dynamics of the discrete-time system used to approximate the continuous-time system. The family of explicit multistep Adams–Bashforth integrators introduced in Section 6.4 is now examined in some detail.

8.3.1 ADAMS–BASHFORTH NUMERICAL INTEGRATORS

Difference equations resulting from the application of second-order and higher Adams–Bashforth integration are higher-order than the LTI continuous-time systems being simulated. For example, a first-order continuous-time system with a pole at $s = \lambda$ simulated using AB-2 integration produces a second-order discrete-time system with discrete-time input $u(n)$ and output $x(n)$, previously referred to as $x_A(n)$. The z -domain transfer function is

$$H(z) = \frac{X(z)}{U(z)} = \frac{1}{s - \lambda} \Big|_{s \leftarrow \frac{1}{H_1(z)}} \quad (8.128)$$

$$= \frac{1}{s - \lambda} \Big|_{s \leftarrow \frac{1}{T(3z-1)/2z(z-1)}} \quad (8.129)$$

$$= \frac{(T/2)(3z-1)}{z^2 - (1 + (3/2)\lambda T)z + (1/2)\lambda T} \quad (8.130)$$

Note that $H_1(z)$ for AB-2 integration is given in Equation 8.93 of the previous section. Multiplying numerator and denominator in Equation 8.130 by z^{-1} followed by inverse z -transformation leads to the second-order difference equation

$$x(n+1) - \left(1 + \frac{3}{2}\lambda T\right)x(n) + \frac{1}{2}\lambda T x(n-1) = \frac{T}{2}[3u(n) - u(n-1)] \quad (8.131)$$

The states $x(n)$ and $x(n-1)$ are needed to compute the updated state $x(n+1)$. This is easily explained by referring to Figure 6.16. $P_1(t)$; the linear interpolating polynomial integrated to generate $x(n+1)$ depends on current and previous derivative functions, which in turn are functions of the current and previous discrete-time states $x(n)$ and $x(n-1)$.

The resulting z -domain transfer function in Equation 8.130 has two poles that are the roots of the characteristic polynomial in the denominator. The dominant pole for the case when $\lambda T \ll 1$ corresponds to an equivalent continuous-time system characteristic root λ^* , which can be estimated from the characteristic root error formula (Howe 1986)

$$e_\lambda = \frac{\lambda^* - \lambda}{\lambda} \approx -e_I(\lambda T)^k \quad (8.132)$$

where e_I and k are the integrator error coefficient and order, respectively. For AB-2 integration, $e_I = 5/12$ and $k = 2$. Hence,

$$\lambda^* = \lambda[1 - e_I(\lambda T)^k] \approx \lambda \left[1 - \frac{5}{12}(\lambda T)^2\right], \quad \lambda T \ll 1 \quad (8.133)$$

Suppose the continuous-time system pole is $\lambda = -100$ and AB-2 integration is used with a step size $T = 0.0001$ s. From Equation 8.133,

$$\begin{aligned} \lambda^* &\approx -100 \left[1 - \frac{5}{12} \{(-100)(0.0001)\}^2\right], \quad \lambda T \ll 1 \\ &\approx -99.99583 \end{aligned} \quad (8.134)$$

The exact value of λ^* is obtained from

$$\lambda^* = \frac{1}{T} \ln z_l \quad (8.135)$$

where z_1 is the dominant pole, that is, larger (in magnitude) root of the characteristic equation

$$z^2 - \left(1 + \frac{3}{2}\lambda T\right)z + \frac{1}{2}\lambda T = z^2 - 0.985z - 0.005 = 0 \quad (8.136)$$

The poles are located at $z_1 = 0.99005$, $z_2 = -0.00505$, and the equivalent characteristic root is from Equation 8.135

$$\lambda^* = \frac{1}{0.0001} \ln(0.99005) = -99.99581$$

There is no real equivalent system characteristic root for the extraneous pole z_2 ; however, $x(n)$, $n = 0, 1, 2, \dots$ does include a transient component $c_2 z_2^k$, which rapidly vanishes to zero leaving the dominant component $c_1 z_1^k$ and input mode (if present) terms to accurately track the continuous-time system response $x(t)$, $t \geq 0$.

Numerical stability of the simulation becomes an issue when the AB-2 integration step size produces z -plane poles in proximity of the Unit Circle. For a given first-order continuous-time system with characteristic root $\lambda < 0$, the discrete-time system resulting from AB-2 integration is marginally stable when the dominant pole is located at 1 or -1 . From Equation 8.136,

$$z = 1: \quad (1)^2 - \left(1 + \frac{3}{2}\lambda T\right)(1) + \frac{1}{2}\lambda T = 0 \Rightarrow \lambda T = 0 \quad (8.137)$$

$$z = -1: \quad (-1)^2 - \left(1 + \frac{3}{2}\lambda T\right)(-1) + \frac{1}{2}\lambda T = 0 \Rightarrow \lambda T = -1 \quad (8.138)$$

Combining the above two results imposes the condition for stability, namely,

$$-1 < \lambda T < 0 \Rightarrow 1 > -\lambda T > 0 \Rightarrow T < \frac{1}{-\lambda} \quad (8.139)$$

In other words, the AB-2 integration step size T is limited by the time constant $\tau = -(1/\lambda)$ of the first-order continuous-time system. Where is the second z -plane pole when $\lambda T = 0$ and $\lambda T = -1$?

Second-order systems can be analyzed in the same way by allowing λ to be complex in the case of an underdamped second-order system or a pair of distinct real values for an overdamped second-order system. For example, a stable, second-order system with complex poles located at $-7.5 \pm j5$ simulated with AB-2 integration using a step size $T = 0.1$ s generates a stable discrete-time system if the two z -plane poles (principal and extraneous) are located inside the Unit Circle. This is easily checked by substituting $\lambda T = (-7.5 + j5)(0.1) = -0.75 + j0.5$ into the characteristic equation,

$$z^2 - \left(1 + \frac{3}{2}\lambda T\right)z + \frac{1}{2}\lambda T \Big|_{\lambda T = -0.75 + j0.5} = z^2 + (0.125 - j0.75)z - 0.375 + j0.25 = 0 \quad (8.140)$$

Solution of Equation 8.140 reveals that the poles are located inside the Unit Circle at

$$z_1 = -0.6188 + j0.6418 = 0.8916e^{j2.3379}$$

$$z_2 = 0.4938 + j0.1082 = 0.2157e^{j0.2157}$$

and the discrete-time system is therefore stable. Increasing T eventually causes one of the z -plane poles to be on the Unit Circle where the system becomes marginally stable.

A closed locus of λT points can be identified in the complex plane with the property that all interior points produce stable discrete-time systems using AB-2 integration. The locus of points is called a stability boundary and the interior points comprise the stability region. There is a different stability boundary for each AB integrator.

The starting point for locating the stability boundary is finding $H(z)$, the z -domain transfer function of the discrete-time system resulting from numerical integration of the stable, continuous-time system

$$\frac{dx}{dt} + \lambda x = u, \quad \operatorname{Re}(\lambda) < 0 \quad (8.141)$$

A similar approach to the one used for finding $H(z)$ for AB-2 integration of the continuous-time system in Equation 8.141 is employed to find $H(z)$ for different-order AB integrators. For AB-1 (Euler), AB-3, and AB-4 integration, $H_1(z)$ in Equation 8.128 is

$$\text{AB-1: } H_1(z) = \frac{T}{z-1} \quad (8.142)$$

$$\text{AB-3: } H_1(z) = \frac{T}{12} \left[\frac{23z^2 - 16z + 5}{z^2(z-1)} \right] \quad (8.143)$$

$$\text{AB-4: } H_1(z) = \frac{T}{24} \left[\frac{55z^3 - 59z^2 + 37z - 9}{z^3(z-1)} \right] \quad (8.144)$$

Replacing s by $1/H_1(z)$ in $H(s) = 1/(s - \lambda)$ leads to the z -domain transfer function $H(z)$. For AB-1 through AB-4 integration, the results are

$$\text{AB-1: } H(z) = \frac{T}{z - (1 + \lambda T)} \quad (8.145)$$

$$\text{AB-2: } H(z) = \frac{T(3z - 1)}{2z^2 - (2 + 3\lambda T)z + \lambda T} \quad (8.146)$$

$$\text{AB-3: } H(z) = \frac{T(23z^2 - 16z + 5)}{12z^3 - (12 + 23\lambda T)z^2 + 16\lambda Tz - 5\lambda T} \quad (8.147)$$

$$\text{AB-4: } H(z) = \frac{T(55z^3 - 59z^2 + 37z - 9)}{24z^4 - (24 + 55\lambda T)z^3 + 59\lambda Tz^2 - 37\lambda Tz + 9\lambda T} \quad (8.148)$$

Note the existence of one, two, and three extraneous z -plane poles in Equations 8.146 through 8.148. The stability boundaries are obtained by setting $z = e^{j\theta}$ in the denominators of Equations 8.145 through 8.148 and solving for λT . For example, with AB-3 integration, λT is given by

$$\lambda T = 12 \left(\frac{e^{j3\theta} - e^{j2\theta}}{5 - 16e^{j\theta} + 23e^{j2\theta}} \right) \quad (8.149)$$

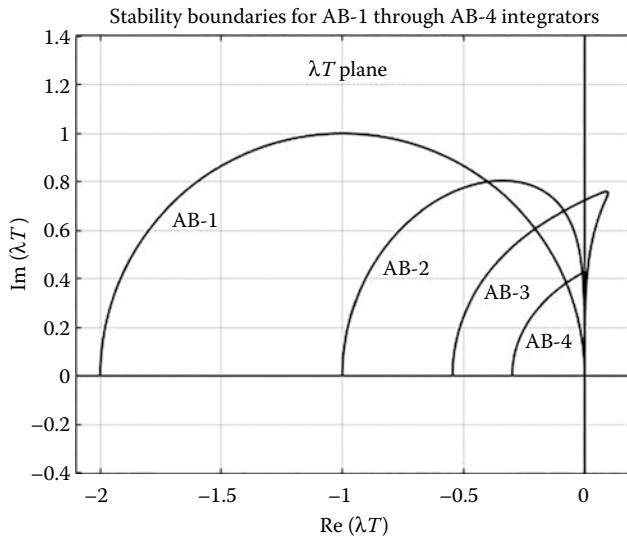


FIGURE 8.21 Stability boundaries for AB-1 through AB-4 integration.

Results for AB-1, AB-2, AB-3, and AB-4 integrators are obtained in the MATLAB M-file “*Ch8_AB_Stability_Boundaries.m*” and shown in [Figure 8.21](#).

Only the top half of each stability boundary is shown since they are symmetric with respect to the real axis. Points along the top half of a stability boundary are computed by varying θ from 0 to π rad causing $e^{j\theta}$ to traverse the top half of the Unit Circle. The lower half is generated by sweeping θ from 0 to $-\pi$ rad.

A note of caution in finding the stability boundaries. The pole moving along the Unit Circle must be the largest in magnitude. For example, in the case of AB-4, the additional three poles must lie inside the Unit Circle. Values of λT for which this is not the case are ignored, that is, they are not points on the stability boundary (see Exercise 8.17).

[Figure 8.21](#) confirms the result in Equation 8.139, namely, $\lambda T < -1$ for AB-2 integration of a stable, first-order system with real characteristic root λ . AB-1 integration is explicit Euler, and it is clear from Equation 8.145 that the lone z -plane pole of $H(z)$ migrates to $z = -1 = 1e^{j\pi}$ when $\lambda T = -2$, also confirmed by observing the leftmost point on the AB-1 stability boundary.

The equation of the stability boundary for AB-1 integration in the λT plane is easily derived. [Figure 8.22](#) shows the z -plane pole of $H(z)$ in Equation 8.145 varying from 1 to -1 along the Unit Circle as θ increases from zero to π .

From Equation 8.145, the parameter λT on the AB-1 stability boundary is

$$\lambda T = e^{j\theta} - 1 = \cos\theta + j\sin\theta - 1 = (\cos\theta - 1) + j\sin\theta \quad (8.150)$$

If we let x and y be the real and imaginary parts of λT , respectively, that is, $\lambda T = x + jy$, then

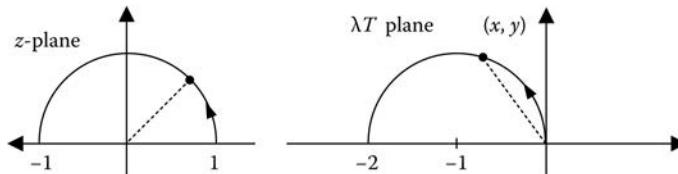


FIGURE 8.22 Variation of z -plane pole for determining AB-1 stability boundary.

$$x = \cos\theta - 1, y = \sin\theta \quad (8.151)$$

$$\Rightarrow (x+1)^2 + y^2 = \cos^2\theta + \sin^2\theta = 1 \quad (8.152)$$

and the AB-1 stability boundary is therefore a circle with center at $(-1, 0)$ and radius 1 in the λT or $x-y$ plane.

AB-1 integration is inappropriate for simulation of an undamped second-order system, a result we observed earlier in Section 3.6. The characteristic roots of a second-order system with $\zeta = 0$ are $\lambda = \pm j\omega_n$, and hence $\lambda T = \pm j\omega_n T$, which corresponds to the imaginary axis in the λT plane. From [Figure 8.21](#), it is clear that the imaginary axis lies outside the AB-1 stability region (except for the origin).

A more general approach to locating stability boundaries is to view them as locus of points in the λT plane resulting from a mapping of the Unit Circle in the z -plane. [Figure 8.23](#) illustrates the AB-2 stability boundary resulting from mapping points $z = re^{j\theta} = 1e^{j\theta}$ ($0 \leq \theta < 2\pi$) along the Unit Circle in the z -plane according to the transformation

$$\lambda T = 2 \left(\frac{e^{j\theta} - e^{j2\theta}}{1 - 3e^{j\theta}} \right) \quad (8.153)$$

obtained by solving for λT in the denominator of Equation 8.146 with z replaced by $e^{j\theta}$. The stability boundary (polar form $\lambda T = Me^{j\psi}$) is shown in [Figure 8.23](#).

Polar coordinates of the points along the AB-2 stability boundary are

$$M = 2 \left| \frac{e^{j\theta} - e^{j2\theta}}{1 - 3e^{j\theta}} \right| \quad (8.154)$$

$$= 2 \left[\frac{(\cos\theta - \cos 2\theta)^2 + (\sin\theta - \sin 2\theta)^2}{(1 - 3\cos\theta)^2 + (-3\sin\theta)^2} \right]^{1/2} \quad (8.155)$$

$$= 2 \left(\frac{1 - \cos\theta}{5 - 3\cos\theta} \right)^{1/2} \quad (8.156)$$

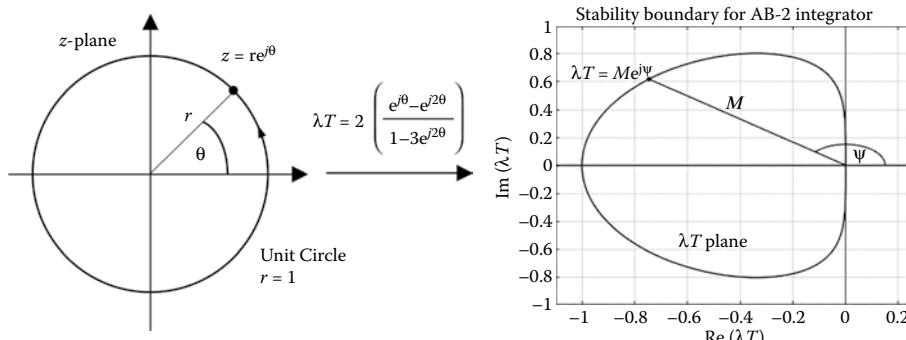


FIGURE 8.23 Mapping the Unit Circle to the AB-2 stability boundary.

$$\psi = \text{Arg} \left(\frac{e^{j\theta} - e^{j2\theta}}{1 - 3e^{j\theta}} \right) \quad (8.157)$$

$$= \tan^{-1} \left(\frac{4 \sin \theta - \sin 2\theta}{4 \cos \theta - \cos 2\theta - 3} \right) \quad (8.158)$$

Rectangular coordinates of λT on the AB-2 stability boundary are given in Exercise 8.30.

EXAMPLE 8.4

For AB-2 integration,

- a. Find the image points on the AB-2 stability boundary in the λT plane of the following points:

$$z = 1, \left(\frac{\sqrt{2}}{2} \right)(1+j), j, -1, e^{j4\pi/3}, -j, \left(\frac{\sqrt{2}}{2} \right)(1-j).$$

- b. Is it possible for an AB-2 simulation of an undamped second-order system to be stable? Verify the result.

- a. The image points are computed using Equations 8.156 and 8.158 in the M-file “Ch8_Ex8_4.m.” They are tabulated in [Table 8.5](#).
b. An undamped second-order system is governed by

$$\frac{d^2}{dt^2} x(t) + \omega_n^2 x(t) = u(t) \quad (8.159)$$

The characteristic roots are located on the imaginary axis at $\lambda = \pm j\omega_n$. Close-ups of the AB-2 stability boundary near the imaginary axis are shown in [Figure 8.24](#). Observation of the left graph in [Figure 8.24](#) implies $\lambda T = j\omega_n T$ is limited to approximately $j0.12$ for the AB-2 integrator to result in a stable discrete-time system. However, a closer look at the AB-2 stability region in the right graph indicates that the limit is considerably smaller. Further enlargement of the AB-2 stability region in the vicinity of the origin will show that the imaginary axis is exterior to the AB-2 stability region (with the exception of the origin, $\lambda T = 0$).

TABLE 8.5
Points on Unit Circle and Image Points on AB-2 Stability Boundary

$r = re^{j\theta}$	$z = a + jb$	$\lambda T = Me^{j\psi}$	$\lambda T = c + jd$
r, θ	a, b	M, ψ	c, d
1, 0	1, 0	0, 0	0, 0
1, $\pi/4$	$\sqrt{2}/2, \sqrt{2}/2$	0.6380, $\pi/4$	-0.0596, 0.6352
1, $\pi/2$	0, 1	0.8944, $\pi/2$	-0.4, 0.8
1, π	-1, 0	1, π	-1, 0
1, $4\pi/3$	$-1/2, -\sqrt{3}/2$	0.9608, -2.0944	-0.6923, -0.6662
1, $3\pi/2$	0, -1	0.8944, $-\pi/2$	-0.4, -0.8
1, $7\pi/8$	$\sqrt{2}/2, -\sqrt{2}/2$	0.6380, $-\pi/4$	-0.0596, -0.6352

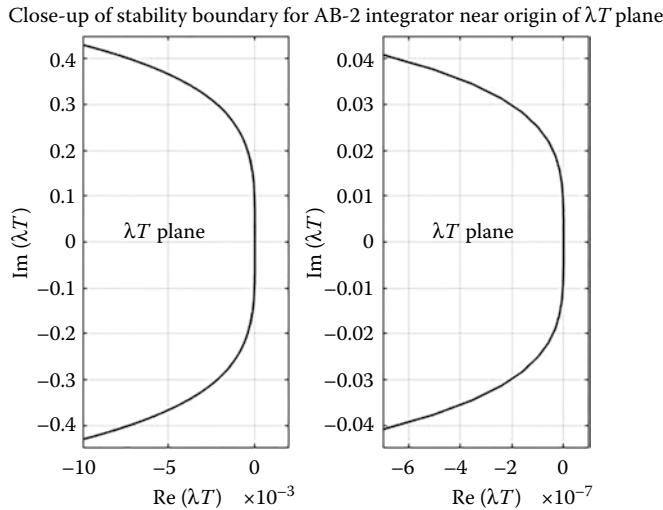


FIGURE 8.24 Close-ups of AB-2 stability boundary near origin of λT plane.

The instability of an AB-2 integrator for simulation of an undamped second-order system can be established by investigating the characteristic polynomial of the z-domain transfer function $H(z)$ given by

$$H(z) = \frac{1}{s^2 + \omega_n^2} \Big|_{s \leftarrow -(2/T)[z(z-1)/(3z-1)]} \quad (8.160)$$

$$\Rightarrow \frac{X(z)}{U(z)} = \frac{0.25T^2(9z^2 - 6z + 1)}{z^4 - 2z^3 + [1 + 2.25(\omega_n T)^2]z^2 - 1.5(\omega_n T)^2z + 0.25(\omega_n T)^2} \quad (8.161)$$

[Figure 8.25](#) is a plot of the loci of the four poles of $H(z)$ corresponding to numerical values of $\omega_n T = 0.05, 0.01, 0.15, \dots, 0.95, 1$. The AB-2 integrator generates a discrete-time output $x(n)$ from the fourth-order system governed by

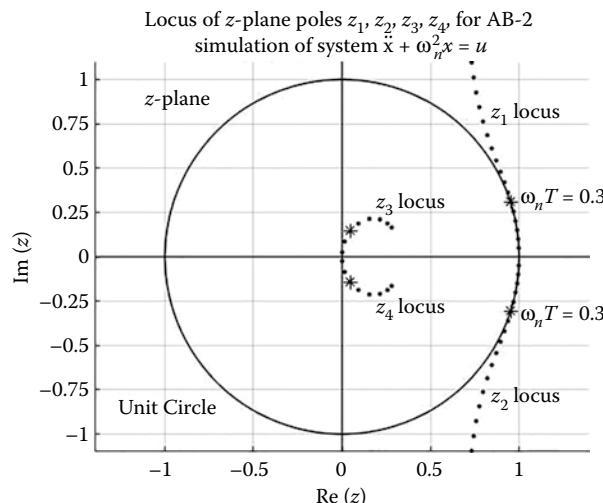


FIGURE 8.25 Locus of poles of $H(z)$ for AB-2 simulation of undamped second-order system $\ddot{x} + \omega_n^2 x = u$.

$$\begin{aligned} x(n+4) - 2x(n+3) + [1 + 2.25(\omega_n T)^2]x(n+2) - 1.5(\omega_n T)^2 x(n+1) \\ + 0.25(\omega_n T)^2 x(n) = 0.25T^2[9u(n+2) - 6u(n+1) + u(n)] \end{aligned} \quad (8.162)$$

Up until $\omega_n T \approx 0.3$, there is a pair of equivalent complex roots that die out rapidly due to the close proximity to the origin of the extraneous poles z_3 and z_4 . At the same time, z_1 and z_2 appear to lie on the Unit Circle implying that the other pair of equivalent, continuous-time poles (corresponding to poles z_1 and z_2) lie on the imaginary axis in the s -plane.

If z_1 and z_2 were actually on the Unit Circle, the damping ratio of the equivalent second-order continuous-time system would be zero and the discrete-time output would reflect an undamped second-order system response once the fast transient component vanishes. In reality, all the points along the two loci shown in Figure 8.25 are outside the Unit Circle, and the equivalent continuous-time second-order system damping ratio is slightly negative (see Exercise 8.20).

Howe (1986) includes asymptotic formulas for natural frequency and damping ratio errors incurred when using low-order multistep Adams–Bashforth integration methods. The formulas for $k = 1, 2, 3, 4$ are

$$k = 1: e_{\omega_n} \approx \zeta e/\omega_n T, \quad e_\zeta \approx (\zeta^2 - 1)e_I/\omega_n T, \quad \omega_n T \ll 1 \quad (8.163)$$

$$k = 2: e_{\omega_n} \approx (1 - 2\zeta^2)e_I(\omega_n T)^2, \quad e_\zeta \approx 2(\zeta - \zeta^3)e_I(\omega_n T)^2, \quad \omega_n T \ll 1 \quad (8.164)$$

$$k = 3: e_{\omega_n} \approx -(3\zeta - 4\zeta^3)e_I(\omega_n T)^3, \quad e_\zeta \approx (1 - 5\zeta^2 + 4\zeta^4)e_I(\omega_n T)^3, \quad \omega_n T \ll 1 \quad (8.165)$$

$$k = 4: e_{\omega_n} \approx -(1 - 8\zeta^2 + 8\zeta^4)e_I(\omega_n T)^4, \quad e_\zeta \approx -4(\zeta - 3\zeta^3 + 2\zeta^5)e_I(\omega_n T)^4, \quad \omega_n T \ll 1 \quad (8.166)$$

where e_I are the integration error coefficients given in Section 8.2. For accurate ($\omega_n T \ll 1$) simulations of undamped ($\zeta = 0$) second-order continuous-time systems, Equations 8.163 and 8.165 imply damping ratio errors of $-1/2(\omega_n T)$ with AB-1 and $3/8(\omega_n T)^3$ with AB-3 integration, respectively. Equations 8.164 and 8.166 imply the damping ratio error is zero for AB-2 and AB-4 integration to order $O(\omega_n T)^2$ and $O(\omega_n T)^4$, respectively. The actual damping ratio error is of order $O(\omega_n T)^3$ for AB-2 and $O(\omega_n T)^5$ for AB-4 integration.

8.3.2 IMPLICIT INTEGRATORS

The Adams–Moulton implicit integrators were introduced in Section 6.4. The z -domain transfer functions for AM-2, AM-3, and AM-4 integrators are obtained in a similar fashion to the Adams–Bashforth integrators, that is, the difference equation approximation of a pure continuous-time integrator is developed and then z -transformed to produce $H_I(z)$. The results for AM-2 through AM-4 integrators are given below (see Exercise 8.22).

$$\text{AM-2: } H_I(z) = \frac{T}{2} \left(\frac{z+1}{z-1} \right) \quad (8.167)$$

$$\text{AM-3: } H_I(z) = \frac{T}{12} \left[\frac{5z^2 + 8z - 1}{z(z-1)} \right] \quad (8.168)$$

$$\text{AM-4: } H_I(z) = \frac{T}{24} \left[\frac{9z^3 + 19z^2 - 5z + 1}{z^2(z-1)} \right] \quad (8.169)$$

Replacing s by $1/H_j(z)$ in the first-order system transfer function $H(s) = 1/(s - \lambda)$ leads to the following expressions for the z -domain transfer functions using AM-2 through AM-4 integration,

$$\text{AM-2: } H(z) = \frac{T(z+1)}{(2 - \lambda T)z - (2 + \lambda T)} \quad (8.170)$$

$$\text{AM-3: } H(z) = \frac{T(5z^2 + 8z - 1)}{(12 - 5\lambda T)z^2 - (12 + 8\lambda T)z + \lambda T} \quad (8.171)$$

$$\text{AM-4: } H(z) = \frac{T(9z^3 + 19z^2 - 5z + 1)}{(24 - 9\lambda T)z^3 - (24 + 19\lambda T)z^2 + 5\lambda Tz - \lambda T} \quad (8.172)$$

We may conclude from Equations 8.170 through 8.172 that AM-2 integration does not introduce extraneous roots (system poles), whereas AM-3 and AM-4 introduce one and two extraneous roots for each state. Stability boundaries for AM-2, AM-3, and AM-4 integration are obtained using the same method for the Adams–Bashforth integrators. Starting with AM-2, the characteristic polynomial for $H(z)$ in Equation 8.170 is

$$(2 - \lambda T)z - (2 + \lambda T) = 0 \quad (8.173)$$

Setting $z = e^{j\theta}$ and solving for λT yield

$$\lambda T = 2 \left(\frac{z+1}{z-1} \right) \Big|_{z \leftarrow e^{j\theta}} = 2 \left(\frac{e^{j\theta} - 1}{e^{j\theta} + 1} \right) \cdot \left(\frac{e^{-j\theta/2}}{e^{-j\theta/2}} \right) \quad (8.174)$$

$$= 2 \left(\frac{e^{j\theta/2} - e^{-j\theta/2}}{e^{j\theta/2} + e^{-j\theta/2}} \right) \quad (8.175)$$

$$= j2 \left[\frac{\sin(\theta/2)}{\cos(\theta/2)} \right] \quad (8.176)$$

$$= j2 \tan(\theta/2) \quad (8.177)$$

From Equation 8.177, the top half of the Unit Circle, that is, $z = e^{j\theta}$, $0 \leq \theta < \pi$, is mapped into the imaginary axis from $\lambda T = 0$ to $\lambda T = j\infty$. The entire Unit Circle is mapped into the imaginary axis in the λT plane, which is the stability boundary for AM-2 or trapezoidal integration. In other words, the entire left-half plane is the stability region assuring that any stable continuous-time system ($\text{Re } \lambda < 0$) simulated by AM-2 integration leads to a stable discrete-time system regardless of the integration step size.

The stability regions for AM-3 and AM-4 integration are obtained from mapping the Unit Circle according to

$$\text{AM-3: } \lambda T = 12 \left(\frac{e^{j2\theta} - e^{j\theta}}{5e^{j2\theta} + 8e^{j\theta} - 1} \right) \quad (8.178)$$

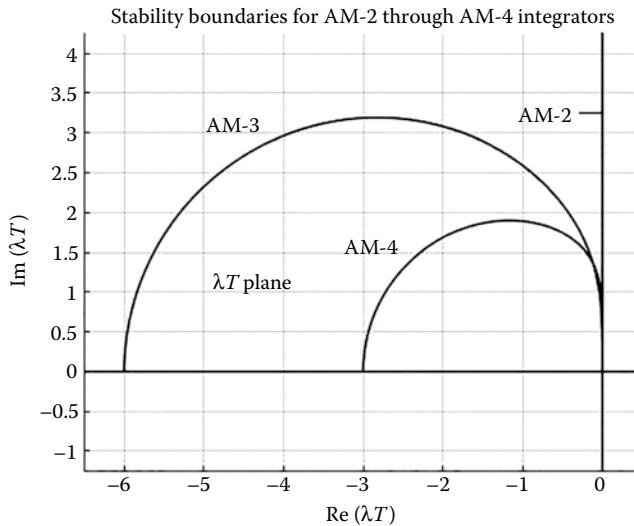


FIGURE 8.26 Stability boundaries for AM-2 through AM-4 integrators.

$$\text{AM-4: } \lambda T = 24 \left(\frac{e^{j3\theta} - e^{j2\theta}}{9e^{j3\theta} + 19e^{j2\theta} - 5e^{j\theta} + 1} \right) \quad (8.179)$$

The stability boundaries for AM-3 and AM-4 integration are computed in “Ch8_AM_Stability_Boundaries.m” and shown along with the AM-2 stability boundary in Figure 8.26.

Note the restrictions imposed on AM-3 and AM-4 simulation of a stable, first-order system ($\lambda < 0$). The integration step size T is limited to less than $-6/\lambda$ and $-3/\lambda$, respectively. Equivalently, the step size $T < 6\tau$ for AM-3 and $T < 3\tau$ for AM-4 integration, where $\tau = -1/\lambda$ is the system time constant.

EXAMPLE 8.5

The concentration of a chemical in the vessel shown in Figure 8.27 is determined by the differential equation

$$\frac{V}{Q} \frac{dc}{dt} + c = c_1 \quad (8.180)$$

where

V is the constant volume of liquid in the vessel

Q is the constant flow rate of liquid in and out of the vessel

c_1 is the concentration of chemical in the liquid flowing in

c is the concentration of the chemical in the well-stirred vessel

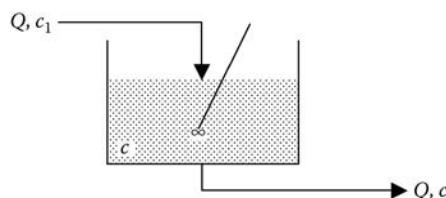


FIGURE 8.27 Chemical flowing in and out of a vessel with constant liquid volume.

- Find the analytical solution for the concentration $c(t)$, $t \geq 0$ when the input $c_1(t) = \bar{c}_1$, $t \geq 0$.
- Find the difference equation for simulating the concentration response using AM-2 integration with step size T .
- Find an expression for the steady-state value $c(n)|_{n \rightarrow \infty}$ and compare it with $c(t)|_{t \rightarrow \infty}$.
- Repeat parts (b) and (c) for AM-3 integration.
- Numerical values of the system parameters are $Q = 25 \text{ m}^3/\text{min}$ and $V = 150 \text{ m}^3$, and the initial concentration of chemical in the tank is $c(0) = 5 \text{ mg/m}^3$. The input $\bar{c}_1 = 60 \text{ mg/m}^3$. Simulate the concentration response using AM-2 and AM-3 integration with step size $T = 1.5\tau$, 1.25τ , τ , 0.75τ where $\tau = V/Q$ is the system time constant. Plot and compare the simulated responses and the analytical solution.
- The analytical solution is obtained by Laplace transforming the differential equation with input c_1 constant along with the given initial condition. Alternatively, the step response of a first-order system is given in Equation 2.6 and repeated as follows using the current notation for the stirred tank.

$$c(t) = c(0)e^{-(Q/V)t} + c_1[1 - e^{-(Q/V)t}], t \geq 0 \quad (8.181)$$

- Rewriting the differential equation as

$$\frac{dc}{dt} = -\frac{Q}{V}c + \frac{Q}{V}c_1 = -\frac{1}{\tau}c + \frac{1}{\tau}c_1 \quad (8.182)$$

and comparing it with $dx/dt = \lambda x + u$, the z-domain transfer function of the discrete-time system is obtained by replacing λ with $-1/\tau$ in Equation 8.170 and inserting $1/\tau$ in the numerator to give

$$H(z) = \frac{C(z)}{C_1(z)} = \frac{(T/\tau)(z+1)}{[2+(T/\tau)]z-[2-(T/\tau)]} \quad (8.183)$$

$$= \frac{(T/2\tau)(z+1)}{[1+(T/2\tau)]z-[1-(T/2\tau)]} \quad (8.184)$$

Inverting Equation 8.184 leads to the difference equation

$$\left(1 + \frac{T}{2\tau}\right)c(n+1) - \left(1 - \frac{T}{2\tau}\right)c(n) = \frac{T}{2\tau}[c_1(n+1) + c_1(n)], \quad n = 0, 1, 2, \dots \quad (8.185)$$

which is used to update the state according to

$$c(n+1) = \left[\frac{1-(T/2\tau)}{1+(T/2\tau)}\right]c(n) + \left[\frac{T/\tau}{1+(T/2\tau)}\right]\bar{c}_1, \quad n = 0, 1, 2, \dots \quad (8.186)$$

- The steady-state value $c(n)|_{n \rightarrow \infty}$ is obtained from Equation 8.186 after replacing $c(n)$ and $c(n+1)$ with $c(n)|_{n \rightarrow \infty}$. Solving for $c(n)|_{n \rightarrow \infty}$ yields

$$c(n)|_{n \rightarrow \infty} = \bar{c}_1 \quad (8.187)$$

Hence, $c(n)|_{n \rightarrow \infty} = \bar{c}_1 = c(t)|_{t \rightarrow \infty}$ the final concentration of the continuous-time system.

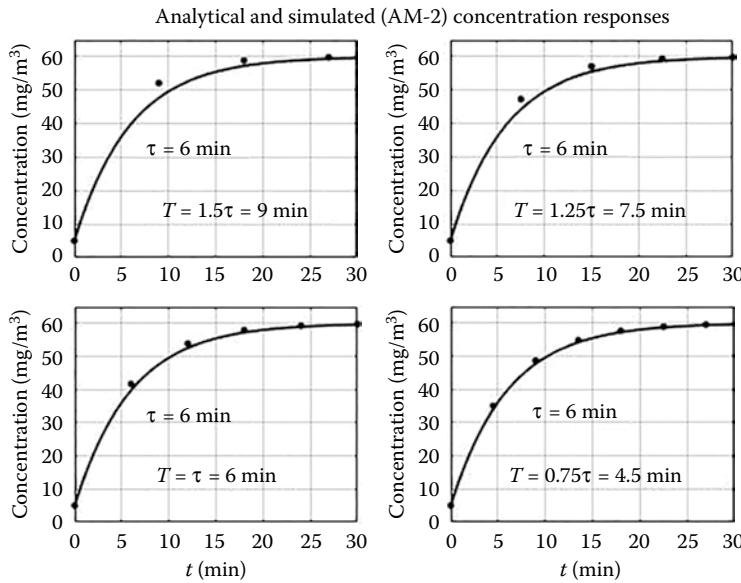


FIGURE 8.28 Analytical and simulated AM-2 concentration response.

d. Inserting $1/\tau$ in the numerator of Equation 8.171, and simplifying the result give

$$H(z) = \frac{C(z)}{C_1(z)} = \frac{5z^2 + 8z - 1}{[12(\tau/T) + 5]z^2 - [12(\tau/T) - 8]z - 1} \quad (8.188)$$

and the difference equation is

$$\left(12\frac{\tau}{T} + 5\right)c(n+2) - \left(12\frac{\tau}{T} - 8\right)c(n+1) - c(n) = 5c_1(n+2) + 8c_1(n+1) - c_1(n) \quad (8.189)$$

$$\Rightarrow c(n+2) = \frac{1}{[12(\tau/T) + 5]} \left[\left(12\frac{\tau}{T} - 8\right)c(n+1) + c(n) + 12\bar{c}_1 \right], \quad n = 0, 1, 2, \dots \quad (8.190)$$

Letting $c(n+2) = c(n+1) = c(n) = c(n)|_{n \rightarrow \infty}$ in Equation 8.190 and solving for $c(n)|_{n \rightarrow \infty}$, give the same result as Equation 8.187, that is, the AM-3 integrator also converges to $c(t)|_{n \rightarrow \infty} = \bar{c}_1$.

e. The simulated responses using AM-2 and AM-3 integration along with the analytical solution are computed in “Ch8_Ex8_5.m” and shown in Figures 8.28 and 8.29. An RK-3 integrator would normally be used to generate $c(1)$, which is required to compute $c(2)$ in Equation 8.190. However, the exact value $c(T)$ was used instead.

Note the improvement in the AM-3 integrator compared with the AM-2 integrator.

8.3.3 RUNGAKUTTA (RK) INTEGRATION

RK numerical integration was introduced in Section 6.2. Unlike the multistep methods, RK integration algorithms are referred to as single pass or one step in nature. Depending on the order of

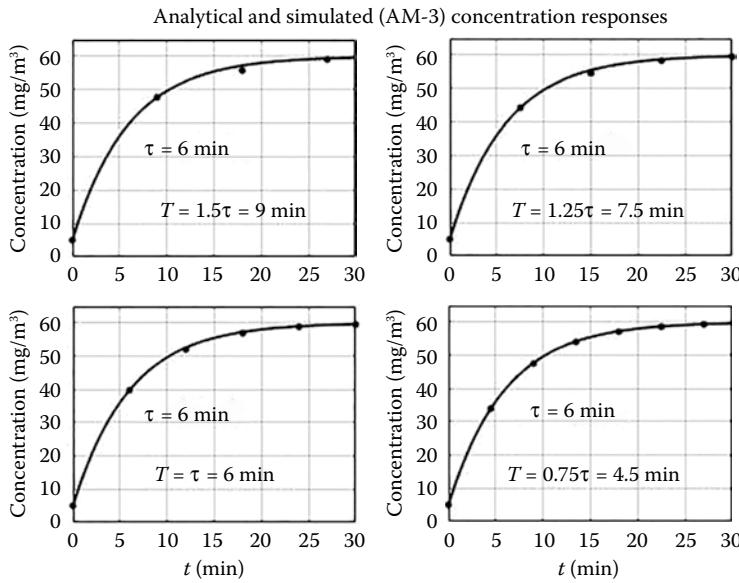


FIGURE 8.29 Analytical and simulated AM-3 concentration response.

the RK integrator, one or more state derivative function evaluations are required per step in order to advance the discrete-time state approximation to the next step. Fixed-step and variable-step RK formulas are popular in continuous-time system simulation.

Numerical stability with fixed-step RK integrators is important because of the limitations imposed on the integration step size. A similar approach to the one used for multistep methods is employed to obtain the stability boundary corresponding to a particular RK integrator. To illustrate, consider the second-order RK-2 integrator first introduced in Section 3.6 known as improved Euler or Heun's method. A continuous-time first-order system modeled by $\frac{dx}{dt} = f(x, u) = \lambda x + u$ is simulated using improved Euler integration by first predicting the updated state as

$$\hat{x}(n+1) = x(n) + Tf[x(n), u(n)] \quad (8.191)$$

$$= x(n) + T[\lambda x(n) + u(n)] \quad (8.192)$$

$$= (1 + \lambda T)x(n) + Tu(n) \quad (8.193)$$

followed by correction to

$$x(n+1) = x(n) + \frac{T}{2}\{f[x(n), u(n)] + f[\hat{x}(n+1), u(n+1)]\} \quad (8.194)$$

$$= x(n) + \frac{T}{2}\{f[x(n), u(n)] + f[1 + \lambda T)x(n) + Tu(n), u(n+1)]\} \quad (8.195)$$

$$= x(n) + \frac{T}{2}\{\lambda x(n) + u(n) + \lambda[1 + \lambda T)x(n) + Tu(n)] + u(n+1)\} \quad (8.196)$$

$$= \left[1 + \lambda T + \frac{(\lambda T)^2}{2} \right] x(n) + \frac{T}{2} [(1 + \lambda T) u(n) + u(n+1)] \quad (8.197)$$

Taking the z -transform of Equation 8.197 and solving for the ratio $X(z)/U(z)$ give

$$H(z) = \frac{X(z)}{U(z)} = \frac{(T/2)(z + \lambda T + 1)}{z - [1 + \lambda T + (\lambda T)^2/2]} \quad (8.198)$$

Another popular RK-2 integrator, first introduced in Section 3.6, is the modified Euler integrator. The difference equation for modified Euler integration with a step size T can be obtained by reference to Figure 8.30. Note that the intervals of width $\hat{T} = T/2$ correspond to one-half the basic simulation frame rate ($1/T$) to accommodate the input sampling rate of two samples per integration step T .

The first step in advancing the state using modified Euler integration with step size T is to compute the value $\hat{x}(n+1)$ halfway through the integration interval, that is,

$$\hat{x}(n \pm 1) = x(n) + \hat{T}f[x(n), u(n)] \quad (8.199)$$

$$= x(n) + \hat{T}[\lambda x(n) + u(n)] \quad (8.200)$$

$$= (1 + \lambda \hat{T})x(n) + \hat{T}u(n) \quad (8.201)$$

The derivative function at $t = (n+1)\hat{T}$ is calculated using the predicted value $\hat{x}(n+1)$ in Equation 8.201. The updated state $x(n+2)$ is computed by taking a step of length $T = 2\hat{T}$ in the direction based on the midpoint derivative. Thus,

$$x(n+2) = x(n) + 2\hat{T}f[\hat{x}(n+1), u(n+1)] \quad (8.202)$$

$$= x(n) + 2\hat{T}[\lambda \hat{x}(n+1) + u(n+1)] \quad (8.203)$$

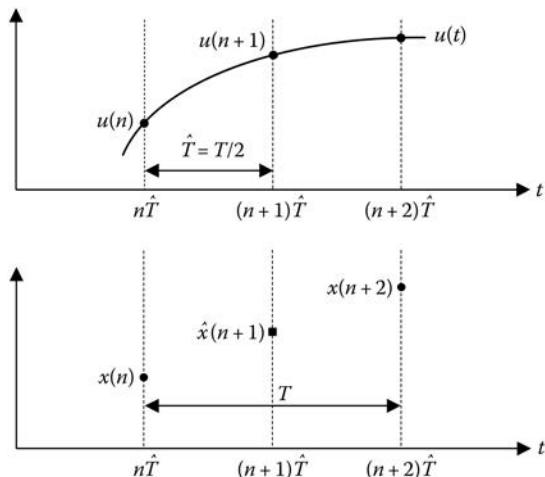


FIGURE 8.30 Modified Euler integration running at state update rate ($1/T$).

$$= x(n) + 2\lambda \hat{T} \hat{x}(n+1) + 2\hat{T} u(n+1) \quad (8.204)$$

$$= x(n) + 2\lambda \hat{T} [(1 + \lambda \hat{T})x(n) + \hat{T} u(n)] + 2\hat{T} u(n+1) \quad (8.205)$$

$$= [1 + 2\lambda \hat{T} (1 + \lambda \hat{T})]x(n) + 2\hat{T} [\lambda \hat{T} u(n) + u(n+1)] \quad (8.206)$$

In terms of the modified RK-2 integration step size $T = 2\hat{T}$, the difference equation for updating the discrete-time state $x(n)$ is

$$x(n+2) = \left[1 + \lambda T + \frac{(\lambda T)^2}{2} \right] x(n) + T \left[\frac{\lambda T}{2} u(n) + u(n+1) \right], \quad n = 0, 1, 2, 3, 4, \dots \quad (8.207)$$

Note that $n = 0, 1, 2, 3, 4, \dots$ in Equation 8.207 corresponds to times $0, T/2, T, 3T/2, 2T, \dots$, and, therefore, $x(n), n = 0, 2, 4, \dots$ are the modified RK-2 states updated every T (s). The z -domain transfer function for modified RK-2 integration with step T is obtained by z -transforming Equation 8.207,

$$H(z) = \frac{T[z + (\lambda T/2)]}{z^2 - [1 + \lambda T + (\lambda T)^2/2]} \quad (8.208)$$

Difference equations and z -domain transfer functions for higher-order RK integrators are obtained in a similar fashion to the procedure outlined in Equations 8.199 through 8.208 for modified RK-2 integration. An RK-3 integrator with step size T requiring input samples at the beginning, one-third and two-thirds into the interval, is described by

$$k_1 = f[x(n), u(n)] \quad (8.209)$$

$$k_2 = f \left[x(n) + \frac{T}{3} k_1, u \left(n + \frac{1}{3} \right) \right] \quad (8.210)$$

$$k_3 = f \left[x(n) + \frac{2T}{3} k_2, u \left(n + \frac{2}{3} \right) \right] \quad (8.211)$$

$$x(n+1) = x(n) + \frac{T}{4} (k_1 + 3k_3) \quad (8.212)$$

Using this RK-3 integrator with a sampling interval $\hat{T} = T/3$ to simulate the first-order continuous-time system $dx/dt = \lambda x + u$ results in the third-order difference equation (see Exercise 8.27)

$$\begin{aligned} x(n+3) &= \left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} \right] x(n) + \left[\frac{T}{4} + \frac{\lambda^2 T^3}{6} \right] u(n) \\ &\quad + \frac{\lambda T^2}{2} u(n+1) + \frac{3T}{4} u(n+2), \quad n = 0, 1, 2, 3, \dots \end{aligned} \quad (8.213)$$

where $x(n), n = 0, 3, 6, 9, \dots$ are the RK-3 states updated once every T (s).

z -Transforming Equation 8.213 leads to the z -domain transfer function

$$H(z) = \frac{(3T/4)z^2 + (\lambda T^2/2)z + (T/4) + (\lambda^2 T^3/6)}{z^3 - [1 + \lambda T + (\lambda T)^2/2 + (\lambda T)^3/6]} \quad (8.214)$$

Consider the RK-4 integrator presented in Section 6.2, Equations 8.60 through 8.64 with integration step size T and input sampled at the beginning and midpoint of each interval. The z -domain transfer function is (Howe 1986, 1995)

$$H(z) = \frac{(T/6)\{z^2 + [4 + 2\lambda T^2 + (\lambda T)^2/2]z + 1 + \lambda T + (\lambda T)^2/2 + (\lambda T)^3/4\}}{z^2 - [1 + \lambda T + (\lambda T)^2/2 + (\lambda T)^3/6 + (\lambda T)^4/24]} \quad (8.215)$$

The characteristic polynomials for the one-step RK integrators with z -domain transfer functions given in Equations 8.198, 8.208, 8.214, and 8.215 are summarized as follows.

$$\text{RK-2 (Improved Euler): } z - \left[1 + \lambda T + \frac{(\lambda T)^2}{2} \right] \quad (8.216)$$

$$\text{RK-2 (Modified Euler): } z^2 - \left[1 + \lambda T + \frac{(\lambda T)^2}{2} \right] \quad (8.217)$$

$$\text{RK-3 (Input sampling at } 3/T\text{): } z^3 - \left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} \right] \quad (8.218)$$

$$\text{RK-4 (Input sampling at } 2/T\text{): } z^2 - \left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} + \frac{(\lambda T)^4}{24} \right] \quad (8.219)$$

For an m th-order RK integrator requiring k_s input samples per integration step T , the characteristic polynomial is given by

$$\text{RK-}m \left(\text{Input sampling at } \frac{k_s}{T} \right): z^{k_s} - \left[1 + \lambda T + \frac{(\lambda T)^2}{2!} + \frac{(\lambda T)^3}{3!} + \cdots + \frac{(\lambda T)^m}{m!} \right] \quad (8.220)$$

Note that the bracketed expression in Equation 8.220 is the truncated Taylor Series approximation for $e^{\lambda T}$. Let us explore this point further. λ^* , the characteristic root of the equivalent continuous-time system, is related to the z -plane pole by

$$z = e^{\lambda^*(T/k_s)} \quad (8.221)$$

The z -plane pole for the RK-4 integrator is from Equation 8.219

$$z = \left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} + \frac{(\lambda T)^4}{24} \right]^{1/2} \quad (8.222)$$

Substituting this z into Equation 8.221 with $k_s = 2$ and squaring both sides lead to

$$\left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} + \frac{(\lambda T)^4}{24}\right] = e^{\lambda^* T} \quad (8.223)$$

Expanding the exponential term in Equation 8.223 in a fifth-order truncated power series eventually leads to the asymptotic formula for the fractional characteristic root error, that is,

$$\text{RK-4: } e_\lambda = \frac{\lambda^*}{\lambda} - 1 \approx -\frac{1}{120}(\lambda T)^4, \quad |\lambda T| \ll 1 \quad (8.224)$$

which implies the integrator error coefficient e_i for RK-4 is $-1/120$.

EXAMPLE 8.6

Find the equivalent continuous-time system characteristic root for the system in Example 8.5 using RK-2, RK-3, and RK-4 integration with step size $T = 0.25$ s.

From Equation 8.223 and similar expressions for RK-2 and RK-3,

$$\text{RK-2: } \lambda^* = \frac{1}{T} \ln \left[1 + \lambda T + \frac{(\lambda T)^2}{2} \right] \quad (8.225)$$

$$\text{RK-3: } \lambda^* = \frac{1}{T} \ln \left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} \right] \quad (8.226)$$

$$\text{RK-4: } \lambda^* = \frac{1}{T} \ln \left[1 + \lambda T + \frac{(\lambda T)^2}{2} + \frac{(\lambda T)^3}{6} + \frac{(\lambda T)^4}{24} \right] \quad (8.227)$$

The characteristic root for the system in Example 8.5 is $\lambda = -1/6$. Substituting $\lambda T = (-1/6)(1/4) = -1/24$ in Equations 8.225 through 8.227 results in

$$\lambda^* = \begin{cases} -0.16661691, & (\text{RK-2}) \\ -0.16666719, & (\text{RK-3}) \\ -0.16666666, & (\text{RK-4}) \end{cases}$$

The stability boundaries for the RK integrators are obtained as before by mapping the Unit Circle in the z -plane into the λT plane using the denominator of $H(z)$ to define the mapping. The MATLAB M-file “Ch8_RK_Stability_Boundaries.m” finds and plots the top half of the RK-2, RK-3, and RK-4 stability boundaries shown in Figure 8.31.

Unlike the Adams–Bashforth and Adams–Moulton integrators, the stability regions become larger for the higher-order RK integrators. There is a single stability boundary for all m th-order RK integrators, independent of the number of input samples required during each integration step. This is logical since stability of the discrete-time system associated with RK integration is an inherent system property unrelated to the possible existence of inputs.

The fractional characteristic root error e_λ for the m th-order numerical integrators discussed in this section and the previous section is related to the integrator error coefficient e_i according to (Howe 1995)

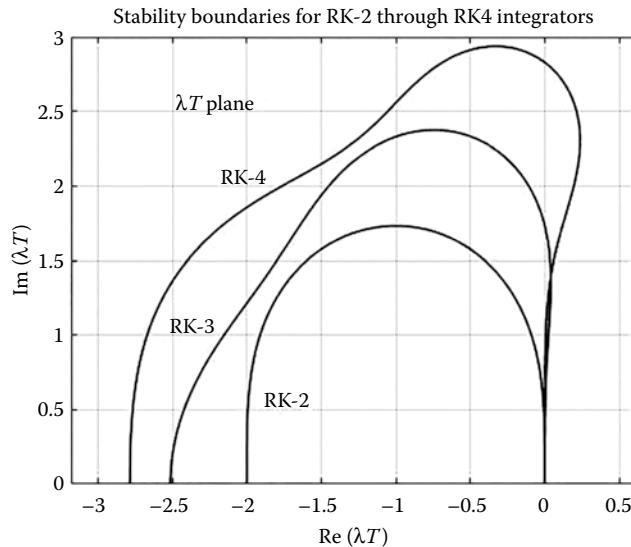


FIGURE 8.31 Stability boundaries for RK-2 through RK-4 integrators.

$$e_\lambda = \frac{\lambda^*}{\lambda} - 1 \approx -e_l(\lambda T)^m, \quad |\lambda T| \ll 1 \quad (8.228)$$

A comparison of characteristic root errors for comparable order, Adams–Bashforth, Adams–Moulton, and RK integrators, is shown in the middle three columns of Table 8.6.

Keep in mind the RK- m integrator requires m derivative function evaluations per step. The RK-4 integrator, for example, would take roughly four times longer than either AB-4 or AM-4 integrators to execute a single step. In order to keep the computational effort between the multistep AB- m and AM- m integrators comparable to the one-step RK- m integrators, the step size should be m times larger with RK- m integration.

The last column in Table 8.6 reflects the effect of increasing the step size with RK integration to make the computational effort approximately the same as the comparable order AB and AM integrators. In the case of RK-4, the effective characteristic root error \tilde{e}_λ is proportional to $-(256/120)(\lambda T)^4$, and the ratio of e_λ for AB-4 integration to \tilde{e}_λ for RK-4 integration is

$$\frac{e_\lambda}{\tilde{e}_\lambda} = \frac{-\frac{251}{720}(\lambda T)^4}{-\frac{256}{120}(\lambda T)^4} = 0.1634 \quad (8.229)$$

TABLE 8.6
Characteristic Root Errors for AB, AM, and RK Integrators

m	e_λ , AB- m Step Size T	e_λ , AM- m Step Size T	e_λ , RK- m Step Size T	\bar{e}_λ , RK- m Step Size T
2	$-\frac{5}{12}(\lambda T)^2$	$\frac{1}{12}(\lambda T)^2$	$-\frac{1}{6}(\lambda T)^2$	$-\frac{1}{6}(\lambda 2T)^2 = -\frac{4}{6}(\lambda T)^2$
3	$-\frac{3}{8}(\lambda T)^3$	$\frac{1}{24}(\lambda T)^3$	$-\frac{1}{24}(\lambda T)^3$	$-\frac{1}{24}(\lambda 3T)^3 = -\frac{27}{24}(\lambda T)^3$
4	$-\frac{251}{720}(\lambda T)^4$	$\frac{19}{720}(\lambda T)^4$	$-\frac{1}{120}(\lambda T)^4$	$-\frac{1}{120}(\lambda 4T)^4 = -\frac{256}{120}(\lambda T)^4$

making AB-4 integration roughly six times more accurate than RK-4 integration when execution time is taken into consideration.

EXAMPLE 8.7

A simplified block diagram for the forward speed control of a ground vehicle is shown in Figure 8.32. The system parameters are the open-loop system gain K and poles located at $s = -a$ and $s = -b$.

- Find expressions for the natural frequency, damping ratio, and steady-state gain of the second-order closed-loop system in terms of the system parameters.
- Find the analytical solution for the unit step response.
- An RK-2 (improved Euler) simulation is performed for the cases where $K = 100, 250$ using step sizes of $T = 0.025$ and 0.1 s. The open-loop poles are located at $s = -a = -2$ s $^{-1}$ and $s = -b = -5$ s $^{-1}$. Plot the analytical and simulated step responses for each case on separate graphs. Comment on the accuracy and numerical stability of the RK-2 integrator. Repeat using RK-4 integration with step sizes of 0.1 and 0.2 s.
- For the case where $K = 250$, $a = 2$, and $b = 5$, find the maximum value of T that can be used to implement RK-3 simulation. Verify the result.

- The closed-loop system transfer function is

$$\frac{V(s)}{V_{com}(s)} = \frac{K}{s^2 + (a+b)s + ab + K} \quad (8.230)$$

Comparing Equation 8.230 to the standard form of a second-order system transfer function

$$\frac{K}{s^2 + (a+b)s + ab + K} = \frac{K_{ss}\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8.231)$$

and solving for the second-order system parameters K_{ss} , ω_n , and ζ results in

$$K_{ss} = \frac{K}{ab + K}, \quad \omega_n = (ab + K)^{1/2}, \quad \zeta = \frac{a + b}{2(ab + K)^{1/2}} \quad (8.232)$$

- The analytical solution for the step response is (see Equation 2.24)

$$y(t) = K_{ss} \left[1 - \frac{\omega_n}{\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t + \varphi) \right], \quad t \geq 0 \quad (8.233)$$

$$\omega_d = (1 - \zeta^2)^{1/2} \omega_n, \quad \varphi = \tan^{-1} \left(\frac{\omega_d}{\zeta \omega_n} \right) \quad (8.234)$$

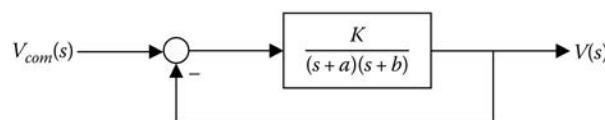


FIGURE 8.32 Block diagram of speed control system.

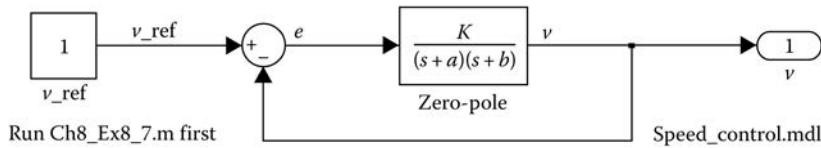


FIGURE 8.33 Simulink diagram for RK simulation of speed control system.

- c. RK integrators “ode1” through “ode5” of order one through five are available in MATLAB and Simulink. The Simulink diagram is shown in Figure 8.33.

The Simulink model file “speed_control.mdl” is called from the MATLAB M-file “Ch8_Ex8_7.m,” which sets the system parameters, selects the numerical integrator as either RK-2 or RK-4, sets the timing parameters (step size and simulation duration), and plots the analytical and simulated responses. The results are shown in Figures 8.34 and 8.35.

Some of the data points at the end of the simulated responses when $T = 0.025$ s in Figure 8.34 are omitted to make it easier to visualize the discrete-time nature of the response. Several of the simulated transient responses are quite accurate, whereas others deviate by a significant amount from the analytical solution. The RK-2 integrator is unstable when $K = 250$ and $T = 0.1$ s, and the RK-4 integrator exhibits instability for the case when $K = 250$ and $T = 0.2$ s. The reader should confirm that $\lambda T = -0.35 + j1.5740$ and $\lambda T = -0.7 + j3.1480$ fall outside the stability regions for RK-2 and RK-4, respectively.

- d. For the case when $K = 250$, the continuous-time system characteristic roots are $\lambda = -3.5 \pm j15.7401$. The limiting value of T for numerical stability is found by locating the intersection of the ray $\lambda T = (-3.5 + j15.7401)T$, $T > 0$ and the RK-3 stability boundary as shown in Figure 8.36. The M-file “Ch8_Ex8_7.m” contains MATLAB code, which tracks the values of λT along the RK-3 boundary as the point z rotates around the Unit Circle in the z -plane. The common point on the ray and stability boundary is located where the angle of λT on the stability boundary is equal to the constant angle of the ray (see Figure 8.36). It occurs at $\lambda T = -0.5198 + j2.3386$. The limiting step size is found from

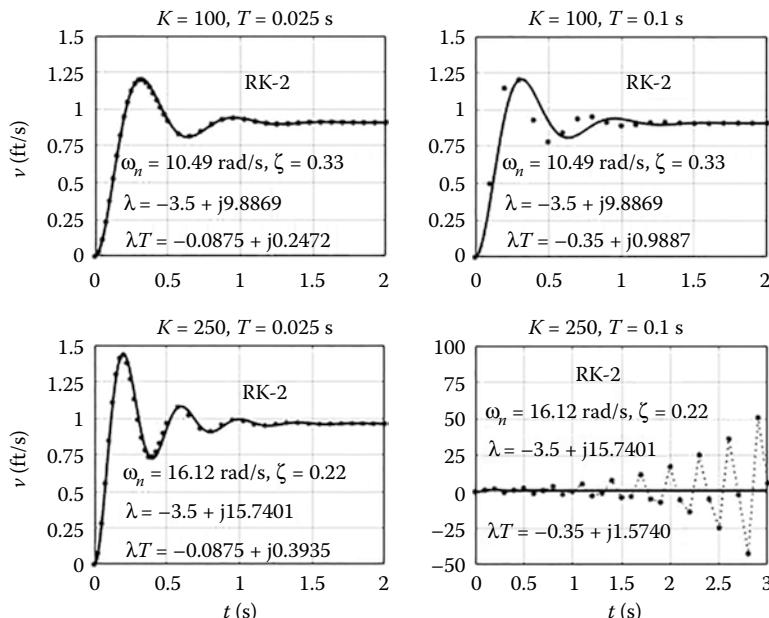


FIGURE 8.34 Analytical and RK-2 simulation of speed control system step response.

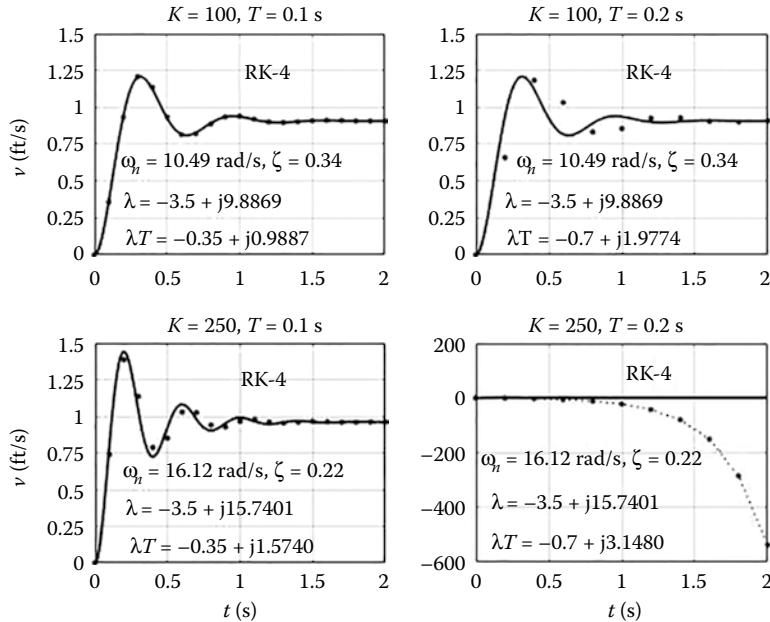


FIGURE 8.35 Analytical and RK-4 simulation of speed control system step response.

$$\lambda T_{max} = (-3.5 + j15.7401)T_{max} = -0.5198 + j2.3386 \quad (8.235)$$

Solving for T_{max} in Equation 8.235,

$$-3.5T_{max} = -0.5198 \Rightarrow T_{max} = 0.1485 \text{ s}$$

The Simulink model was run using the “ode3” RK-3 integrator with a step size of T_{max} . The marginally stable simulated response is shown in [Figure 8.37](#).

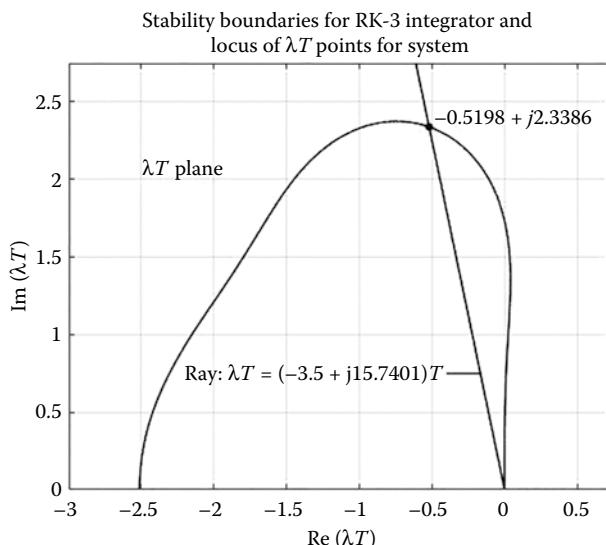


FIGURE 8.36 Finding λT_{max} point for RK-3 simulation of system.

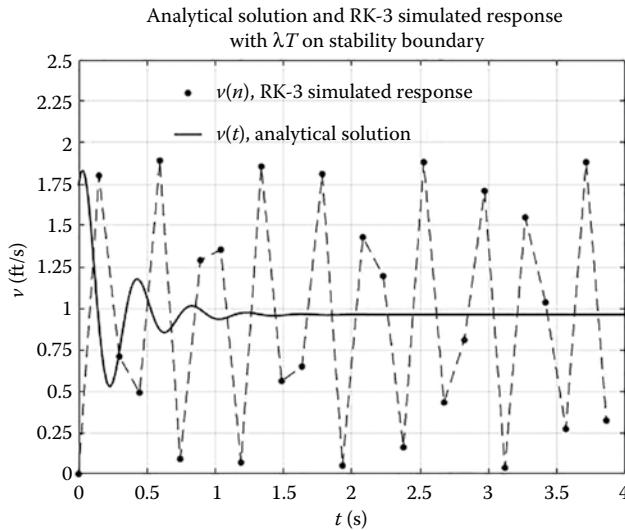


FIGURE 8.37 Analytical step response and marginally stable RK-3 simulated response.

EXERCISES

- 8.15 Show that the extraneous z -plane pole z_2 resulting from AB-2 integration of the first-order system $dx/dt = \lambda x + u$ is approximately equal to $0.5\lambda T$ when $\lambda T \ll 1$.
- 8.16 Simulate the unit step response of the first-order system $dx/dt = \lambda x + u$ using AB-2 integration, and plot both $x(t)$, $t \geq 0$ and $x(n)$, $n = 0, 1, 2, \dots$ for the following cases:

λ	T
-0.1	1, 2, 5, 9, 10, 11
-2	0.1, 0.2, 0.3, 0.4, 0.5, 0.6
-50	0.002, 0.005, 0.01, 0.019, 0.02, 0.021

- 8.17 Show that the procedure for finding the AB-4 stability region must be modified to account for the existence of extraneous poles outside the Unit Circle, that is, for certain values of λT , the principal pole may lie on the Unit Circle; however, there may be other poles of $H(z)$ larger in magnitude.
- 8.18 The stability boundary for AB integration in polar form is $\lambda T = Me^{j\psi}$ where $M = |\lambda T|$ and $\psi = \text{Arg}(\lambda T)$ are both functions of the angle θ as shown in Figure 8.23 for AB-2 integration.
- Show that $M = (2 - 2 \cos \theta)^{1/2}$ and $\psi = \tan^{-1}(\sin \theta / (\cos \theta - 1))$ for AB-1 integration.
 - Derive Equations 8.156 and 8.158.
 - Find M and ψ for AB-3 and AB-4 integration.
- 8.19 Investigate the stability of AB-3 and AB-4 integration for undamped continuous-time second-order systems. Specifically,
- Find the z -domain transfer functions of the system, and plot the loci of the poles as the parameter $\omega_n T$ varies, similar to Figure 8.25 for AB-2 integration.
 - Include close-ups of the stability boundaries near the imaginary axis of the λT plane.
- 8.20 In Example 8.4,
- Find all the z -plane poles when $\omega_n T = 0.05, 0.1, \dots, 0.45, 0.5$. Comment on how the results affect the stability of AB-2 integration of undamped second-order systems.
 - Show that the difference equation for implementing AB-2 integration of the system $\ddot{x} + \omega_n^2 x = u$ is

$$\begin{aligned} x(n+4) - 2x(n+3) + [1 + 2.25(\omega_n T)^2]x(n+2) - 1.5(\omega_n T)^2 x(n+1) \\ + 0.25(\omega_n T)^2 x_A(n) = 0.25T^2[9u(n+2) - 6u(n+1) + u(n)], \quad n = 0, 1, 2, 3, \dots \end{aligned}$$

- c. Find the difference equation for explicit Euler integration of the undamped second-order system.
 - d. Write a MATLAB M-file that accepts values for ω_n and T and implements AB-2 integration to simulate the unit step response of the system. Use the explicit Euler integrator to compute the starting values $x(2)$ and $x(3)$.
 - e. Plot the exact and simulated step responses for the following cases:
 - i. $\omega_n = 1$ rad/s, $T = 0.01$ s
 - ii. $\omega_n = 100$ rad/s, $T = 0.002$ s
 - iii. $\omega_n = 0.02$ rad/s, $T = 15$ s
 - iv. $\omega_n = 10$ rad/s, $T = 0.5$ s
- 8.21 Discuss the implications of the AB-3 stability boundary extending into the first quadrant of the λT plane. Illustrate by simulating the step response of the continuous-time second-order system
- $$\frac{d^2}{dt^2}x(t) - \frac{d}{dt}x(t) + 49.25x(t) = u(t)$$
- using AB-3 integration with step size $T = 0.1$ s. Plot the exact and simulated response on the same graph. What is the damping ratio and natural frequency of the continuous-time system?
- 8.22 Derive expressions for
- a. $H_f(z)$ for AM-2, AM-3, and AM-4 integrators given in Equations 8.167 through 8.169.
 - b. $H(z)$ for AM-2, AM-3, and AM-4 integration of $dx/dt = \lambda x + u$, ($\text{Re } \lambda < 0$) given in Equations 8.170 through 8.172.
 - c. λT in Equations 8.178 and 8.179.
- 8.23 Use the final value theorem (see Table 4.5) to obtain the final value for $c(n)|_{n \rightarrow \infty}$, given in Equation 8.187.
- 8.24 Find an expression for the equivalent continuous-time system characteristic root λ^* corresponding to the z -plane pole resulting from AM-2 simulation of the system $dx/dt = \lambda x + u$. Compute λ^* and e_λ (the characteristic root error) for the values of λ and T used in Example 8.5. Are your answers consistent with the responses in Figures 8.28 and 8.29?
- 8.25 For RK-2 integration of $dx/dt = \lambda x + u$ resulting in Equation 8.198 for $H(z)$,
- a. Find the z -plane pole of the discrete-time system.
 - b. Find the equivalent continuous-time system characteristic root λ^* .
 - c. Find asymptotic formulas for λ^* and the fractional error in λ^* , that is, $e_\lambda = \lambda^*/\lambda - 1$.
- 8.26 Find the difference equation for the RK-4 integrator in Section 6.2.
- 8.27 Derive the result in Equation 8.214 for the z -domain transfer function of the RK-3 integrator in Equations 8.209 through 8.212.
- 8.28 Derive the expression in Equation 8.224 for the fractional characteristic root error incurred using RK-4 integration.
- 8.29 Consider an unstable, second-order system with DC gain $k_{ss} = 1$, natural frequency $\omega_n = 50$ rad/s, and damping ratio $\zeta = -0.02$. The initial conditions are $x(0) = 1$ and $\dot{x}(0) = 0$.
- a. Use Simulink to simulate the transient response of the autonomous system using RK-2 and RK-4 integration with a step size of $T = 0.05$ s.
 - b. Find the analytical solution and plot it along with the RK-2 and RK-4 simulated responses on the same graph.
 - c. Comment on the results. Does λT lie inside the RK-2 and RK-4 stability regions?
- 8.30 Polar coordinates of the AB-2 stability boundary are expressed parametrically in Equations 8.156 and 8.158. Show that a parametric representation for the rectangular coordinates of the AB-2 stability boundary is given by

$$x = \operatorname{Re}(\lambda T) = \frac{4 \cos \theta - \cos 2\theta - 3}{5 - 3 \cos \theta}$$

$$y = \operatorname{Im}(\lambda T) = \frac{4 \sin \theta - \sin 2\theta}{5 - 3 \cos \theta}$$

for $0 \leq \theta \leq 2\pi$.

8.4 MULTIRATE INTEGRATION

The topic of stiff systems was introduced in Section 6.5. Recall that the stiffness property is a measure of the variation in magnitude between the smallest and largest characteristic roots (eigenvalues of the coefficient matrix A in state variable model) of a linear or linearized system. When the characteristic roots of a stiff system are as portrayed in Figure 8.38a, variable-step stiff integrators like MATLAB's "ode15s," "ode23s," "ode23tb" are more computationally efficient in simulating the system dynamics than fixed-step numerical integrators owing to the excessively small time steps necessary with fixed-step integrators to assure numerical stability.

When the system poles are clustered in distinct regions of the s -plane as shown in Figure 8.38b, the overall continuous-time system is composed of two or more subsystems that effectively operate at different speeds. Different time scales are required to view the time histories of the individual subsystem state variables. The pole locations in Figure 8.38b implies the existence of three subsystems, a relatively slow sixth-order subsystem associated with the six dominant poles nearest to the origin and imaginary axis, an intermediate speed fourth-order subsystem corresponding to the middle four poles, and a third-order fast subsystem arising from the three poles furthest from the origin.

Multirate integration methods are often effective in simulating continuous-time systems with identifiable subsystems like the one shown in Figure 8.38b. As the name suggests, numerical integrators running at different frame rates (step sizes) are tailored to the individual subsystems. The explanation and example that follow are geared toward a two-time scale system, that is, a system with characteristic roots in two distinct regions located an order of magnitude apart from the origin of the s -plane. By implication, a subset of the system's state variables are predominantly characterized by fast dynamics, that is, short time constants, high natural frequencies, and bandwidth, and the remaining states are just the opposite, namely, those associated with slow natural modes and longer transient responses.

Electromechanical control systems are frequently composed of fast and slow subsystems. Components in electronic controllers and sensors are much faster than the mechanical systems being controlled. The result is an overall system with fast and slow dynamics. Figure 8.39 is the block diagram of an aircraft pitch control system similar to one in Howe (1995). The airframe is modeled as a linear second-order system to account for the short-period longitudinal dynamics. The actuating signal for the controller is the difference between the commanded elevator deflection δ_c coming from the autopilot and the actual elevator deflection δ_e . The control surface actuator

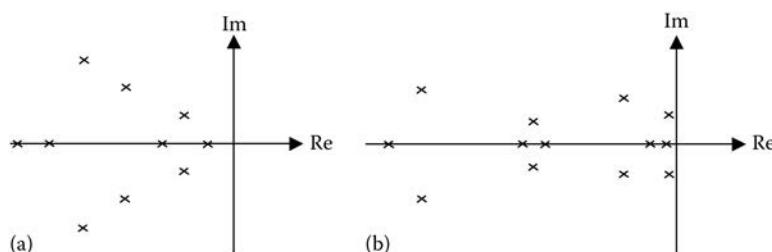


FIGURE 8.38 Stiff system (a) without distinct grouping of poles and (b) with distinct grouping of poles.

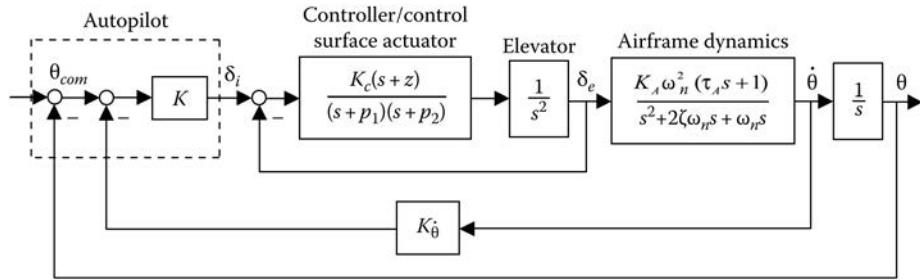


FIGURE 8.39 Block diagram of aircraft pitch control system.

(lumped with the controller) moves the elevator. The pitch θ and pitch rate $\dot{\theta}$ are fed back to the autopilot, which receives the pitch angle command θ_{com} from the pilot.

The airframe dynamics and subsequent integrator constitute the slow subsystem, and the fast subsystem is composed of the remaining components. Since the slow and fast states are to be integrated at different rates, it is necessary to define the slow and fast states and express the state derivatives in terms of the states and the command input. We begin with the slow subsystem blocks and perform the steps necessary to generate an equivalent simulation diagram. The transfer function of the airframe dynamics is

$$\frac{\dot{\theta}(s)}{\delta_e(s)} = \frac{K_A \omega_n^2 (\tau_A s + 1)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8.236)$$

leading to the differential equation

$$\frac{d^2\theta}{dt^2} + 2\zeta\omega_n \frac{d\theta}{dt} + \omega_n^2\dot{\theta} = K_A \omega_n^2 \tau_A \frac{d\delta_e}{dt} + K_A \omega_n^2 \delta_e \quad (8.237)$$

The simulation diagram is shown in [Figure 8.40](#). The integrator outputs are chosen as the slow system states x_1 , x_2 , and x_3 .

The fast systems states are obtained by breaking the controller/control surface actuator transfer function into serial first-order blocks as shown in [Figure 8.41](#). The simulation diagrams for the first-order blocks are drawn using the techniques introduced in Section 2.4.

The simulation diagram with the fast states x_4 , x_5 , x_6 , and x_7 is shown in [Figure 8.42](#).

From [Figures 8.39](#), [8.40](#), and [8.42](#), we are able to write algebraic and state derivative equation, which eventually lead to the following state model (see Exercise 8.31).

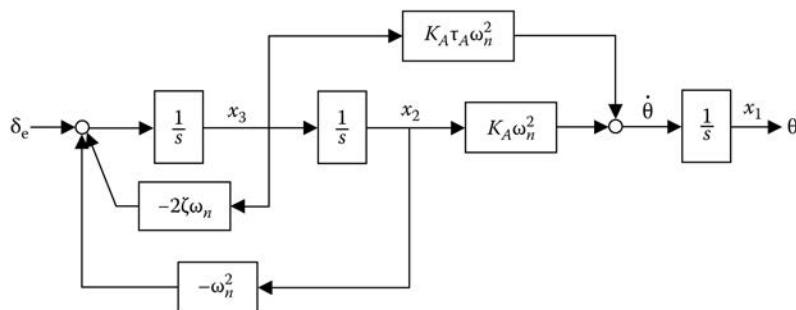


FIGURE 8.40 Simulation diagram of airframe dynamics with states x_1 , x_2 , and x_3 .

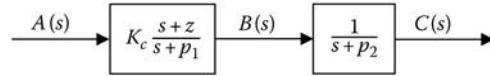


FIGURE 8.41 Controller/control surface actuator.

$$\dot{\underline{x}} = A\underline{x} + B\theta_{com} \quad (8.238)$$

$$\underline{y} = C\underline{x} + D\theta_{com} \quad (8.239)$$

where the matrices A , B , C , and D are given by

$$A = \begin{bmatrix} 0 & K_A\omega_n^2 & K_A\omega_n^2\tau_A & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\omega_n^2 & -2\zeta\omega_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -K_C K & -K_C K K_{\theta} K_A \omega_n^2 & -K_C K K_{\theta} K_A \omega_n^2 \tau_A & -K_C & 0 & -p_2 & K_C(z - p_1) \\ -K & -K K_{\theta} K_A \omega_n^2 & -K K_{\theta} K_A \omega_n^2 \tau_A & -1 & 0 & 0 & -p_1 \end{bmatrix} \quad (8.240)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ K_C K \\ K \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8.241)$$

Note that the output vector $\underline{y} = [y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ y_6 \ y_7]^T$ is chosen to be identical to the state vector \underline{x} . Decomposing the state vector \underline{x} into a vector of slow states $\underline{u} = [u_1 \ u_2 \ u_3]^T = [x_1 \ x_2 \ x_3]^T$ and fast states $\underline{w} = [w_1 \ w_2 \ w_3 \ w_4]^T = [x_4 \ x_5 \ x_6 \ x_7]^T$ leads to a definition of the slow state derivatives $\dot{\underline{u}} = \dot{\underline{x}}(u, w)$ as

$$\dot{u}_1 = \dot{x}_1 = f_1(u, w) = A_{1,2}u_2 + A_{1,3}u_3 \quad (8.242)$$

$$\dot{u}_2 = \dot{x}_2 = f_2(u, w) = A_{2,3}u_3 \quad (8.243)$$

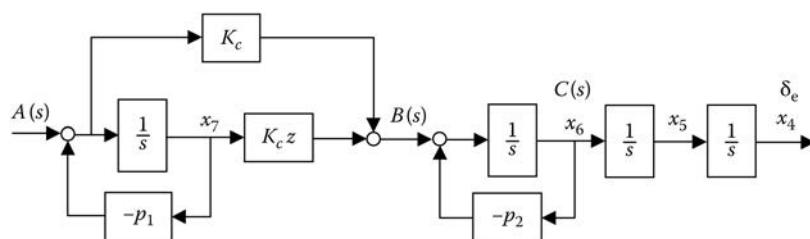


FIGURE 8.42 Simulation diagram of fast subsystem with states x_4 , x_5 , x_6 , and x_7 .

$$\dot{u}_3 = \dot{x}_3 = f_3(\underline{u}, \underline{w}) = A_{3,2}u_2 + A_{3,3}u_3 + A_{3,4}W_1 \quad (8.244)$$

and the fast state derivative vector $\dot{\underline{w}} = \underline{g}(\underline{u}, \underline{w}, \theta_{com})$ is

$$\dot{w}_1 = \dot{x}_4 = g_1(\underline{u}, \underline{w}, \theta_{com}) = A_{4,5}w_2 \quad (8.245)$$

$$\dot{w}_2 = \dot{x}_5 = g_2(\underline{u}, \underline{w}, \theta_{com}) = A_{5,6}w_3 \quad (8.246)$$

$$\begin{aligned} \dot{w}_3 &= \dot{x}_6 = g_3(\underline{u}, \underline{w}, \theta_{com}) \\ &= A_{6,1}u_1 + A_{6,2}u_2 + A_{6,3}u_3 + A_{6,4}w_1 + A_{6,6}w_3 + A_{6,7}w_4 + B_6\theta_{com} \end{aligned} \quad (8.247)$$

$$\dot{w}_4 = \dot{x}_7 = g_4(\underline{u}, \underline{w}, \theta_{com}) = A_{7,1}u_1 + A_{7,2}u_2 + A_{7,3}u_3 - w_1 + A_{7,7}w_4 + B_7\theta_{com} \quad (8.248)$$

where the coefficients $A_{i,j}$ are the elements in the coefficient matrix A in Equation 8.240.

[Figure 8.43](#) portrays the slow and fast subsystems and the coupling between them. Once the system is decomposed into a slow and fast subsystem, the numerical integration routine and frame times (step size T_s for the slow subsystem and T_f for the fast one) must be selected. The numerical integrator to update the slow states is referred to as the “master” routine, and the integration method for advancing the fast states is called the “slave” routine (Palusinski 1986). The situation is illustrated in [Figure 8.44](#) for the case where both “master” and “slave” are the classic RK-4 integrator (see Equations 6.60 through 6.64) with step sizes T_s and T_f ; respectively. The quotient $N = T_s/T_f$ is called the frame ratio. A single slow and fast state is shown for simplicity.

There are several choices when it comes to scheduling the order of execution for slow and fast frames. Referring to [Figure 8.44](#), starting at time t_n , we can take a half step through the slow frame starting from u_n in the direction defined by slope k_1 . The endpoints (t_n, u_n) and $(t_n + 0.5T_s, u_{n+1/2})$ determine the equation of a line that is interpolated to provide the value of the slow state at the beginning of each fast frame. The fast state is then advanced using RK-4 integration up until the time $t_n + 0.5NT_f$, generating values for $w_{nN+1}, w_{nN+2}, \dots, w_{nN+0.5N}$. Next, the slow state derivative function k_2 is evaluated at $t_n + 0.5T_s$, using the predicted slow state $u_{n+1/2}$ along with the previously computed fast state $w_{(n+0.5)N}$.

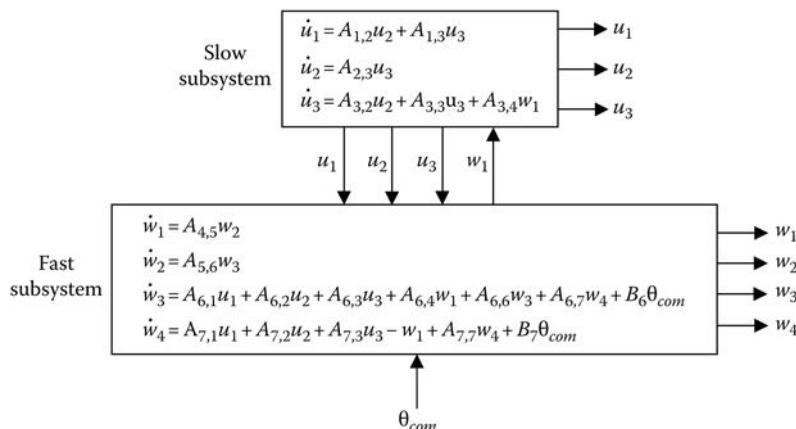


FIGURE 8.43 Slow and fast subsystem interaction.

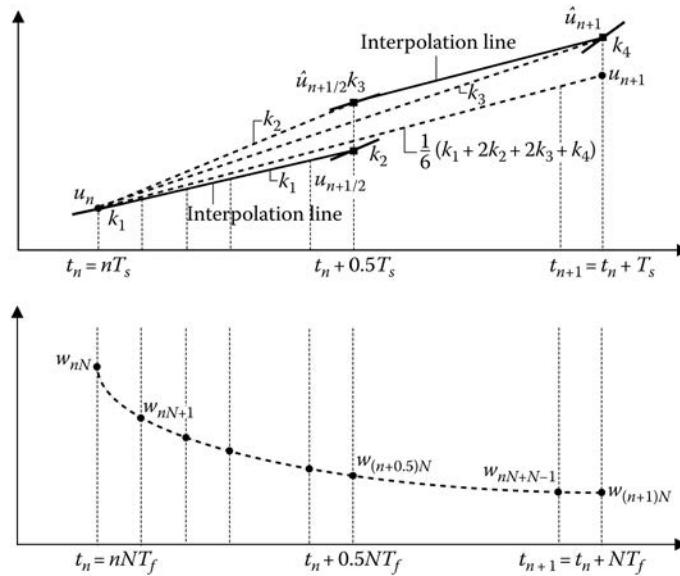


FIGURE 8.44 Multirate integration for one frame of slow state and N frames of fast state.

The step-by-step process for updating the slow state vector \underline{u} and fast state vector \underline{w} from t_n to t_{n+1} is outlined in the following.

8.4.1 PROCEDURE FOR UPDATING SLOW AND FAST STATES: MASTER/SLAVE = RK-4/RK-4

1. Compute $k_1 = f(\underline{u}_n, \underline{w}_{nN})$
2. Compute $\underline{u}_{n+1/2} = \underline{u}_n + 0.5T_s k_1$
3. Determine equation of lines connecting (t_n, \underline{u}_n) and $(t_{n+1/2}, \underline{u}_{n+1/2})$
4. Use “slave” RK-4 to integrate fast state from t_n to $t_n + 0.5NT_f$ based on interpolated values for slow state at beginning of fast frame times.
5. Compute $k_2 = f(\underline{u}_{n+1/2}, \underline{w}_{(n+0.5)N})$
6. Compute $\hat{\underline{u}}_{n+1/2} = \underline{u}_n + 0.5T_s k_2$
7. Compute $k_3 = f(\hat{\underline{u}}_{n+1/2}, \underline{w}_{(n+0.5)N})$
8. Compute $\hat{\underline{u}}_{n+1} = \underline{u}_n + T_s k_3$
9. Determine equation of line connecting $(t_n + 0.5T_s, \hat{\underline{u}}_{n+1/2})$ and $(t_n + 1, \hat{\underline{u}}_{n+1})$
10. Use “slave” RK-4 to integrate fast state from $t_n + 0.5NT_f$ to $t_{n+1} = t_n + NT_f$ based on interpolated values for slow state at beginning of fast frame times.
11. Compute $k_4 = f(\hat{\underline{u}}_{n+1}, \underline{w}_{(n+1)N})$
12. Compute updated slow state $\underline{u}_{n+1} = \underline{u}_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

The choice of frame times T_f and T_s depends on the integrators used for the “master” and “slave” routines as well as the dynamics of the slow and fast subsystems. Baseline values of the system parameters for the following discussion are (see Figure 8.39)

Airframe dynamics: $K_A = 10$, $\tau_A = 0.8$ s, $\omega_n = 5$ rad/s, $\zeta = 0.2$

Controller/control surface actuator: $K_c = 4 \times 10^5$, $z = 12.5$, $p_1 = p_2 = 100$

Autopilot gain: $K = 0.1625$, pitch rate feedback sensor gain: $K_{\dot{\theta}} = 0.2$

Substituting the parameter values into Equation 8.240 gives the coefficient matrix A with eigenvalues (characteristic roots) equal to the closed-loop system poles. The result is

$$\lambda_1 = -0.67, \lambda_{2,3} = -4.31 \pm j6.48, \lambda_4 = -21.97, \lambda_{5,6} = -11.07 \pm j37.46, \lambda_7 = -148.59$$

Magnitudes of the system poles range from a low of $|\lambda_1| = 0.67$ to a high of $|\lambda_7| = 148.59$, demonstrating the stiffness of the system. The magnitude of the remaining poles suggests the existence of a slow subsystem characterized by the first three poles λ_1, λ_2 , and λ_3 , and a fast subsystem corresponding to the remaining four poles $\lambda_4, \lambda_5, \lambda_6$, and λ_7 located further from the origin than λ_1, λ_2 , and λ_3 .

8.4.2 SELECTION OF STEP SIZE BASED ON STABILITY

Simulation of the seventh-order control system with classic RK-4 integration and step size T is stable provided the points $\lambda_i T, i = 1, 2, \dots, 7$ fall within the RK-4 stability region. Figure 8.45a shows the location of $\lambda_i T, i = 1, 2, \dots, 7$ when the integration step size T is 0.01 s. Since all 7 $\lambda_i T$ points are inside the stability boundary, the RK-4 simulation is stable. The RK-4 simulation is marginally stable when the leftmost $\lambda_i T$ point is located on the stability boundary at -2.785 . The step T_{max} is obtained from

$$-\max |\lambda_1| T_{max} = -148.59 T_{max} = -2.785 \Rightarrow T_{max} = 0.0187 \text{ s} \quad (8.249)$$

Figure 8.45b illustrates the case where $T = T_{max} = 0.0187$ s. The leftmost value of $\lambda_i T$ is on the RK-4 stability boundary at -2.785 , leading to a z -plane pole of the discrete-time system located on the Unit Circle.

A Simulink diagram of the pitch control system is shown in Figure 8.46.

The diagram includes a “State-Space” block to implement the state equations in Equations 8.238 through 8.241. The pitch input command is given by the exponential rise

$$\theta_{com}(t) = \bar{\theta}_{com}(1 - e^{-t/\tau_{com}}), \quad t \geq 0 \quad (8.250)$$

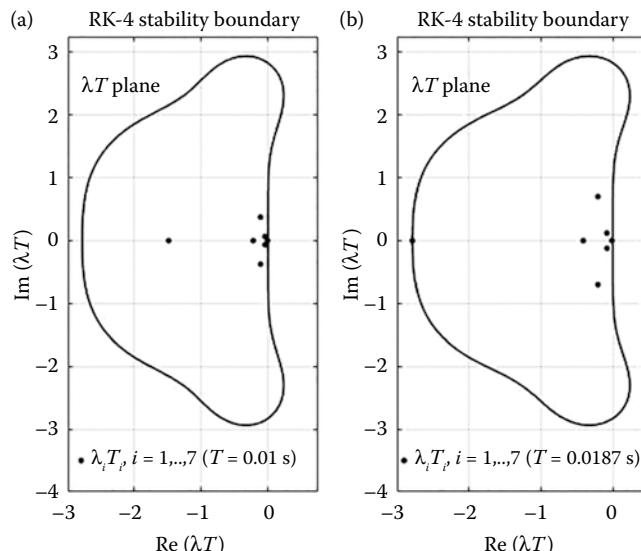


FIGURE 8.45 RK-4 simulation boundary: (a) stable and (b) marginally stable.

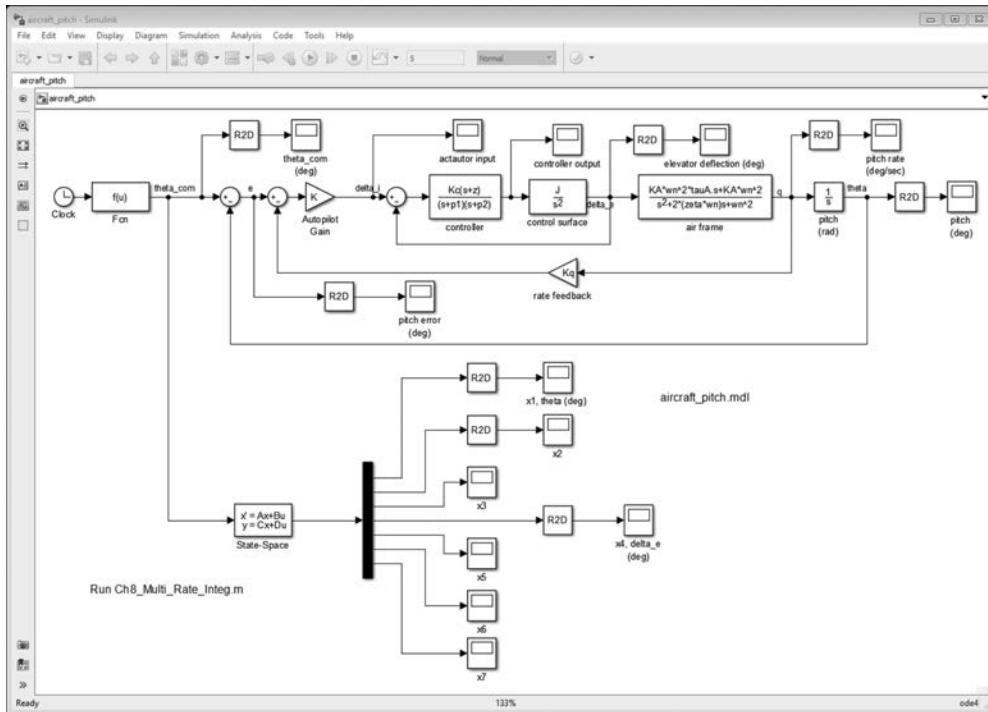


FIGURE 8.46 Simulink diagram of aircraft pitch control system in Figure 8.39.

which represents a real-world approximation to a step input provided τ_{com} is chosen appropriately, that is, $5\tau_{com}$ is set equal to 10–20 times the fixed integration step size.

Figure 8.47 shows the pitch responses from the scopes labeled “pitch (deg)” and “ x_1 theta (deg)” for the integration step sizes of 0.018 and 0.019 s. The numerical instability when $T = 0.019$ s is predicted by Equation 8.249.

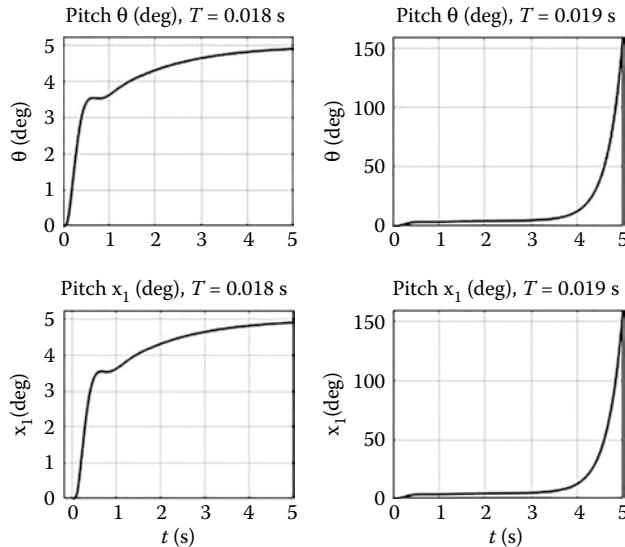


FIGURE 8.47 Simulink scope outputs showing stable ($T = 0.018$ s) and unstable ($T = 0.019$ s) pitch responses with RK-4 integration.

The reader can run the M-file “*Ch8_Multi_Rate_Integ.m*” with different system parameter values to compare outputs $x_1(t)$, $x_4(t)$, and $x_6(t)$ from the state variable model and the equivalent signals $\theta(t)$ and $\delta_e(t)$ and the output of the controller block.

8.4.3 SELECTION OF STEP SIZE BASED ON DYNAMIC ACCURACY

The transfer function of the closed-loop system in Figure 8.39 can be obtained using block diagram reduction or other graphical techniques such as Mason’s gain formula for signal flow graphs. The result is

$$G_{\theta_{com} \rightarrow \theta}(s) = \frac{\theta(s)}{\theta_{com}(s)} = \frac{\beta_2 s^2 + \beta_1 s + \beta_0}{s^7 + \alpha_6 s^6 + \alpha_5 s^5 + \alpha_4 s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0} \quad (8.251)$$

where

$$\beta_0 = KK_c K_A \omega_n^2 z, \quad \beta_1 = KK_c K_A (1 + \tau_A z) \omega_n^2, \quad \beta_2 = KK_c K_A \omega_n^2 \tau_A \quad (8.252)$$

$$\left. \begin{aligned} \alpha_0 &= KK_c K_A \omega_n^2 z, \\ \alpha_1 &= KK_c K_A \omega_n^2 z + K_c \omega_n^2 z (1 + KK_A K_{\dot{\theta}}) \\ \alpha_2 &= KK_c K_A \omega_n^2 \tau_A + K_c [2\zeta \omega_n z + \omega_n^2 + KK_c K_{\dot{\theta}} \omega_n^2 (1 + \tau_A z)] \\ \alpha_3 &= K_c z + 2\zeta \omega_n K_c + \omega_n^2 p_1 p_2 + KK_c K_A K_{\dot{\theta}} \tau_A \omega_n^2 \\ \alpha_4 &= K_c + 2\zeta \omega_n p_1 p_2 + \omega_n^2 (p_1 + p_2) \\ \alpha_5 &= p_1 p_2 + 2\zeta \omega_n (p_1 p_2) + \omega_n^2 \\ \alpha_6 &= p_1 + p_2 + 2\zeta \omega_n \end{aligned} \right\} \quad (8.253)$$

An equivalent implementation of the system with transfer function in Equation 8.251 consists of the input $\theta_{com}(t)$ feeding parallel first-and second-order components with the outputs of each block summed to generate the pitch response $\theta(t)$. Partial fraction expansion of Equation 8.251 using numerical values for β_0 , β_1 , β_2 , and α_0 , α_1 ..., α_6 based on the given system parameter values leads to the configuration shown in Figure 8.48.

Dashed lines identify the slow and fast subsystem components. Despite the fact that $\theta = x_1$ is classified as one of the slow states, it is clear from Figure 8.48 that $\theta(t)$ comprises both fast and slow components. However, it will be shown later that the fast component is negligible compared with the slow component.

The slow subsystem constants are

$$A_l = 0.3373, \lambda_1 = -0.67, B_l = 0.7930, B_0 = 37.0325, b_l = 8.6254, b_0 = 60.6057$$

and the fast subsystem constants are

$$\begin{aligned} C_1 &= -1.7568, \lambda_4 = -21.97, D_1 = 0.6477, D_0 = -49.8169, \\ d_1 &= 22.1383, d_0 = 1525.7, E_1 = 0.0328, \lambda_7 = -148.59 \end{aligned}$$

The poles of the slow subsystem second-order component (roots of $s^2 + b_l s + b_0$) are $\lambda_{2,3} = -4.31 \pm j6.48$. The fast subsystem second-order component poles are roots of $s^2 + d_1 s + d_0$, namely, $\lambda_{5,6} = -11.07 \pm j37.46$.

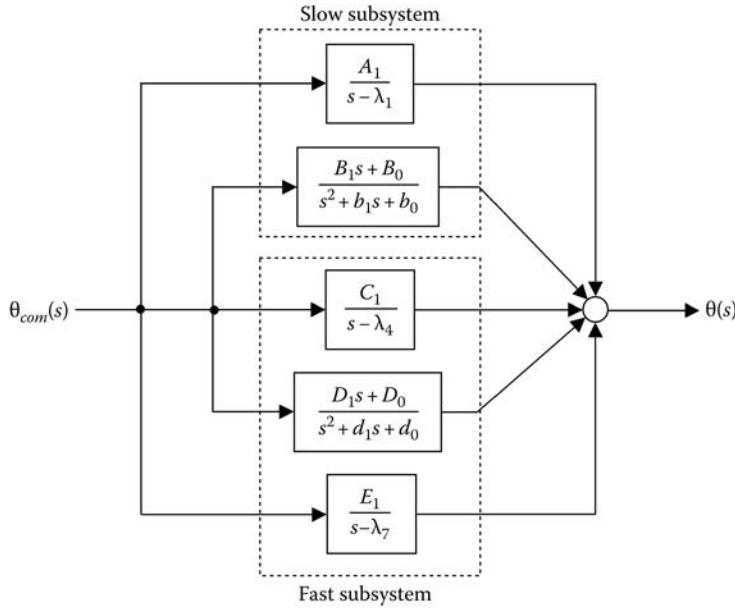


FIGURE 8.48 Parallel implementation of pitch control system transfer function.

Table 8.6 lists asymptotic formulas for the characteristic root errors resulting from the use of certain low-order numerical integrators. In particular, for RK-4 with step size T and integrator error coefficient $e_I = 1/120$,

$$\text{RK-4: } e_\lambda = \frac{\lambda^*}{\lambda} - 1 \approx -e_I(\lambda T)^4 \approx -\frac{1}{120}(\lambda T)^4, \quad |\lambda T| \ll 1 \quad (8.254)$$

where λ^* is the characteristic root of the equivalent continuous-time system.

For second-order systems Howe (1986) presents formulas for dynamic errors in damping ratio ζ , natural frequency ω_n , and damped natural frequency ω_d using first-order through fourth-order integration methods. For RK-4, the asymptotic expressions are

$$e_\zeta = \frac{\zeta^* - \zeta}{\zeta} \approx -4(\zeta - 3\zeta^3 + 2\zeta^5)e_I(\omega_n T)^4 \quad (8.255)$$

$$\approx -\frac{1}{30}(\zeta - 3\zeta^3 + 2\zeta^5)(\omega_n T)^4, \quad \omega_n T \ll 1 \quad (8.256)$$

$$e_{\omega_n} = \frac{\omega_n^* - \omega_n}{\omega_n} \approx -(1 - 8\zeta^2 + 8\zeta^4)e_I(\omega_n T)^4 \quad (8.257)$$

$$\approx -\frac{1}{120}(1 - 8\zeta^2 + 8\zeta^4)(\omega_n T)^4, \quad \omega_n T \ll 1 \quad (8.258)$$

$$e_{\omega_d} = \frac{\omega_d^* - \omega_d}{\omega_d} \approx -(1 - 12\zeta^2 + 16\zeta^4)e_I(\omega_n T)^4 \quad (8.259)$$

$$\approx -\frac{1}{120}(1-12\zeta^2+16\zeta^4)(\omega_n T)^4, \quad \omega_n T \ll 1 \quad (8.260)$$

For the first-order component in the slow subsystem in [Figure 8.48](#),

$$e_\lambda \approx -\frac{1}{120}(\lambda_1 T)^4 \approx -\frac{1}{120}(-0.673)^4 T^4 \approx -0.00171 T^4 \quad (8.261)$$

The damping ratio, natural frequency, and damped natural frequency of the slow subsystem second-order component are found by equating the term $s^2 + b_1 s + b_0$ and the standard form of a quadratic characteristic polynomial $s^2 + 2\zeta\omega_n s + \omega_n^2$. The results are $\zeta_{slow} = 0.554$, $(\omega_n)_{slow} = 7.785$ rad/s, and $(\omega_d)_{slow} = 6.481$ rad/s. Substituting the values of ζ_{slow} and $(\omega_n)_{slow}$ into Equations 8.256, 8.258, and 8.260 gives

$$e_\zeta \approx -\frac{1}{30}(\zeta_{slow} - 3\zeta_{slow}^3 + 2\zeta_{slow}^5)[(\omega_n)_{slow} T]^4 \quad (8.262)$$

$$\approx -\frac{1}{30}[0.554 - 3(0.554)^3 + 2(0.554)^5](7.785T)^4 \quad (8.263)$$

$$\approx -18.1562 T^4 \quad (8.264)$$

$$e_{\omega_n} \approx -\frac{1}{120}(1-8\zeta_{slow}^2+8\zeta_{slow}^4)[(\omega_n)_{slow} T]^4 \quad (8.265)$$

$$\approx -\frac{1}{120}(1-8(0.554)^2+8(0.554)^4)(7.785T)^4 \quad (8.266)$$

$$\approx 21.4777 T^4 \quad (8.267)$$

$$e_{\omega_d} \approx -\frac{1}{120}(1-12\zeta_{slow}^2+16\zeta_{slow}^4)[(\omega_n)_{slow} T]^4 \quad (8.268)$$

$$\approx -\frac{1}{120}[1-12(0.554)^2+16(0.554)^4](7.785T)^4 \quad (8.269)$$

$$\approx 35.9894 T^4 \quad (8.270)$$

Choosing the RK-4 step size to limit the characteristic error in damped natural frequency to 0.025%,

$$T_{slow} = \left[\frac{(e_{\omega_d})_{des}}{35.9894} \right]^{1/4} = \left[\frac{0.00025}{35.9894} \right]^{1/4} = 0.0513 \text{ s} \quad (8.271)$$

The actual characteristic error in damped natural frequency will be slightly different from $(e_{\omega_d})_{des} = 0.025\%$ because $(\omega_n)_{slow} T_{slow} = 0.3997$, which is not an order of magnitude less than 1, a requirement for the asymptotic formula in Equation 8.260.

A similar procedure can be performed to determine an appropriate step size for RK-4 simulation of the fast subsystem. Suppose the fast subsystem step size is selected to limit the sum of the characteristic root errors associated with the fast poles $\lambda_4 = -21.97$ and $\lambda_7 = -148.59$. From Equation 8.261,

$$e_{\lambda_4} + e_{\lambda_7} \approx -\frac{1}{120} [\lambda_4^4 + \lambda_7^4] T^4 \leq E_{des} \quad (8.272)$$

$$\Rightarrow -\frac{1}{120} [(-21.97)^4 + (-148.59)^4] T^4 \leq E_{des} \quad (8.273)$$

Choosing $E_{des} = -0.02\%$,

$$\begin{aligned} T^4 &\leq \frac{-0.0002}{-(1/120)[(-21.97)^4 + (-148.59)^4]} \\ \Rightarrow T_{fast} &\leq 0.0026 \text{ s} \end{aligned} \quad (8.274)$$

Once again, there will be a slight difference between $e_{\lambda_4} + e_{\lambda_7}$ and E_{des} when $T_{fast} = 0.0026$ s, because the product $|\lambda_7 T_{fast}| = |(-148.59)0.0026| = 0.3936$ is not significantly less than 1 as required in the asymptotic formula of Equation 8.254.

Henceforth, multirate integration using RK-4 for both slow and fast systems will be performed with $T_f = 0.0025$ s and $T_s = 0.05$ s resulting in a frame ratio $N = 20$.

8.4.4 ANALYTICAL SOLUTION FOR STATE VARIABLES

In most cases, analytical solutions for the state variables are not available with the possible exception of linear (or linearized) system models and elementary input signals. An advantage of knowing the analytical solution for the state variables is that it can serve as a benchmark for comparing results obtained by different simulation-based approaches. Consequently, the analytical solution for a subset of the state variables in the pitch control system will be determined with this purpose in mind.

Laplace transforming the pitch command signal given in Equation 8.250 gives

$$\theta(s) = \left[\frac{\beta_2 s^2 + \beta_1 s + \beta_0}{s^7 + \alpha_6 s^6 + \alpha_5 s^5 + \alpha_4 s^3 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0} \right] \frac{\bar{\theta}_{com}}{s(\tau_{com}s + 1)} \quad (8.275)$$

Choosing $(\bar{\theta}_{com} = 5^\circ, \tau_{com} = 0.01$ s and substituting the baseline parameter values into Equations 8.252 and 8.253 determine $\theta(s)$. Using MATLAB's "conv" function to expand the denominator into a ninth-order polynomial and then the "residue" function results in the partial fraction expansion of $\theta(s)$. Converting pairs of terms with complex poles and coefficients into real terms results in the analytical pitch response

$$\theta(t) = \theta_{slow}(t) + \theta_{fast}(t) + \theta_{forced}(t) \quad (8.276)$$

where $\theta_{slow}(t)$ comprises the slow subsystem natural mode terms,

$$\begin{aligned} \theta_{slow}(t) = 5\{ &-0.5047e^{-0.673t} + e^{-4.313t}[-0.6150\cos(6.4812t) \\ &- 0.3474\sin(6.4812t)] \} \end{aligned} \quad (8.277)$$

$\theta_{fast}(t)$ is made up of fast subsystem natural mode terms,

$$\begin{aligned}\theta_{fast}(t) = & 5\{0.0004548e^{-148.589t} + 0.1025e^{-21.975t} \\ & + e^{-11.069t}[0.0204\cos(37.4583t) + 0.0388\sin(37.4583t)]\}\end{aligned}\quad (8.278)$$

and $\theta_{forced}(t)$ includes the input mode terms

$$\theta_{forced}(t) = 5(1 - 0.0035e^{-100t}) \quad (8.279)$$

The exponential decay in the forced component results from the exponential term in the command input (see Equation 8.250).

Plots of the slow component $\theta_{slow}(t)$ and fast component $\theta_{fast}(t)$ shown in the top half of Figure 8.49 suggest that the fast component contributes a negligible amount to the overall response. Hence, $\theta(t)$ is appropriately classified as a slow subsystem state variable.

The bottom half of Figure 8.49 shows the forced component given in Equation 8.279 and the total pitch response comprising the slow, fast, and forced components. Note that on the time scale used in Figure 8.49, the forced component appears to be a step input. In reality, it contains an exponential rise term with time constant $\tau_{com} = 0.01$ s.

A similar approach can be used to find the analytical solution for the fast state variable $x_4 = \delta_e$. The transfer function from $\theta_{com}(s)$ to $\delta_e(s)$ is

$$G_{\theta_{com} \rightarrow \delta_e(s)} = \frac{\delta_e(s)}{\theta_{com}(s)} = \frac{\gamma_4 s^3 + \gamma_3 s^3 + \gamma_2 s^2 + \gamma_1 s + \gamma_0}{s^7 + \alpha_6 s^6 + \alpha_5 s^5 + \alpha_4 s^4 + \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0} \quad (8.280)$$

$$\gamma_0 = 0, \gamma_1 = KK_c \omega_n^2 z, \gamma_2 = KK_c (\omega_n^2 + 2\zeta \omega_n z), \gamma_3 = KK_c (2\zeta \omega_n + z), \gamma_4 = KK_c \quad (8.281)$$

and the analytical solution for the elevator deflection $\delta_e(t)$ is

$$\delta_e(t) = (\delta_e)_{slow}(t) + (\delta_e)_{fast}(t) + (\delta_e)_{forced}(t) \quad (8.282)$$

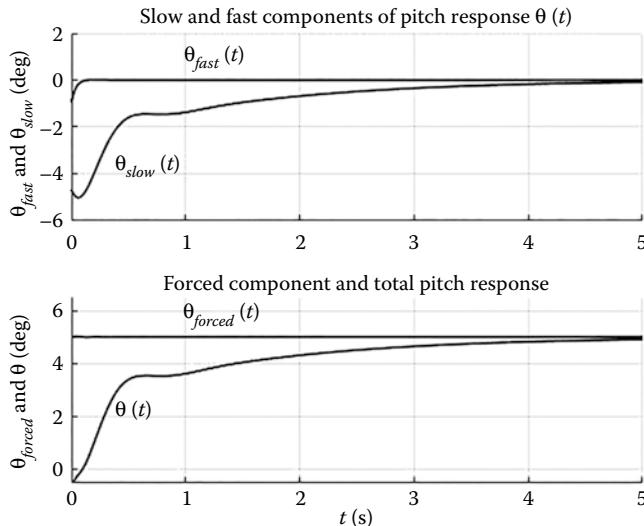


FIGURE 8.49 Total pitch response and its components.

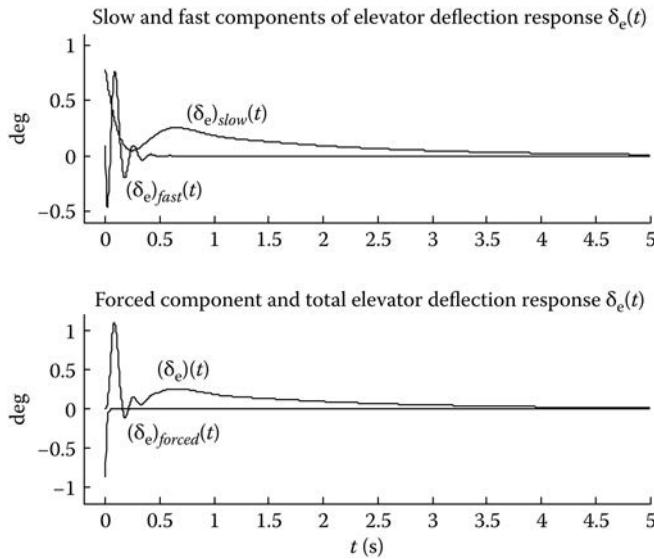


FIGURE 8.50 Total elevator deflection and its components.

$$(\delta_e)_{slow}(t) = 5\{0.0709e^{-0.673t} + e^{-4.313t}[0.0844 \cos(6.4812t) - 0.1440 \sin(6.4812t)]\} \quad (8.283)$$

$$(\delta_e)_{fast}(t) = 5\{0.050e^{-148.589t} + 0.25e^{-21.975t} + e^{-11.069t}[-0.2842 \cos(37.4583t) - 0.1644 \sin(37.4583t)]\} \quad (8.284)$$

$$(\delta_e)_{forced}(t) = 5(-0.1733e^{-100t}) \quad (8.285)$$

The total response $\delta_e(t)$ and its components $(\delta_e)_{slow}(t)$, $(\delta_e)_{fast}(t)$, and $(\delta_e)_{forced}(t)$ are shown in Figure 8.50.

From the top graph, it is clear that both fast and slow components are present in $\delta_e(t)$. Despite the existence of an appreciable slow component, $\delta_e(t)$ is nonetheless identified as a fast state variable. Multirate simulation of the overall system must integrate $\delta_e(t)$ at the fast frame rate due to the significant high-frequency component $(\delta_e)_{fast}(t)$.

8.4.5 MULTIRATE INTEGRATION OF AIRCRAFT PITCH CONTROL SYSTEM

The MATLAB M-file “Ch8_multi_rate_integ.m” includes code for implementing multirate integration of Equations 8.242 through 8.248 with RK-4 as the “master” and “slave” integration routines.

Figure 8.51 shows Simulink and multirate integration results for the slow states $u_1 = x_1$, $u_2 = x_2$, and $u_3 = x_3$. The Simulink model was integrated using RK-4 with integration step size identical to the fast frame time $T_f = 0.0025$ s. The slow frame time was $T_s = 0.05$ making the frame ratio $N = 20$.

The Simulink and multirate simulation responses are in general agreement; however, the accuracy of each can only be established by comparison with the analytical solutions. Accordingly, Figures 8.52 and 8.53 show the analytical solutions for $x_1(t)$ and $x_2(t)$ on the same graph with the RK-4 and multirate simulation results. For purposes of clarity, not all simulated points are shown in

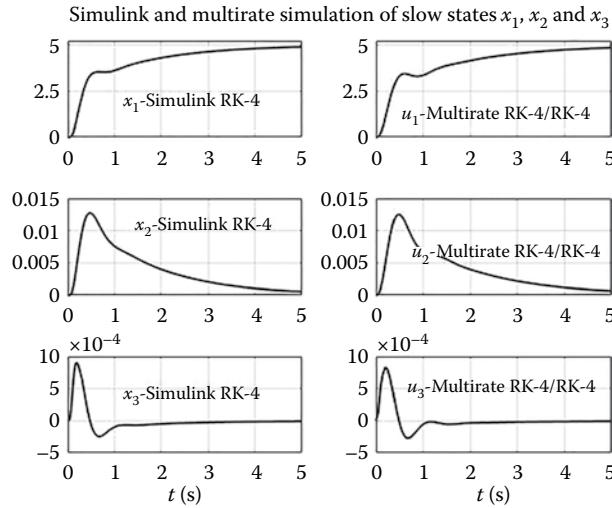


FIGURE 8.51 Simulation of slow states using Simulink and multirate integration.

the graph. The analytical solution for $x_1(t) = \theta(t)$ is given by Equations 8.276 through 8.279, and the one for $x_2(t)$ is obtained in the MATLAB M-file “*Ch8_multi_rate_integ.m*.”

From Figures 8.52 and 8.53, it is clear that the simulated responses using Simulink with RK-4 and step size $T_f = 0.0025$ s are virtually identical with the analytical solutions. As expected, the responses obtained using multirate integration with RK-4/RK-4 and $T_f = 0.0025$ s, $T_f = 0.05$ s are not as accurate.

Simulated responses of two of the fast states, namely, $w_1 = x_4$ and $w_2 = x_5$, obtained using Simulink and multirate integration are shown in Figure 8.54. The M-file “*Ch8_multi_rate_integ.m*” plots the remaining fast states $w_3 = x_6$ and $w_4 = x_7$.

The analytical solution for $x_4 = \delta_e$, the elevator deflection, is shown in Figure 8.55 along with the responses from Simulink and multirate integration. Once again, the Simulink RK-4 and analytical

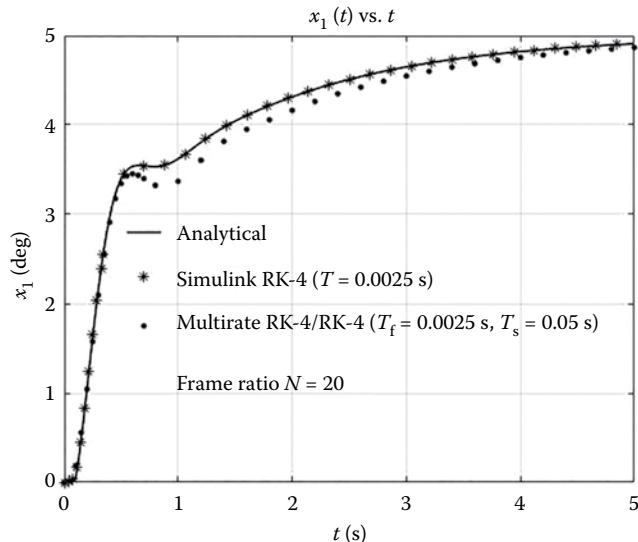


FIGURE 8.52 Comparison of analytical, Simulink, and multirate $x_1(t)$ responses.

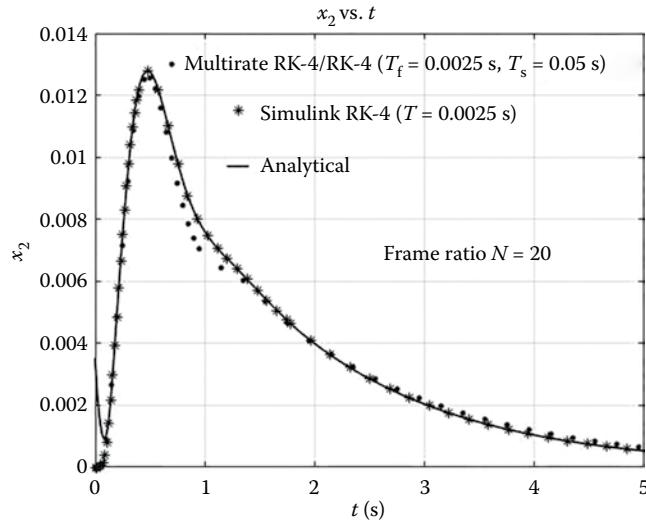


FIGURE 8.53 Comparison of analytical, Simulink, and multirate $x_2(t)$ responses.

responses are indistinguishable from each other, while the multirate solution deviates from both during the transient response period.

Multirate integration introduced errors in the transient response of each state variable in the aircraft pitch control system. The errors can be reduced by decreasing the frame ratio; however, the benefits from using multirate integration are lessened. An acceptable trade-off is generally possible.

Significant increases in performance are achieved using multirate integration for multiple time scale systems where the predominant number of states are associated with the slow subsystem(s). Moreover, the computational savings can be substantial when the times required to compute the slow subsystem state derivatives are appreciable due to the complex nature of the derivative functions or possibly due to the use of table lookups involved in the computation process.

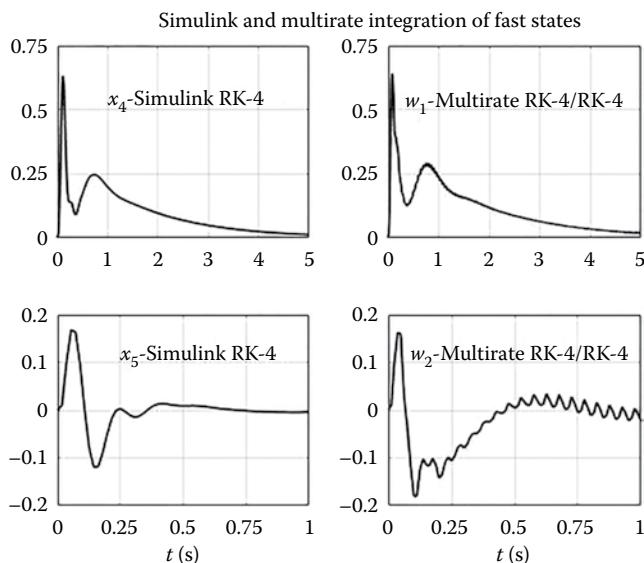


FIGURE 8.54 Simulation of fast states using Simulink and multirate integration.

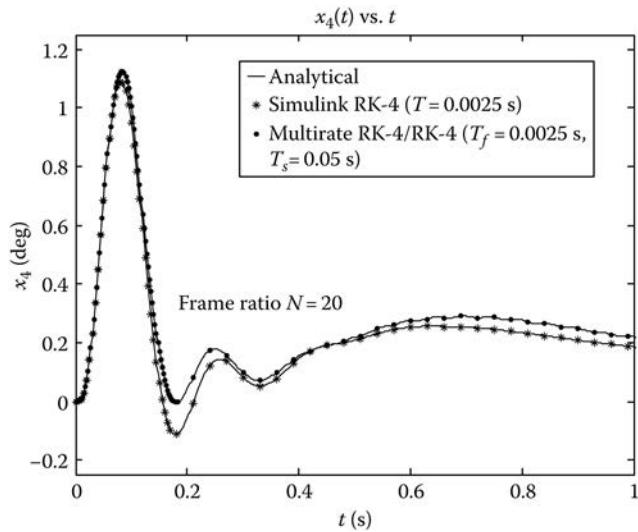


FIGURE 8.55 Comparison of analytical, Simulink, and multirate $x_4(t)$ responses.

8.4.6 NONLINEAR DUAL SPEED SECOND-ORDER SYSTEM

We now turn our attention to a second-order stiff system with a fast and a slow state. Furthermore, the system dynamics are nonlinear and the stiffness varies with the operating point of the linearized system. The system consists of two cylindrical tanks in series as shown in Figure 8.56.

Flow $F_0(t)$ into the first tank (open at the top) is completely controlled by a regulating valve in the inflow line. The outflow $F_2(t)$ from the second tank (sealed at the top) is a function of valve opening in the outflow line along with the liquid pressure at the bottom of the tank.

The system is modeled by differential and algebraic equations. Dynamics of the first tank are governed by

$$A_1 \frac{d}{dt} H_1(t) + F_{12}(t) = F_0(t), \quad H_1(t) \leq L_1 \quad (8.286)$$

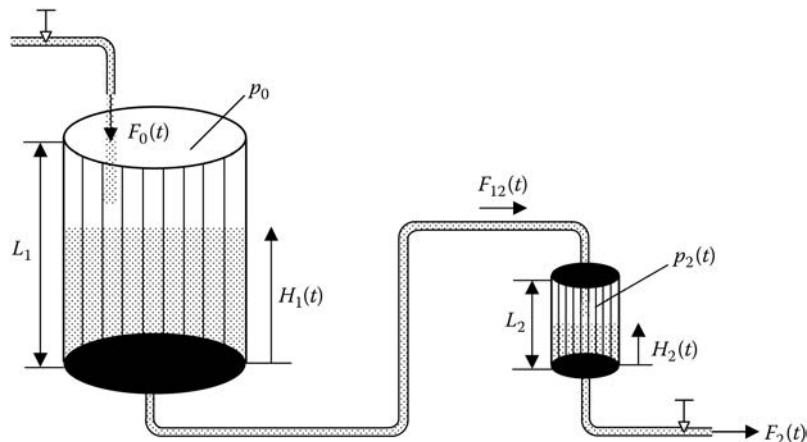


FIGURE 8.56 System of two different capacity tanks in series.

where

A_1 is the cross-sectional area

L_1 is the height of the first tank

The flow from the first tank into the second tank $F_{12}(t)$ depends on the pressure differential between the bottom of the first tank and the top of the second tank.

$$F_{12}(t) = \begin{cases} c_1[p_0 + \gamma H_1(t) - \gamma L_2 - p_2(t)]^{1/2}, & p_0 + \gamma H_1(t) - \gamma L_2 - p_2(t) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (8.287)$$

where

c_1 is a constant related to the fluid resistance in the line connecting the tanks

p_0 is atmospheric pressure (14.7 psi)

γ is the specific weight of water (62.4 lb/ft³)

The air pressure above the liquid in the sealed tank $p_2(t)$ is related to the liquid level $H_2(t)$ according to

$$p_2 = \left(\frac{L_2}{L_2 - H_2} \right) p_0 \quad (8.288)$$

Equation 8.288 assumes that the air pressure in the sealed tank obeys the relationship $p_2 V_2 = \text{constant}$ and that $p_2 = P_0$ when $H_2 = 0$. Hence,

$$p_0 A_2 L_2 = p_2 A_2 (L_2 - H_2) \quad (8.289)$$

The differential equation for the second tank is

$$A_2 \frac{d}{dt} H_2(t) + F_2(t) = F_{12}(t), \quad H_2(t) \leq L_2 \quad (8.290)$$

The flow out of the second tank is governed by the algebraic relation

$$F_2(t) = c_2[(p_2(t) + \gamma H_2(t)) - p_0]^{1/2} \quad (8.291)$$

where c_2 is a constant related primarily to the physical construction of the valve in the discharge line and its percent opening.

Stiffness is a property of linear systems relating the magnitudes of the fast poles (eigenvalues) to the slower poles. We can linearize the nonlinear system modeled by Equations 8.286 through 8.291 about a steady-state operating point (\bar{H}_1, \bar{H}_2) corresponding to a constant input flow $F_0(t) = \bar{F}_0, t \geq 0$.

At steady state, $\bar{F}_2 = F_2(\infty) = \bar{F}_0$ and it follows from Equation 8.291

$$\bar{F}_2 = c_2[\bar{p}_2 + \gamma \bar{H}_2] - p_0]^{1/2} = \bar{F}_0 \quad (8.292)$$

$$\Rightarrow c_2 \left[\left(\frac{L_2}{L_2 - \bar{H}_2} \right) p_0 + \gamma \bar{H}_2 - p_0 \right]^{1/2} = \bar{F}_0 \quad (8.293)$$

Rearranging Equation 8.293 leads to a quadratic equation in \bar{H}_2 ,

$$\gamma\bar{H}_2^2 - \left[p_0 + \gamma L_2 + \left(\frac{\bar{F}_0}{c_2} \right)^2 \right] \bar{H}_2 + \left(\frac{\bar{F}_0}{c_2} \right)^2 L_2 = 0 \quad (8.294)$$

It is left as an exercise to show that the steady-state operating level in the first tank is

$$\bar{H}_1 = L_2 + \frac{1}{\gamma} \left[\left(\frac{\bar{F}_0}{c_1} \right)^2 + \left(\frac{\bar{H}_2}{L_2 - \bar{H}_2} \right) p_0 \right] \quad (8.295)$$

Given \bar{F}_0 , Equations 8.294 and 8.295 can be solved in that order to find the operating point levels \bar{H}_2, \bar{H}_1 and ultimately the remaining dependent variable operating point values, namely, \bar{F}_{12}, \bar{F}_2 , and \bar{p}_2 .

The nonlinear system model in Equations 8.286 through 8.291 can be reduced to

$$\frac{dH_1}{dt} = f_1(H_1, H_2, F_0) \quad (8.296)$$

$$= \frac{1}{A_1} \left[F_0 - c_1 \left\{ \gamma(H_1 - L_2) - \left(\frac{H_2}{L_2 - H_2} \right) p_0 \right\}^{1/2} \right], \quad H_1 \leq L_1 \quad (8.297)$$

$$\frac{dH_2}{dt} = f_2(H_1, H_2, F_0) \quad (8.298)$$

$$= \frac{1}{A_2} \left[c_1 \left\{ \gamma(H_1 - L_2) - \left(\frac{H_2}{L_2 - H_2} \right) p_0 \right\}^{1/2} - c_2 \left\{ \left(\frac{H_2}{L_2 - H_2} \right) p_0 + \gamma H_2 \right\}^{1/2} \right] \quad (8.299)$$

The linearized state model is

$$\frac{d}{dt} \underline{\Delta H}(t) = A \underline{\Delta H}(t) + B \Delta F_0(t) \quad (8.300)$$

$$\underline{\Delta y}(t) = C \underline{\Delta H}(t) + D \Delta F_0(t) \quad (8.301)$$

where

$$\underline{\Delta H}(t) = \begin{bmatrix} \Delta H_1(t) \\ \Delta H_2(t) \end{bmatrix} = \begin{bmatrix} H_1(t) - \bar{H}_1 \\ H_2(t) - \bar{H}_2 \end{bmatrix} \quad (8.302)$$

$$\underline{\Delta y}(t) = \begin{bmatrix} \Delta H_1(t) \\ \Delta H_2(t) \\ \Delta F_{12}(t) \\ \Delta F_2(t) \\ \Delta p_2(t) \end{bmatrix} = \begin{bmatrix} H_1(t) - \bar{H}_1 \\ H_2(t) - \bar{H}_2 \\ F_{12}(t) - \bar{F}_{12} \\ F_2(t) - \bar{F}_2 \\ p_2(t) - \bar{p}_2 \end{bmatrix} \quad (8.303)$$

The coefficient matrix A comprises the first partial derivatives

$$A_{11} = \frac{\partial}{\partial H_1} f_1(\bar{H}_1, \bar{H}_2, \bar{F}_0) \quad (8.304)$$

$$= \frac{-\gamma c_1}{2A_1} \left[\gamma(\bar{H}_1 - L_2) - \left(\frac{\bar{H}_2}{L_2 - \bar{H}_2} \right) p_0 \right]^{-1/2} \quad (8.305)$$

$$A_{12} = \frac{\partial}{\partial H_2} f_1(\bar{H}_1, \bar{H}_2, \bar{F}_0) \quad (8.306)$$

$$= \frac{p_0 c_1}{2A_1} \left[\gamma(\bar{H}_1 - L_2) - \left(\frac{\bar{H}_2}{L_2 - \bar{H}_2} \right) p_0 \right]^{-1/2} \left\{ \frac{L_2}{(L_2 - \bar{H}_2)^2} \right\} \quad (8.307)$$

$$A_{21} = \frac{\partial}{\partial H_1} f_2(\bar{H}_1, \bar{H}_2, \bar{F}_0) \quad (8.308)$$

$$= \frac{\gamma c_1}{2A_2} \left[\gamma(\bar{H}_1 - L_2) - \left(\frac{\bar{H}_2}{L_2 - \bar{H}_2} \right) p_0 \right]^{-1/2} \quad (8.309)$$

$$A_{22} = \frac{\partial}{\partial H_2} f_2(\bar{H}_1, \bar{H}_2, \bar{F}_0) \quad (8.310)$$

$$= \frac{-p_0 L_2 c_1}{2A_2} \left[\gamma(\bar{H}_1 - L_2) - \left(\frac{\bar{H}_2}{L_2 - \bar{H}_2} \right) p_0 \right]^{-1/2} \left\{ \frac{1}{(L_2 - \bar{H}_2)^2} \right\}$$

$$- \frac{c_2}{2A_2} \left[\left(\frac{\bar{H}_2}{L_2 - \bar{H}_2} \right) p_0 + \gamma \bar{H}_2 \right]^{-1/2} \left\{ \frac{p_0 L_2}{(L_2 - \bar{H}_2)^2} + \gamma \right\} \quad (8.311)$$

The components of the input matrix B and output matrix C are obtained from partial derivatives as well (see Exercise 8.41).

EXAMPLE 8.8

The baseline numerical values of the system parameters are

$$\begin{aligned} R_1 &= 15 \text{ ft}, L_1 = 50 \text{ ft}, c_1 = 4 \text{ ft}^3/\text{min}/(\text{lb}/\text{ft}^2)^{1/2}, A_1 = \pi R_1^2 = 225\pi \text{ ft}^2 \\ R_2 &= 5 \text{ ft}, L_2 = 7.5 \text{ ft}, c_2 = 2 \text{ ft}^3/\text{min}/(\text{lb}/\text{ft}^2)^{1/2}, A_2 = \pi R_2^2 = 56.25\pi \text{ ft}^2 \end{aligned}$$

and the baseline inflow under steady-state operating conditions is $\bar{F}_0 = 60 \text{ ft}^3/\text{min}$. For the given baseline conditions,

- Find the steady-state operating point values $\bar{H}_1, \bar{H}_2, \bar{F}_1, \bar{F}_2$, and \bar{p}_2 .
- Compute the numerical values of the components of matrix A .
- Find the eigenvalues of A and compute the stiffness ratio.

- d. Draw a Simulink diagram for simulating the system dynamics.
- e. Use the MATLAB “linmod” function to approximate the matrices A , B , C , and D .
- f. Compare the linearized system response and the simulated response of the nonlinear system for the case where the system is initially in steady state and the inflow to the first tank is given by

$$F_0(t) = \begin{cases} \bar{F}_0, & t \leq 50 \\ \bar{F}_0 - 5, & t > 50 \text{ min} \end{cases} \quad (8.312)$$

- g. Determine the new steady-state levels in both tanks predicted by the nonlinear model and the linearized model.
- a. The steady-state operating levels are obtained from Equations 8.294 and 8.295 in the M-file “Ch8_Ex8_8.m.” The results are

$$\bar{H}_1 = 23.52 \text{ ft}, \bar{H}_2 = 2.01 \text{ ft}$$

$$\bar{F}_{12} = \bar{F}_2 = 60 \text{ ft}^3/\text{min}$$

$$\bar{p}_2 = 2891.4 \text{ lb/ft}^2$$

- b. The same M-file contains code for evaluating the components of matrix A using Equations 8.304 through 8.311. The result is

$$A = \begin{bmatrix} -0.0118 & 0.0993 \\ 0.1059 & -1.1440 \end{bmatrix}$$

- c. The eigenvalues of A are $\lambda_1 = -0.00255151$ and $\lambda_2 = -1.15318422$, and the stiffness ratio of the system linearized about the given steady-state operating point is

$$\frac{\lambda_2}{\lambda_1} = \frac{-1.15318422}{-0.00255151} = 451.96$$

The time constants of the linearized system are

$$\tau_1 = \frac{-1}{\lambda_1} = \frac{-1}{-0.00255151} = 391.92 \text{ min}$$

$$\tau_2 = \frac{-1}{\lambda_2} = \frac{-1}{-1.15318422} = 0.867 \text{ min}$$

demonstrating the dual time scales involved.

- d. A Simulink diagram is shown in [Figure 8.57](#).
- e. MATLAB statements in “Ch8_Ex8_8.m” for employing the “linmod” function are

```
[sizes, X0, states] = TwoTanks ([] , [] , [] , 0)
H_opert = [H1_ss; H2_ss];
u0 = F0;
[A, B, C, D] = linmod('TwoTanks_linmod', H_opert, u0)
```

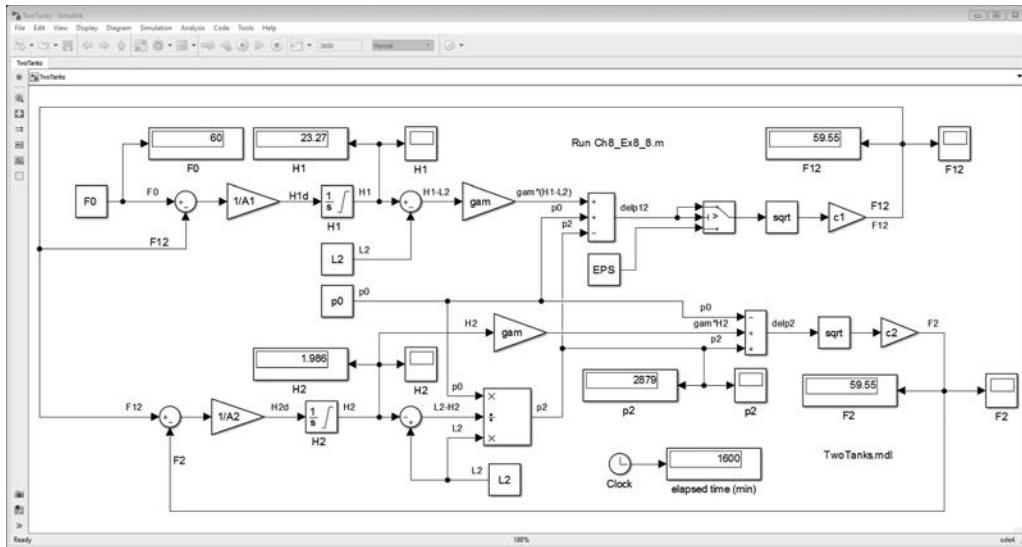


FIGURE 8.57 Simulink diagram for simulation of two-tank system with stiff dynamics.

The first line returns the variable “states,” which identifies the limited integrator outputs “H1” and “H2” as the first and second states, respectively. The last line refers to a Simulink model file “TwoTanks_linmod.mdl,” which is similar to “TwoTanks.mdl” shown in Figure 8.57 except an input port block replaces the “Constant” block with parameter “F0” and the addition of five output port blocks to identify the system outputs. The last line produces the linearized system matrices

$$A = \begin{bmatrix} -0.0118 & 0.0993 \\ 0.1059 & -1.1440 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0014 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 8.3200 & -70.2135 \\ 0 & 19.6334 \\ 0 & 526.6010 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Note that the coefficient matrix A using “linmod” is identical (to at least four places after the decimal point) to the previous result based on the analytical expressions for the partial derivatives in Equations 8.304 through 8.311.

- f. The Simulink diagram in Figure 8.57 is supplemented with additional blocks to generate the deviation input variable $\Delta F_0(t) = F_0(t) - \bar{F}_0$ into a “state-space” block with output $\Delta y(t) = [\Delta H_1(t) \Delta H_2(t) \Delta F_{12}(t) \Delta F_2(t) \Delta p_2(t)]^T$. The linearized system outputs $H_1(t) = \bar{H}_1 + \Delta H_1(t)$ and $H_2(t) = \bar{H}_2 + \Delta H_2(t)$ are compared with the simulated nonlinear system responses in Figures 8.58 and 8.59. RK-4 simulation with a short time step $T = 0.1$ s was used to generate accurate approximations of the nonlinear responses. The Simulink model file is “TwoTanks_NL_and_L.mdl.” The linearized responses are approaching steady state after 1500 min in agreement with the larger time constant of 391.92 min.
- g. The new steady-state levels established when the inflow is held constant at $55 \text{ ft}^3/\text{min}$ are obtained from Equations 8.294 and 8.295. The result is $(H_1)_{ss} = 20.89 \text{ ft}$ and $(H_2)_{ss} = 1.76 \text{ ft}$. From Equation 8.300 at steady state,

$$\Delta H_{ss} = -A^{-1}B\Delta F_0 \quad (8.313)$$

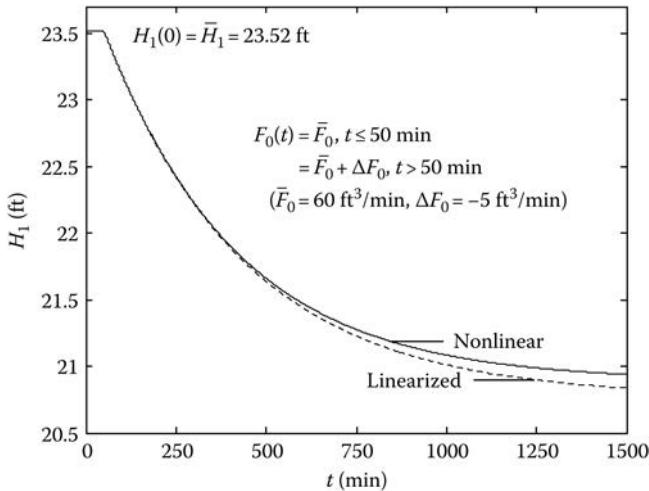


FIGURE 8.58 Tank 1 nonlinear and linearized system level responses.

$$\begin{aligned}
 &= -\begin{bmatrix} -0.0118 & 0.0993 \\ 0.1059 & -1.1440 \end{bmatrix}^{-1} \begin{bmatrix} 0.0014 \\ 0 \end{bmatrix} \begin{bmatrix} -5 \\ -5 \end{bmatrix} = \begin{bmatrix} -2.7501 \\ -0.2547 \end{bmatrix} \\
 \Rightarrow (H_1)_{ss} &= \bar{H}_1 + (\Delta H_1)_{ss} = 23.52 + (-2.75) = 20.77 \\
 \Rightarrow (H_2)_{ss} &= \bar{H}_2 + (\Delta H_2)_{ss} = 2.01 + (-0.25) = 1.76 \text{ ft}
 \end{aligned}$$

It is clear from Figures 8.58 and 8.59 that the linearized system approximation to the nonlinear system about the given steady-state operating point is more than adequate when the perturbation in $F_0(t)$ about \bar{F}_0 is limited to $-5 \text{ ft}^3/\text{min}$.

Figure 8.60 contains graphs showing how the steady-state operating point levels $\bar{H}_1, H_2, \bar{p}_2$ vary with changes in the flow \bar{F}_0 . The baseline operating point in Example 8.8 with $\bar{F}_0 = 60 \text{ ft}^3/\text{min}$ is also shown.

The stiffness of the linearized system is shown in the top left corner. Note how the stiffness increases from a little over 200 to around 950 before the first tank starts to overflow

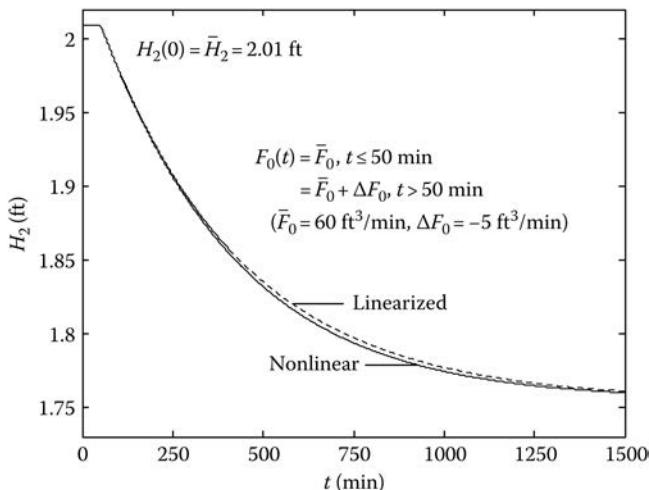


FIGURE 8.59 Tank 2 nonlinear and linearized system level responses.

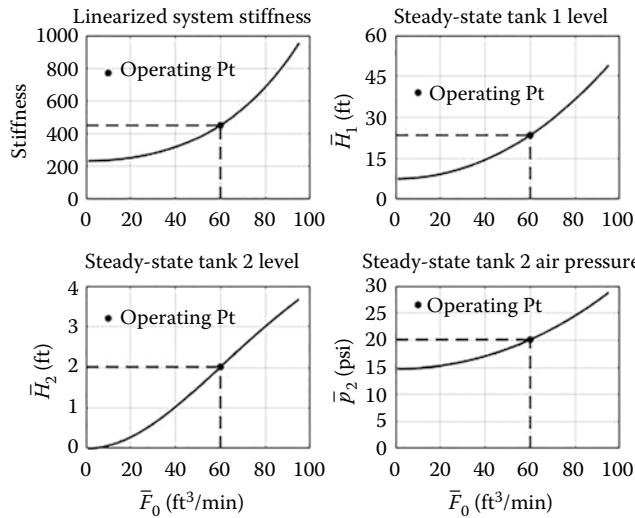


FIGURE 8.60 Graph of linearized system stiffness and nonlinear system steady-state operating characteristics.

when the level reaches $L_1 = 50$ ft. At that point, the linearized system eigenvalues are -0.0015 and -1.4724 , resulting in natural modes with time constants of approximately 0.679 and 645.8 min. The large difference in time constants of the linearized system results primarily from the significant disparity in the capacities of the two tanks.

8.4.7 MULTIRATE SIMULATION OF TWO-TANK SYSTEM

In view of the large difference between the linearized system time constants, multirate integration offers the possibility of reducing simulation execution time without significant loss of accuracy. The first step is to choose the “master” and “slave” integration routines and determine the slow and fast frame times. The aircraft pitch example used the one-step RK-4 for “master” and “slave.” For this example, the multistep AB-2 integrator will be used to integrate both the slow and fast states.

AB-2 integration is a popular numerical integrator, particularly in applications involving ground vehicle, aircraft, missile, ship, power plant, and chemical process simulators where a real-time solution of the model equations is required. Real-time numerical integration is the subject of the following section.

Looking at the AB stability regions in Figure 8.21 of the previous section, the simulation step size T is limited by the condition $\lambda T = -1$ for AB-2 integration of a linear first-order continuous-time system with characteristic root λ . Consequently, for small changes from the baseline operating point of the two-tank system, that is, $(\bar{F}_0 = 60 \text{ ft}^3/\text{min}; \bar{H}_1 = 23.52 \text{ ft}, \bar{H}_2 = 2.01 \text{ ft})$, AB-2 simulation will be stable provided

$$\lambda T = (-1.1532)T < -1 \Rightarrow T < 0.8672 \text{ min}$$

Figure 8.61 shows the results of AB-2 simulation of the system when the inflow $F_0(t) = 55 \text{ ft}^3/\text{min}$, $t \geq 0$.

The initial tank levels are the steady-state values when $\bar{F}_0 = 60 \text{ ft}^3/\text{min}$. The two plots on the left are the result of selecting the step size $T = 0.1 \text{ min}$ while the graphs on the right correspond to an integration step of $T = 0.87 \text{ min}$, just slightly larger than the upper limit for AB-2 stability. The unstable nature of both tank level responses when $T = 0.87 \text{ min}$ is apparent. The unstable responses

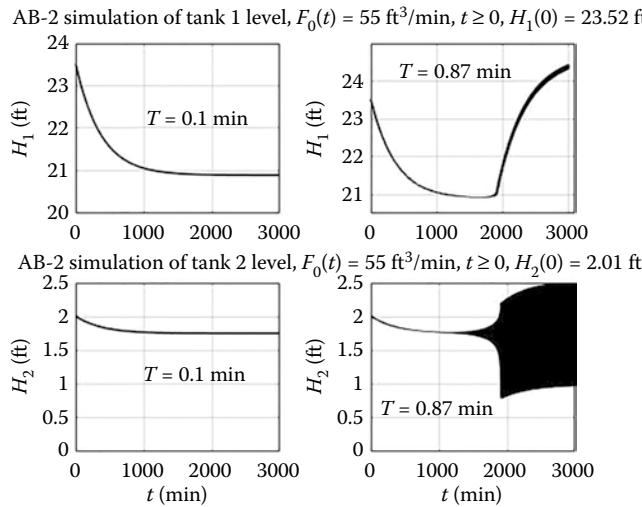


FIGURE 8.61 Stable and unstable AB-2 simulation for $H_1(0) = \bar{H}_1$, $H_2(0) = \bar{H}_2$.

are similar to the stable transient responses up to a point. All graphs were generated in M-file “*Ch8_TwoTanks_AB2.m*.”

The next set of graphs in Figure 8.62 illustrate AB-2 simulation of tank level responses when both tanks are initially empty and the inflow is a step input described by $F_0(t) = \bar{F}_0 = 60 \text{ ft}^3/\text{min}, t \geq 0$. The fluid level $H_2(t)$ remains at zero until the first tank level.

$H_1(t)$ reaches a height of $L_2 = 7.5 \text{ ft}$, that is, high enough to push fluid from the bottom of the first tank up to the top of the second tank. Thus, $F_{12}(t) = 0$ as long as $H_1(t) < L_2$.

Due to the magnitude of the step, the system variables $H_1(t)$ and $H_2(t)$ are not confined to a small region about the initial steady-state operating point $(\bar{H}_1, \bar{H}_2) = (0, 0)$. Consequently, a single linearized model to accurately predict deviations in both levels is not valid, and the discussion in Section 7.4 dealing with multiple linearized models is applicable.

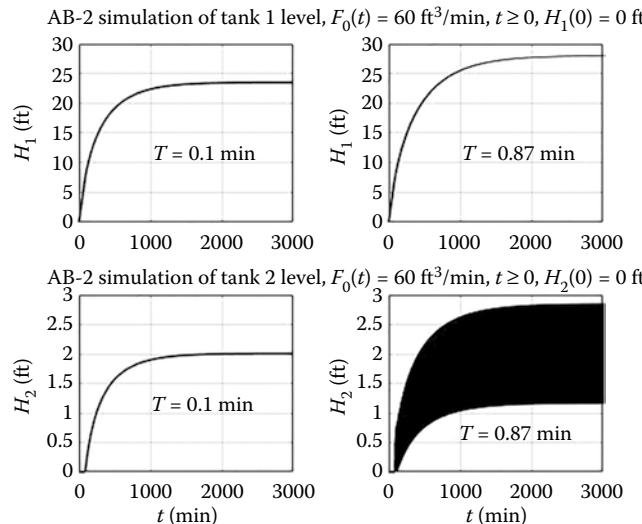


FIGURE 8.62 Stable and unstable AB-2 simulation for $H_1(0) = H_2(0) = 0$.

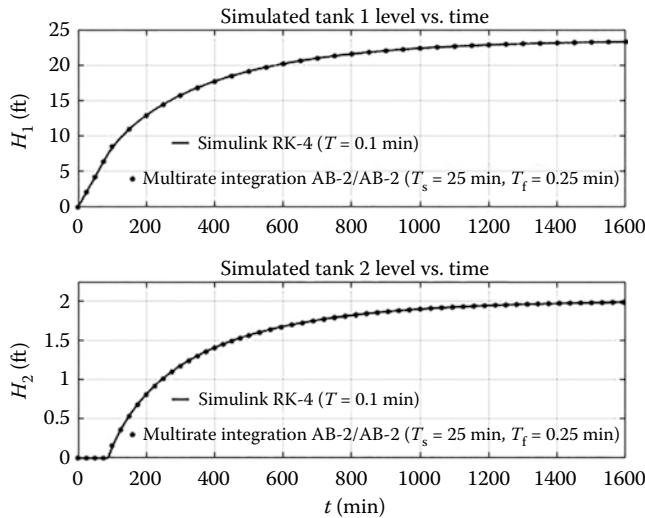


FIGURE 8.63 Multirate simulation (AB-2/AB-2) of nonlinear two-tank system ($T_s = 25$ min, $T_f = 0.25$ min, $N = 100$).

Multirate integration of the nonlinear two-tank system with AB-2 integration as the “master” routine and AB_2 integration as the “slave” routine is straightforward to implement (see “*Ch8_TwoTanks_Multirate_AB2.m*”). Time histories of each tank level when the fast frame time $T_f = 0.25$ min and the slow frame time $T_s = 25$ min (frame ratio $N = T_s/T_f = 100$) are shown in Figure 8.63. Note that every 200th point in the $H_2(t)$ response is plotted in the lower graph.

Results from another multirate simulation run are shown in Figure 8.64. The fast state $H_2(t)$ was updated using AB-2 integration with frame time $T_f = 0.1$ min while the slow state $H_1(t)$ was advanced by AB-2 integration every $T_s = 100$ min. The frame ratio was 1000. Every 200th point in the fast subsystem is plotted.

The solid lines in Figures 8.63 and 8.64 were obtained from Simulink RK-4 integration of the state equations with integration step size $T = 0.1$ min. Due to the small step size, they serve as

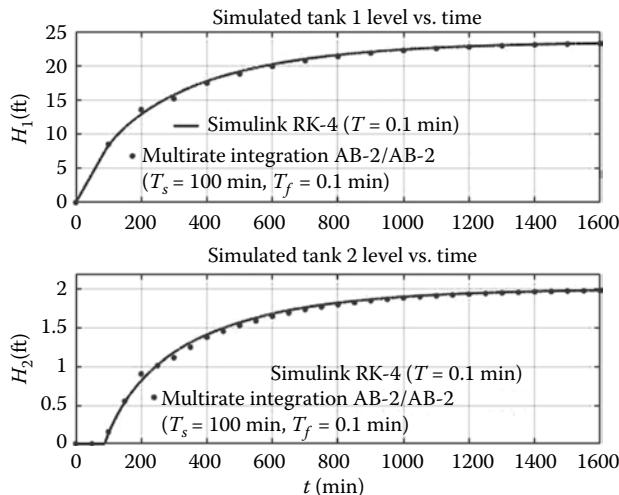


FIGURE 8.64 Multirate simulation (AB-2/AB-2) of nonlinear two-tank system ($T_s = 100$ min, $T_f = 0.1$ min, $N = 1000$).

accurate approximations to the exact solutions of the nonlinear state equations. Note that the AB-2/AB-2 response for $H_1(t)$ is quite accurate in both cases; however, the level response $H_2(t)$ is superior in the first case where the frame ratio is less. Exercise 8.45 explores the effect of frame ratio on the overall accuracy of the multirate simulation results.

8.4.8 SIMULATION TRADE-OFFS WITH MULTIRATE INTEGRATION

The case for multirate integration is based on the reduced number of derivative evaluations of the slow state required compared with the number of evaluations required when both slow and fast states are integrated at the same frame rate. The savings in execution time can be dramatic for high-order systems in which the majority of the state variables are associated with the slow subsystem. Even low-order systems experience significant reduction in simulation time when the slow derivatives are computationally more intensive. Be aware that real-world derivative functions often involve more than a few simple calculations. Logical branching, multidimensional lookup tables along with the sheer number of model equations to be evaluated contribute to the duration as well as uncertainty in the cpu time required to compute the state derivatives.

Without multirate integration, the total number of frames (integration steps) is given by t_{final}/T , where t_{final} is the simulation time and T is the integration step size. In the simplest case with only two states, fixed execution times of each derivative function and single-pass integration routines for “master” and “slave,” the reduction in execution time from implementing multirate integration is straightforward. Suppose the cpu times required to execute the slow and fast derivative functions are Δ_s and Δ_f , respectively.

Case I: Without multirate integration ($T_s = T_f = T$)

The derivatives are numerically integrated at the simulation frame rate ($1/T$). The total execution time for fast derivative evaluations is

$$\Gamma_f = \left(\frac{t_{final}}{T_f} \right) \Delta_f = \left(\frac{t_{final}}{T} \right) \Delta_f \quad (8.314)$$

with a similar expression for the time required to perform slow derivative calculations,

$$\Gamma_s = \left(\frac{t_{final}}{T_s} \right) \Delta_f = \left(\frac{t_{final}}{T} \right) \Delta_s \quad (8.315)$$

The total time to compute both fast and slow derivatives is therefore

$$\Gamma_{w/o} = \Gamma_f + \Gamma_s = \left(\frac{t_{final}}{T} \right) \Delta_f + \left(\frac{t_{final}}{T} \right) \Delta_s = \frac{t_{final}}{T} (\Delta_f + \Delta_s) \quad (8.316)$$

Case II: With multirate integration ($T_s = NT_f = NT$)

The total time required for both fast and slow derivatives is

$$\Gamma_w = \Gamma_f + \Gamma_s = \left(\frac{t_{final}}{T_f} \right) \Delta_f + \left(\frac{t_{final}}{T_s} \right) \Delta_s \quad (8.317)$$

$$= \left(\frac{t_{final}}{T} \right) \Delta_f + \left(\frac{t_{final}}{NT} \right) \Delta_s \quad (8.318)$$

$$= \frac{t_{final}}{T} \left(\Delta_f + \frac{\Delta_s}{N} \right) \quad (8.319)$$

Assuming cpu times to execute fast and slow derivative functions are related by

$$\Delta_s = \alpha \Delta_f, \quad \alpha > 0 \quad (8.320)$$

From Equations 8.319 and 8.320,

$$\Gamma_w = \frac{t_{final}}{T} \Delta_f \left(1 + \frac{\alpha}{N} \right) \quad (8.321)$$

The cpu time (in seconds) required to evaluate two state derivatives using single-pass, multirate integration is illustrated in [Figure 8.65](#) for the case where the transient response requires $t_{final}/T = 100,000$ simulation frames. This number of integration steps would be required, for example, if the step size needed to satisfy numerical stability and dynamic accuracy requirements was $T = 0.01$ s and the transient response lasted for 1000 s. The cpu time to execute the fast state derivative function Δ_f was fixed at 100 μ s, and the slow state derivative requires $\alpha \Delta_f \mu$ s where α ranges from 0.5 to 2.5.

Observe from [Figure 8.65](#) that the total cpu time without multirate integration varies from a low of 15 s when $\Delta_s = 0.5\Delta_f$ to a high of 35 s when $\Delta_s = 2.5\Delta_f$. The reduction in cpu time is more pronounced for lower values of frame ratio, that is, $N \leq 10$. Also, note that when the fast and slow state derivatives require the same amount of cpu time to execute, that is, $\alpha = 1$, the savings in overall cpu time is reduced from 20 min down to the limiting value of 10 min as expected.

The reduction in cpu time for the conditions illustrated in [Figure 8.65](#) may seem trivial. The largest reduction in cpu time only approaches 25 s for the case where $\alpha = 2.5$ and the frame ratio is large. Simulation studies often entail multiple simulation runs with one or more system parameters varying from run to run. A two-parameter sensitivity study where each parameter assumes 10 numerical values requires 100 simulation runs. In this scenario, the use of multirate integration can achieve significant savings in overall computational time at the slight expense of reduced accuracy in the simulated responses.

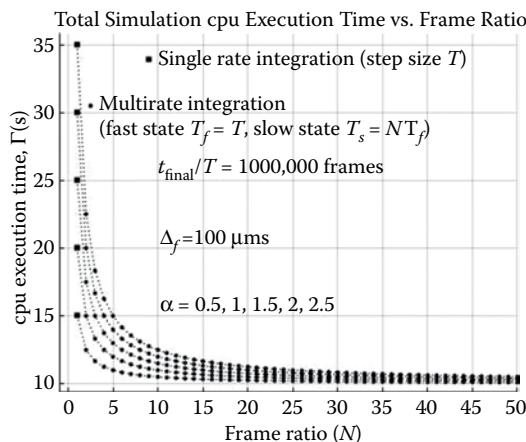


FIGURE 8.65 Total cpu time required to simulate transient response of system with single rate ($N = 1$) and multirate ($N > 1$) integration.

EXERCISES

- 8.31 Derive the state equation matrices A , B , C , and D given in Equations 8.240 and 8.241.
- 8.32 In the aircraft pitch control system, find the maximum step size allowable for stable RK-4 simulation of the slow subsystem second-order component with poles located at $-4.3127 \pm j6.4812$.
- 8.33 In the aircraft pitch control system, use MATLAB to find
- The analytical solution for state variable $x_3(t)$ and compare with the simulated results obtained with Simulink RK-4 and multirate RK-4/RK-4
 - The analytical solution for the pitch rate $\dot{\theta}(t)$ and compare it with the simulated response obtained from Simulink using RK-4
- 8.34 For the aircraft pitch control system represented by the block diagram shown in Figure 8.48,
- Draw a simulation diagram and label the states $x_1, x_2, x_3, \dots, x_7$.
 - Write the state equations and find the matrices A , B , C , and D in $\dot{x} = Ax + B\theta_{com}$, $y = Cx + D\theta_{com}$. The output is $y(t) = \theta(t)$. Leave your answer in terms of parameters $A_1, B_1, B_0, b_1, b_0, \dots, E_1$.
 - Using the given baseline values for the control system parameters K, K_c, \dots, K_b , evaluate the matrices A, B, C , and D .
 - Use MATLAB to verify that the eigenvalues of the coefficient matrix A are identical to those of the matrix A in Equation 8.240. Compare the eigenvalues to the roots of the characteristic polynomial in Equation 8.251.
 - Supplement the diagram shown in Figure 8.46 with additional Simulink blocks to simulate the pitch response based on the block diagram in Figure 8.48. Plot the three pitch responses on the same graph.
- 8.35 Label the five inputs to the summer in Figure 8.48 as $\theta_1, \theta_2, \dots, \theta_5$. Find and plot the analytical solutions for $\theta_1(t), \theta_2(t), \dots, \theta_5(t)$, on the same graph in response to the command pitch input in Equation 8.250. Comment on the results.
- 8.36 Simulate the aircraft control system pitch response to the input given in Equation 8.250 using multirate integration with $T_f = 0.001$ s and $T_s = 0.02$ s. Choose RK-1 for the “slave” routine and RK-4 for the “master” integration. Plot the response along with the analytical solution.
- 8.37 Consider the aircraft pitch control system operating in regulator mode, that is, zero input and initial condition $\theta(0) = \theta_0$.
- Find analytical solutions for the pitch response $\theta(t)$ and the elevator deflection $\delta_e(t)$ when $\theta_0 = 10^\circ$.
 - Find T_{max} , the maximum integration step for a stable simulation using RK-2 integration.
 - Simulate the pitch and elevator responses of the regulator control system ($\theta_0 = 10^\circ$) using Simulink with RK-2 integration. Choose the step size $T = 0.1 T_{max}$.
 - Simulate the pitch and elevator responses of the regulator control system using RK-2/RK-2 multirate integration. Choose the fast frame time T_f so that the characteristic error in damping ratio of the fast subsystem second-order component in Figure 8.48 is 0.1%. Round T_f to three places after the decimal point. Choose the slow frame time T_s to make the frame ratio $N = T_s/T_f = 10$.
 - Plot the three pitch responses (analytical and two simulated) on the same graph. Repeat for the three elevator responses.
- 8.38 In the aircraft pitch control system, find the analytical solution for the fast state variables $x_5(t)$, $x_6(t)$, and $x_7(t)$ and plot the analytical, Simulink RK-4 and multirate RK-4/RK-4 solutions on the same graph similar to Figure 8.55.
- 8.39 Run the multirate integration of the aircraft pitch control system in the M-file “*Ch8_multi_rate_integ.m*” for the cases where the frame ratio $N = 20, 10, 5, 1$, and plot the simulated and analytical responses for $x_1(t) = \theta(t)$, $\dot{\theta}(t)$, and $x_4(t) = \delta_e(t)$.

- 8.40 Derive Equation 8.295 for the steady-state operating level in the first tank.
- 8.41 Find analytical expressions in terms of the system parameters A_1 , c_1 , A_2 , L_2 , and c_2 and the steady-state levels \bar{H}_1, \bar{H}_2 for the components of matrices B and C in Equations 8.300 and 8.301. Evaluate B and C for the given baseline values of the system parameters when $\bar{F}_0 = 60 \text{ ft}^3/\text{min}$, and compare your results with those given in the text.
- 8.42 Generate responses similar to those in Figure 8.58 for the case where the initial conditions correspond to an input flow $F_0(t) = \bar{F}_0 = 20 \text{ ft}^3/\text{min} = 20 \text{ ft}^3/\text{min}$. The inflow suddenly increases by $\Delta F_0(t) = 2.5 \text{ ft}^3/\text{min}$ at $t = 50 \text{ min}$.
- 8.43 Plot an \bar{H}_1 vs. \bar{H}_2 operating characteristic for the two-tank system.
Hint: Vary \bar{F}_0 from zero until the first tank begins to overflow. Find the steady-state values for \bar{H}_1 and \bar{H}_2 .
- 8.44 Use AB-2 integration with step size T to simulate and plot the fluid level responses of both tanks like the ones shown in Figures 8.58 and 8.59 for $T = 0.05, 0.1, \dots, 1.0$. Comment on the results.
- 8.45 For the baseline nonlinear two-tank system with tanks initially empty and tank one inflow given by $F_0(t) = 75 \text{ ft}^3/\text{min}$, $t \geq 0$.
- Run the Simulink model “*TwoTanks.mdl*” using RK-4 integration for a simulated time of 1500 min with decreasing step sizes T until there is negligible change in output for consecutive runs. Save the simulated tank levels at the end of each minute and denote them $H_{1,A}(n), H_{2,A}(n)$, $n = 0, 1, 2, \dots, 1500$. Assume that the simulated values are exact, that is, $H_{1,A}(n) \approx \hat{H}_{1,A}(nT)$, $H_{2,A}(n) \approx \hat{H}_{2,A}(nT)$, $n = 0, 1, 2, \dots, 1500$.
 - Run the MATLAB M-file “*Chap8_TwoTanks_Multirate_AB2_AB2.m*” or write your own to implement multirate AB-2/AB-2 integration for a simulated time of 1500 min with fixed frame time $T_f = 0.1 \text{ min}$. Let the frame ratio $N = T_f/T$ vary according to 1, 5, 10, 15, 20, 25, 50, 75, 100, 500, 1000 and denote tank levels at the end of each minute by $\hat{H}_{1,A}(n), \hat{H}_{2,A}(n)$, $n = 1, 2, \dots, 1500$. Compute the mean squared errors for each value of N as

$$E_{H_1}(N) = \frac{1}{1500} \sum_{n=0}^{1500} \left[\left\{ \hat{H}_{1,A}(n) - H_{1,A}(n) \right\}^2 \right]$$

$$E_{H_2}(N) = \frac{1}{1500} \sum_{n=0}^{1500} \left[\left\{ \hat{H}_{2,A}(n) - H_{2,A}(n) \right\}^2 \right]$$

- c. Plot $E_{H_1}(N)$ and $E_{H_2}(N)$ vs. N and comment on the results.
- 8.46 Eight of ten natural modes of a linear 10th-order system are slow in comparison with the remaining two natural modes. The average cpu time required to compute the slow and fast state derivatives is 12 and $0.3 \mu\text{s}$, respectively. A multirate integration scheme is proposed to simulate the transient response using RK-4 to integrate the slow states and RK-2 for the fast states. The fast states are updated at a rate of 250 Hz to assure numerical stability and reasonable dynamic accuracy. The dominant mode of the system corresponds to a real pole at $s = -0.05$.
- A simulation study to investigate the effect of three parameters calls for $10 \times 10 \times 10$ simulation runs. Generate a graph like the one shown in Figure 8.65 relating the total simulation study cpu time vs. the multirate integration frame ratio.

8.5 REAL-TIME SIMULATION

Until now, the simulation execution time required by whatever computer resources might be available to update the state and algebraic variables of the system received minimal attention. A simulation

study could “run long” for a number of reasons such as model complexity, dynamic accuracy and numerical stability requirements, limited cpu processing capabilities, and so forth; however, the consequences of waiting on the simulation to complete were not a critical concern. Simulations of this nature fall in the category of “off-line,” “batch,” or, more generally, nonrealtime simulation.

In some real-time simulations, a component that may have been simulated in the past has been replaced by the actual hardware. Alternatively, the component of interest may be physically integrated into the simulation from the beginning, making it unnecessary to simulate it beforehand. The component could be a gyroscopic sensor, a control surface actuator, an autopilot, or a combination of various sensors, actuators, and controllers in a particular system. The hardware must communicate with the simulation computer at precise intervals of time. The situation, illustrated in [Figure 8.66](#), is referred to as “hardware-in-the-loop” simulation or **HIL** simulation for short.

HIL is used extensively in the development and testing of missile systems. Missile sensors are stimulated with input signals generated from real-time control computers representing the motion of targets during an engagement. Guidance and control hardware respond by providing inputs to the missile flight dynamics model, which is simulated in real-time to determine the missile’s trajectory and calculate target intercept conditions.

The automotive industry incorporates real-time **HIL** simulation to design and test electronic control units (ECUs) for efficient operation of key systems such as power train control, the antilock braking system (ABS), and traction and cruise control. Classical simulation was performed off-line using simulation models of the vehicle’s dynamics, sensors, and ECUs. While it was beneficial to demonstrate interaction of the various components and subsystems, it was still necessary to evaluate an ECU design using expensive prototype vehicles on a test track. Reproducing test track conditions to investigate unexpected results posed additional challenges.

One solution was to use **HIL** simulation composed of a real-time computer that runs a model of the vehicle to be controlled and the input/output (I/O) interfaces required to electrically connect to the controller. Benefits include a reduction in control system development and testing, no need for expensive prototype vehicles, elimination of risk that improper control software could lead to a hazardous failure during a test track run, and no concern about test track interactions with a prototype vehicle.

[Figure 8.67](#) shows the main components of an **HIL** implementation for testing an ABS controller used by the German automaker Audi (Hanselmann). A digital-to-analog (D2A) converter generates wheel speed signals, sinusoidal voltages proportional in both frequency and amplitude to wheel speed, that replaces those from magnetic sensors in the actual vehicle. This accounts for the “interface-to-hardware” component in [Figure 8.66](#). The “interface from hardware” consists of an analog-to-digital (A2D) converter for generating pressure sensor signals (in digital form) required by the vehicle dynamics model in the simulation computer to simulate the vehicle’s response. Steering angle and other signals shown in [Figure 8.67](#) are used for testing advanced levels of vehicle

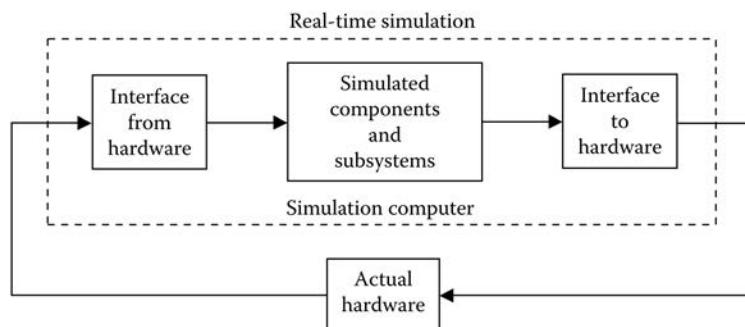


FIGURE 8.66 Hardware-in-the-loop real-time simulation.

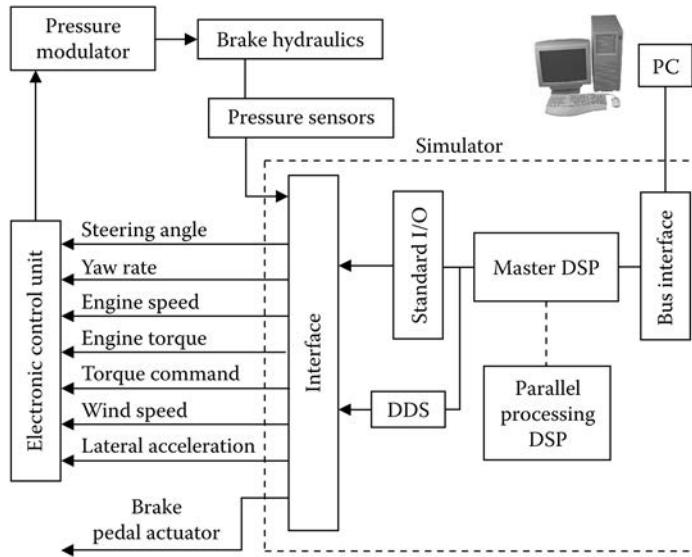


FIGURE 8.67 HIL simulation of vehicle ABS system. (From Hanselman, H. and Smith, K., *Test Meas. World Manage.*, 35, 1996.)

dynamics control such as automated braking on individual wheels at different intensity levels to stabilize vehicle motion in extreme situations.

It is imperative that the simulation computer be able to integrate the state variables in synchronization with real time. The beginning of each integration step must be properly aligned with the corresponding point in real time. In other words, the simulation must be capable of running fast enough on the digital computer that the computed outputs, in response to real-time inputs, occur at the exact time these outputs would take place in the real world.

A realistic vehicle dynamics model consists of coupled algebraic and differential equations with lookup tables for evaluating certain vehicle parameters that vary as driving conditions change. The equations are available as commercial C-language modules or in block diagram form (Simulink or other continuous simulation modeling program) with blocks representing transfer functions and system nonlinearities. Code is generated automatically from the block diagrams for real-time execution on the target DSP hardware.

Pushing the vehicle dynamics envelope to model evasive driving maneuvers adds to the complexity while imposing even more stringent timing requirements for numerically stable simulation of the differential equations. Additional mechanical degrees of freedom are present in the more detailed models used by Audi to account for slight movements in the vehicle's axles and suspension.

The time required to read input devices, perform simulation computations, and write to output devices determines the required frame rate for the simulation. In Audi's case, the simulation frame rate is less than 1 ms. The simulator generates signals and communicates them to the ECU in a matter of microseconds.

The simulated portion of the system in an HIL simulation may be all continuous-time, all discrete-time, or a combination of both. Furthermore, other types of signals, other than analog, are frequently encountered in HIL simulation. It is not uncommon for actual hardware to communicate with the simulation computer via I/O devices involving discrete digital (TTL), serial (RS-232, RS-422), instrumentation bus (IEEE-488) or network (Ethernet) signals (Ledin 2001).

Sometimes, the hardware in Figure 8.66 is actually a human such as a pilot in a flight simulator or an operator in a power plant simulator. A "human-in-the-loop" simulation can be used to evaluate the dynamic response of the system, the effectiveness of instrumentation displays and controls, or as a trainer to instruct the human in routine and emergency operation of the system. In the case of

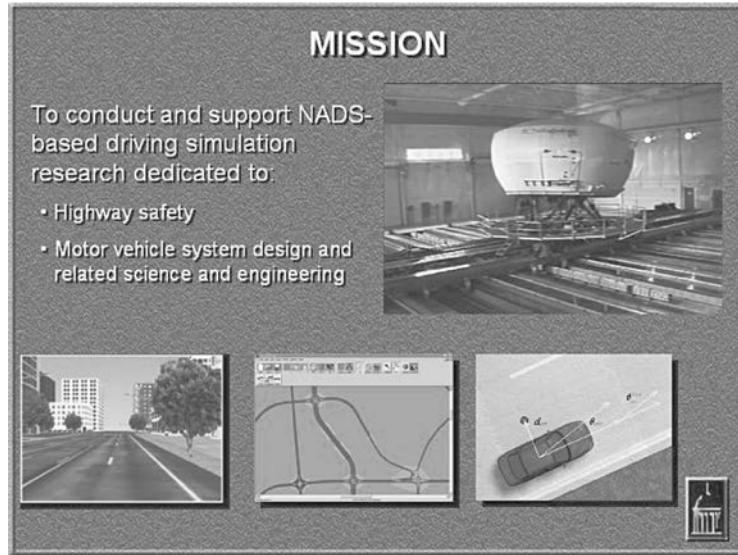


FIGURE 8.68 The National Advanced Driving Simulator used for Traffic Engineering Research and Vehicle System Design. (Courtesy of NHTSA, Washington, DC.)

real-time interactive simulators (vehicle, aircraft, train, ship, plant, and so forth), several channels of output from the simulation computer may be used to drive motion systems as well as audio and visual displays to provide additional cues designed to enhance the overall sense of being physically immersed in a realistic, high-fidelity simulation environment. [Figure 8.68](#) is a picture of a high-fidelity-driving simulator used for conducting research in traffic engineering, human factors, and design of new vehicle systems.

8.5.1 NUMERICAL INTEGRATION METHODS COMPATIBLE WITH REAL-TIME OPERATION

The timing issues inherent in real-time simulation preclude the use of variable-step methods, which adaptively regulate the integration step size. The iterative nature of implicit methods makes the solution times unpredictable and, therefore, unsuitable for real-time applications as well. We will begin by looking at several one-step RK integrators (see Section 6.2) and determine whether they are compatible with real-time simulation. A continuous-time dynamic system is assumed to be modeled by the scalar, possibly nonlinear differential equation

$$\frac{dx}{dt} = f(x, u) \quad (8.322)$$

where

$x = x(t)$ is the state

$u = u(t)$ is the single input

For simplicity, the derivative function is assumed not to be an explicit function of “ t .” If the system is time-varying, the derivative function should be expressed as $f(t, x, u)$.

[Figure 8.69](#) illustrates the sequence of operations for a real-time simulation running at a basic frame rate of $1/T$ using an integrator requiring two passes, that is, two derivative function evaluations per frame. The initial operation is an I/O read of the input $u_n = u(t_n)$. The next process is the two derivative function evaluations including the calculations to advance the state from $x_A(n)$ to

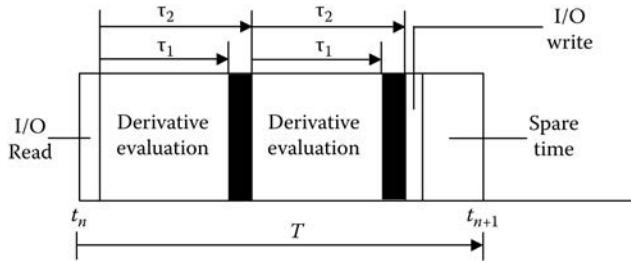


FIGURE 8.69 Real-time simulation with two-pass numerical integration method.

$x_A(n + 1)$. Note that the time to evaluate the derivative function may be random due to the necessity of searching through tables of empirical data or as a result of branching when the code is executed. Lower and upper limits to compute the derivative functions and perform the calculations necessary for updating the state are τ_1 and τ_2 s. The final operation is an I/O write to the hardware interface as shown in Figure 8.66. The residual time before the frame ends is spare time to minimize the chances of a frame overrun and allow for expansion of the derivative function evaluation time should the model increase in complexity.

The following analysis of compatibility of real-time, single frame rate (as opposed to multirate) integration is based on the following assumptions:

1. The time required to complete the I/O read and write operations is negligible in comparison with the time to evaluate the derivative function and update the state.
2. The execution time to compute the derivative function is deterministic.
3. The spare time per frame is zero.

The net effect of these assumptions is that the frame time T is subdivided into m equal subframes where m is the number of passes through the derivative function. Several RK- m integrators will now be considered. In each instance, the test for compatibility with real-time simulation is whether or not the input $u(t)$ is needed at a point in time within the frame prior to it being available in real time.

8.5.2 RK-1 (EXPLICIT EULER)

The simplest of all the numerical integrators, explicit Euler, is compatible with real-time simulation because the input $u(t)$ is needed only at the beginning of the frame. Thus, updating the discrete-time state from x_n to x_{n+1} with RK-1 requires u_n be available at t_n , the start time of the n th frame, which is certainly true (see Figure 8.70). Note that x_n is short for $x_A(n)$, the discrete-time approximation to $x(t_n)$.

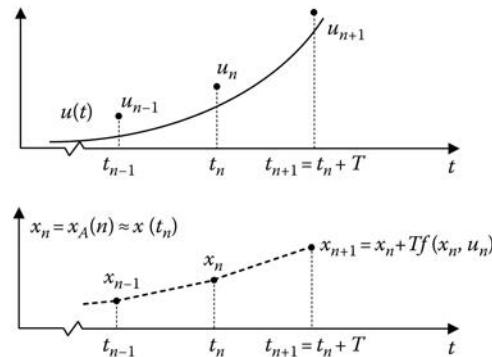


FIGURE 8.70 RK-1 (Euler) integration compatibility with real-time simulation.

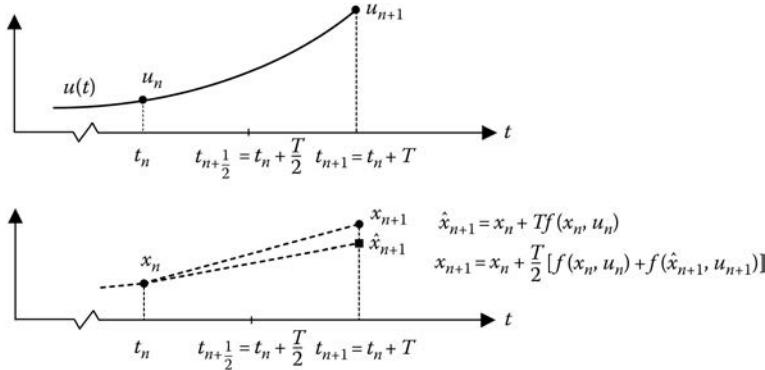


FIGURE 8.71 RK-2 (improved Euler) incompatibility with real-time simulation.

8.5.3 RK-2 (IMPROVED EULER)

Improved Euler RK-2 integration was introduced in Section 3.6 and again in Section 6.2. Figure 8.71 helps to explain why this one-step, two-pass numerical integrator is not suitable for real-time simulation under the previously assumed conditions. Specifically, the calculation of x_{n+1} commencing at $t_{n+1/2}$ requires knowledge of u_{n+1} , which is not available until $T/2$ s later at the end of the frame.

8.5.4 RK-2 (MODIFIED EULER)

This version of RK-2 integration was first introduced in Section 3.6. The equations for updating the discrete-time state from x_n to x_{n+1} are

$$\hat{x}_{n+1/2} = x_n + \frac{T}{2} f(x_n, u_n) \quad (8.323)$$

$$x_{n+1} = x_n + T f(\hat{x}_{n+1/2}, u_{n+1/2}) \quad (8.324)$$

The initial derivative evaluation starts at t_n and requires u_n . The second pass at evaluating the derivative function begins at $t_{n+1/2}$, precisely the time at which $u_{n+1/2}$ becomes available. Hence, both inputs are synchronized in real time with requirements of Equations 8.323 and 8.324. Howe (1995) refers to modified RK-2 integration as RTRK-2 to designate its suitability for real-time simulation.

Next, we look at two versions of RK-3 integration, one that is compatible with real-time simulation and the other that is not compatible.

8.5.5 RK-3 (REAL-TIME INCOMPATIBLE)

The equations for the first RK-3 integrator are as follows:

$$\text{Starting at } t_n: \quad k_1 = f(x_n, u_n), \quad \hat{x}_{n+1/2} = x_n + \frac{T}{2} k_1 \quad (8.325)$$

$$\text{Starting at } t_{n+1/2}: \quad k_2 = f(\hat{x}_{n+1/2}, u_{n+1/2}), \quad \hat{x}_n = x_n + T(-k_1 + 2k_2) \quad (8.326)$$

$$\text{Starting at } t_{n+2/3}: \quad k_3 = f(\hat{x}_n, u_{n+1}), \quad x_{n+1} = x_n + \frac{T}{6} (k_1 + 4k_2 + k_3) \quad (8.327)$$

The unsuitability for real-time implementation of Equations 8.325 through 8.327 stems from the requirement of needing $u_{n+1/2}$ at $t_{n+1/3}$, which is before it is available (see Equation 8.326) and a similar dilemma at time $t_{n+2/3}$ where u_{n+1} is required according to Equation 8.327.

8.5.6 RK-3 (REAL-TIME COMPATIBLE)

A real-time compatible RK-3 integrator is described by

$$\text{Starting at } t_n: \quad k_1 = f(x_n, u_n), \quad \hat{x}_{n+1/3} = x_n + \frac{T}{3} k_1 \quad (8.328)$$

$$\text{Starting at } t_{n+1/3}: \quad k_2 = f(\hat{x}_{n+1/3}, u_{n+1/3}), \quad \hat{x}_{n+2/3} = x_n + \frac{2T}{3} k_2 \quad (8.329)$$

$$\text{Starting at } t_{n+2/3}: \quad k_3 = f(\hat{x}_{n+2/3}, u_{n+2/3}), \quad x_{n+1} = x_n + \frac{T}{4}[k_1 + 3k_3] \quad (8.330)$$

8.5.7 RK-4 (REAL-TIME INCOMPATIBLE)

Fourth-order RK integration is widely used in applications not requiring real-time simulation. It can be shown that all RK-4 integrators require the input u_{n+1} for evaluation of the state derivative on the fourth pass at a time prior to the end of the current frame. Hence, none is compatible with real-time; however, a five-pass RK integrator with fourth-order accuracy suitable for real-time exists.

8.5.8 MULTISTEP INTEGRATION METHODS

The entire family of Adams–Bashforth numerical integrators presented in Section 6.4 is compatible with real time. The lower-order formulas are commonly used in real-time simulation applications. They are preferable to similar order real-time compatible RK integrators because they are single pass in nature and, hence, require less time to execute. For example, AB- m integration requires approximately $1/m$ as much time as any of the RK- m integrators. In **HIL** applications, AB- m simulation can run at frame rates roughly m times greater than any real-time compatible RK- m integrator. Dynamic errors are less for RK- m than AB- m integration with identical step size T ; however, the advantage goes to AB- m integration running at $m \times (1/T)$ frames per second (fps) compared with RK- m integration at $1/T$ fps.

AB-1 integration is identical to explicit Euler. It is used sparingly in real-time simulation for the same reason it is used infrequently in nonreal-time simulation mode, namely, it is a first-order method, and even moderately accurate results require excessively small integration time steps. AB-2 through AB-4 are the most popular choices for real-time simulation. The stability regions of AB integrators higher than fourth order are quite small and become smaller as the order increases (see [Figure 8.21](#)). As a result, numerical stability constraints imposed by high-order AB integrators require the magnitude of λT (λ is the largest magnitude characteristic root of the stable linear or linearized system) be excessively small, thus requiring higher frame rates.

The predictor–corrector multistep methods (referred to by some as Adams–Moulton predictor–correctors) are not real-time compatible. They are two-pass integration algorithms, which combine an explicit Adams–Bashforth integrator to predict the new state followed by an implicit formula based on the predicted state to correct it. Equations 6.204 through 6.209 represent second-through fourth-order methods. The dynamic error properties of predictor–corrector methods (see [Table 8.4](#)) are comparable to the single-pass implicit integrators (which are referred to in this text as Adams–Moulton integrators).

A real-time compatible predictor–corrector formula is possible. An example from Howe (1995) of a second-order method is now presented. The scalar state equation is the same as Equation 8.322. The first step is to generate an estimate of the state $\hat{x}_{n+1/2}$ at the midpoint of the current interval (see Figure 8.72).

This is accomplished by using a form of modified Euler integration, that is, $\hat{x}_{n+1/2}$ is computed based on a step size of $T/2$ according to

$$\hat{x}_{n+1/2} = x_n + \frac{T}{2} \hat{f}_{n+1/4} \quad (8.331)$$

The derivative estimate $\hat{f}_{n+1/4}$ is obtained by linear extrapolation through (t_{n-1}, f_{n-1}) and (t_n, f_n) as shown in Figure 8.72. From the principle of similar triangles,

$$\frac{f_n - f_{n-1}}{t_n - t_{n-1}} = \frac{\hat{f}_{n+1/4} - f_{n-1}}{t_{n+1/4} - t_{n-1}} \quad (8.332)$$

Setting $t_n - t_{n-1} = T$, $t_{n+1/4} - t_{n-1} = 5T/4$ and solving for $\hat{f}_{n+1/4}$ give

$$\hat{f}_{n+1/4} = f_{n-1} + \frac{5}{4}(f_n - f_{n-1}) \quad (8.333)$$

Substituting $\hat{f}_{n+1/4}$ into Equation 8.331 results in the second-order predictor

$$\hat{x}_{n+1/2} = x_n + \frac{T}{8}(5f_n - f_{n-1}) \quad (8.334)$$

The derivative estimate $\hat{f}_{n+1/4}$ is calculated from

$$\hat{f}_{n+1/2} = f(\hat{x}_n + 1/2, un + 1/2) \quad (8.335)$$

Finally, the new state x_{n+1} is obtained from modified Euler integration,

$$x_n + 1 = x_n + Tf_{n+1/2} \quad (8.336)$$

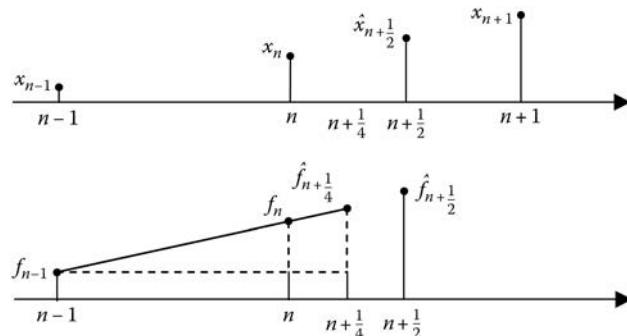


FIGURE 8.72 Diagram for illustrating real-time predictor–corrector method.

Equations 8.334 through 8.336 describe a real-time, predictor–corrector algorithm (which Howe refers to as RTAM-2). It is a two-pass method since it requires two derivative function evaluations per step—the dynamic error coefficient $e_l = 1/24$ making it twice as accurate as the implicit (trapezoidal) and the AB-2/AM-2 predictor–corrector, since both have error coefficients of $e_l = -1/12$ (see Table 8.4), and neither is compatible with real time. It is 10 times more accurate than AB-2 integration, which has an error coefficient $e_l = 5/12$.

For execution times comparable to single-pass formulas, this method would utilize a step size twice as large and generate state updates at half the frequency. After compensating for different step sizes, it still exhibits two and half times the dynamic accuracy of the single-pass AB-2 based on the approximate asymptotic formulas for small step sizes. The estimate $\hat{x}_{n+1/2}$ is available for real-time output; hence, the real-time predictor–corrector can output the state at the same frequency as the single-pass integrators. The integrator requires inputs at the beginning and midpoint of the frame making the sampling frequency twice that of a single-pass integrator. Higher-order real-time compatible predictor–correctors are possible (Howe 1995).

8.5.9 STABILITY OF REAL-TIME PREDICTOR-CORRECTOR METHOD

The stability region for the real-time predictor–corrector given in Equations 8.334 through 8.336 is obtained in the same way as for the explicit Adams–Bashforth, implicit Adams–Moulton, and RK integrators (see Section 8.3). Both the nonreal-time and real-time compatible predictor–correctors introduce extraneous roots in the z -domain and, therefore, are subject to stability limitations on step size. The characteristic polynomials for each integrator are

$$\text{Nonreal-time predictor-corrector: } \Delta(z) = z^2 - \left[1 + \lambda T + \frac{3}{4}(\lambda T)^2 \right] z + \frac{1}{4}(\lambda T)^2 \quad (8.337)$$

$$\text{Real-time predictor-corrector: } \Delta(z) = z^4 - \left[1 + \lambda T + \frac{5}{8}(\lambda T)^2 \right] z^2 + \frac{1}{8}(\lambda T)^2 \quad (8.338)$$

The stability regions are shown in Figure 8.73.

The real-time compatible predictor–corrector has a somewhat larger region, making it preferable from a stability standpoint.

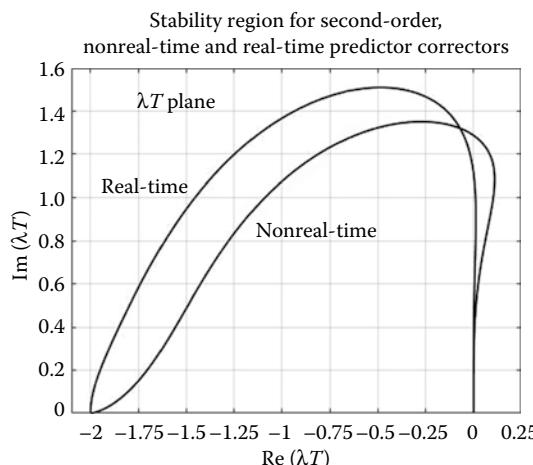


FIGURE 8.73 Stability regions for second-order predictor–corrector methods.

EXAMPLE 8.9

Obtain difference equations for simulating the unit step response of the system

$$\frac{dx}{dt} = \lambda x + u, \quad \lambda = -0.25 \quad (8.339)$$

using the real-time compatible integrators

- a. Modified Euler.
- b. AB-2.
- c. Real-time predictor–corrector.
- d. Choose the step size T , so that $\lambda T = -0.25, -1$ for the modified Euler and real-time predictor–corrector and $\lambda T = -0.125, -0.5$ for the AB-2 integrator. Graph the step responses along with the exact solution and comment on the results.

- a. From Equation 8.323 for modified Euler, the estimated state at the halfway point is

$$\hat{x}_{n+1/2} = x_n + 0.5Tf_n \quad (8.340)$$

$$= x_n + 0.5T(\lambda x_n + u_n) \quad (8.341)$$

$$= (1 + 0.5\lambda T)x_n + 0.5Tu_n \quad (8.342)$$

and the second pass produces the updated state from Equation 8.324 as

$$x_{n+1} = x_n + T\hat{f}_{n+1/2} \quad (8.343)$$

$$= x_n + T(\lambda\hat{x}_{n+1/2} + u_{n+1/2}) \quad (8.344)$$

$$= x_n + [T\lambda \{(1 + 0.5\lambda T)x_n + 0.5Tu_n\} + u_{n+1/2}] \quad (8.345)$$

$$= [1 + \lambda T(1 + 0.5\lambda T)]x_n + T(0.5\lambda Tu_n + u_{n+1/2}) \quad (8.346)$$

- b. The AB-2 difference equation for computing the state is

$$x_{n+1} = x_n + 0.5T(3f_n - f_{n-1}) \quad (8.347)$$

$$= x_n + 0.5T[3(\lambda x_n + u_n) - (\lambda x_{n-1} + u_{n-1})] \quad (8.348)$$

$$= (1 + 1.5\lambda T)x_n - 0.5\lambda Tx_{n-1} + 1.5Tu_n - 0.5Tu_{n-1} \quad (8.349)$$

- c. The real-time predictor–corrector first step is from Equation 8.334,

$$\hat{x}_{n+1/2} = x_n + 0.125T(5f_n - f_{n-1}) \quad (8.350)$$

$$= x_n + 0.125T[5(\lambda x_n + u_n) - (\lambda x_{n-1} + u_{n-1})] \quad (8.351)$$

$$= (1 + 0.625\lambda T)x_n - 0.125\lambda Tx_{n-1} + 0.625Tu_n - 0.125Tu_{n-1} \quad (8.352)$$

The new state is obtained from Equations 8.335 and 8.336,

$$x_{n+1} = x_n + T\hat{f}_{n+1/2} \quad (8.353)$$

$$= x_n + T(\lambda\hat{x}_{n+1/2} + u_{n+1/2}) \quad (8.354)$$

$$= x_n + T[\lambda\{1 + 0.625\lambda T\}x_n - 0.125\lambda Tx_{n-1} + 0.625Tu_n - 0.125Tu_{n-1}] + u_{n+1/2} \quad (8.355)$$

$$= [1 + \lambda T(1 + 0.625\lambda T)]x_n - 0.125(\lambda T)^2 x_{n-1} + \lambda T(0.625Tu_n - 0.125Tu_{n-1}) + Tu_{n+1/2} \quad (8.356)$$

- d. The difference Equations 8.346, 8.349, and 8.356 were solved recursively in the M-file "Ch8_Ex8_9.m." The AB-2 integrator and real-time predictor–corrector were started with a single step of the improved Euler integrator. The results are shown in Figures 8.74 and 8.75 along with the exact solution for the unit step response,

$$x(t) = \frac{1}{\lambda} [e^{\lambda t} - 1], \quad t \geq 0 \quad (8.357)$$

The single-pass AB-2 integrator was running at twice the frame rate of the two-pass modified Euler and real-time predictor–corrector to keep the execution times comparable. In Figure 8.74, the simulated responses using the numerical integrators are in close agreement with the exact solution. In Figure 8.75, the accuracy of the numerical integrators has deteriorated as a result of the increased values of the parameter λT . The M-file "Ch8_Ex8_9.m" includes runs for intermediate values of λT as well.

8.5.10 EXTRAPOLATION OF REAL-TIME INPUTS

A solution to the problem of numerical integrators being incompatible with real-time simulation is to employ extrapolated input data. Consider the improved Euler integrator illustrated in Figure 8.71.

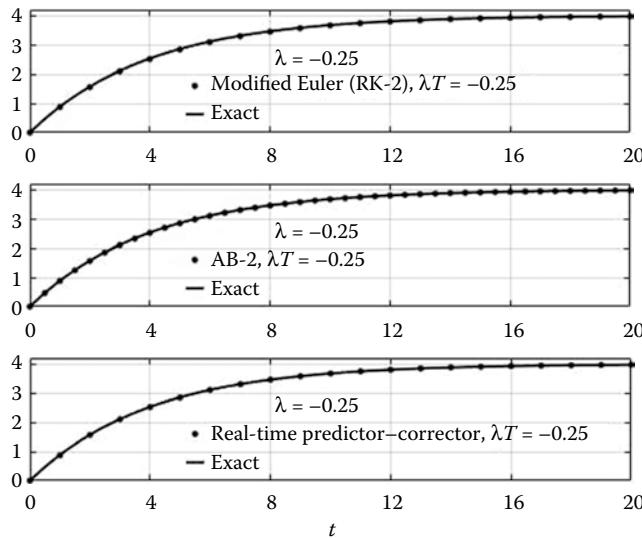


FIGURE 8.74 Unit step response of first-order system using three real-time compatible numerical integrators with $\lambda T = -0.25, -0.125$ and the exact solution.

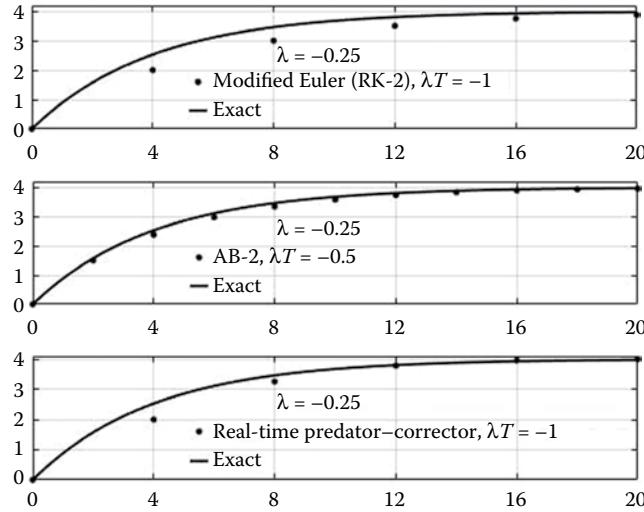


FIGURE 8.75 Unit step response of first-order system using three real-time compatible numerical integrators with $\lambda T = -1, -0.5$ and the exact solution.

The evaluation of $f_n = f(x_n, u_n)$ lasts from t_n to approximately $t_{n+1/2}$. After calculating \hat{x}_{n+1} , the evaluation of, $\hat{f}_{n+1} = f(\hat{x}_{n+1}, u_{n+1})$ is scheduled to begin at $t_{n+1/2}$. The input u_{n+1} is required a half frame before it is available, thus explaining why improved Euler is incompatible with real-time simulation.

A possible remedy is to use first-order (linear) extrapolation based on u_{n-1} and u_n to predict u_{n+1} . An alternative approach is to sample the input at $t_{n+1/2}$ and predict u_{n+1} based on linear extrapolation of u_n , and $u_{n+1/2}$. The predicted values for each approach is denoted \hat{u}_{n+1} in Figure 8.76.

Adopting the first approach leads to

$$\hat{u}_{n+1} = u_n + (u_n - u_{n-1}) \quad (8.358)$$

If we think of Equation 8.358 as the difference equation for a discrete-time system with input u_n and output $y_n = \hat{u}_{n+1}$, $n = 0, 1, 2, \dots$, the first several values of the output are

$$n = 0: y_0 = \hat{u}_1 = u_0 + (u_0 - u_{-1}) = 2u_0 \quad (8.359)$$

$$n = 1: y_1 = \hat{u}_2 = u_1 + (u_1 - u_0) = 2u_1 - u_0 \quad (8.360)$$

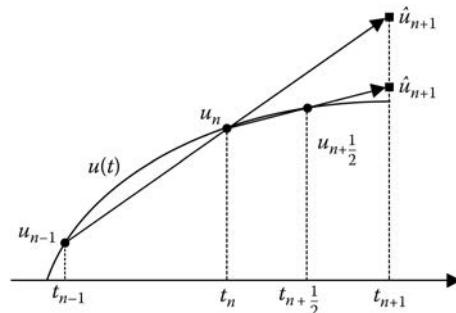


FIGURE 8.76 Use of extrapolation to make improved Euler compatible with real-time.

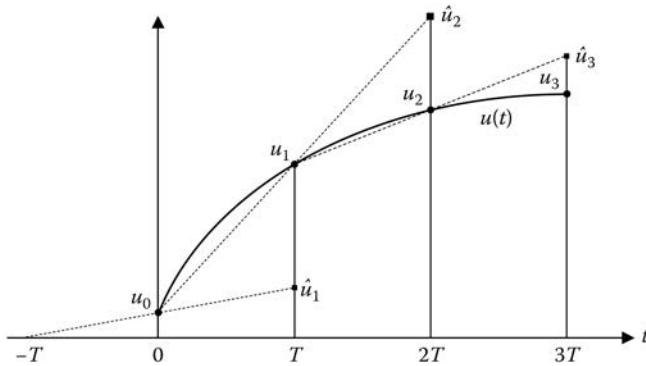


FIGURE 8.77 Linear extrapolation of input $u(t)$.

$$n = 2: y_2 = \hat{u}_3 = u_2 \pm (u_2 - u_1) = 2u_2 - u_1 \quad (8.361)$$

Figure 8.77 shows a continuous-time function $u(t)$, the first four sampled values u_0, u_1, u_2 , and u_3 , and the first three extrapolated values $\hat{u}_1, \hat{u}_2, \hat{u}_3$.

The z -transform of the output sequence $y_n, n = 0, 1, 2, 3, \dots$ is by definition

$$Y(z) = y_0 + y_1 z^{-1} + y_2 z^{-2} + y_3 z^{-3} \dots \quad (8.362)$$

$$= 2u_0 + (2u_1 - u_0)z^{-1} + (2u_2 - u_1)z^{-2} + (2u_3 - u_2)z^{-3} + \dots \quad (8.363)$$

Rearranging the terms in Equation 8.363 gives

$$Y(z) = 2(u_0 + u_1 z^{-1} + u_2 z^{-2} + \dots) - z^{-1}(u_0 + u_1 z^{-1} + u_2 z^{-2} + \dots) \quad (8.364)$$

$$= (2 - z^{-1})U(z) \quad (8.365)$$

The same result follows directly from Equation 8.358 with \hat{u}_{n+1} replaced by y_n . The z -domain transfer function of the linear extrapolator is therefore

$$G(z) = \frac{Y(z)}{U(z)} = 2 - z^{-1} \quad (8.366)$$

Before we discuss the dynamic errors incurred from the use of extrapolation, it is necessary to define the characteristics of an ideal extrapolator. Figure 8.78 illustrates the point for an arbitrary signal $u(t)$ sampled at regular intervals of T units of time.

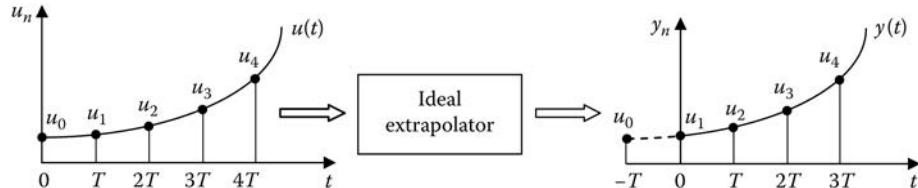


FIGURE 8.78 Illustration of an ideal extrapolator.

At time $t = nT$, if the input to an ideal extrapolator is $u_n = u(nT)$, the output $y_n = u_{n+1} = u[(n+1)T]$. Hence, an ideal extrapolator advances the input $u(t)$ by an amount T to the left along the t -axis. In contrast, a pure delay of the same duration shifts the input $u(t)$ by the same amount to the right along the t -axis.

The Laplace transform of the ideal extrapolator $G_I(s)$ can be obtained by replacing T in the transform for a pure delay of length T with $-T$ leading to

$$G_I(s) = \frac{Y(s)}{U(s)} = e^{-(s-T)} = e^{Ts} \quad (8.367)$$

The frequency response functions of the real and ideal extrapolators are

$$G(z)|_{z=e^{j\omega T}} = G(e^{j\omega T}) = 2 - e^{-j\omega T} \quad (8.368)$$

$$G_I(s)|_{s=j\omega} = G_I(j\omega) = e^{j\omega T} \quad (8.369)$$

The fractional error in $G(e^{j\omega T})$, the extrapolator frequency response function, is

$$e_G = \frac{G(e^{j\omega T}) - G_I(j\omega)}{G_I(j\omega)} = \frac{2 - e^{-j\omega T} - e^{j\omega T}}{e^{j\omega T}} = 2e^{-j\omega T} - e^{-2j\omega T} - 1 \quad (8.370)$$

The fractional error in extrapolator frequency response gain is

$$e_{|G|} = \frac{|G(e^{j\omega T})| - |G_I(j\omega)|}{|G_I(j\omega)|} = |2 - e^{-j\omega T}| - 1 \quad (8.371)$$

Replacing $e^{-j\omega t}$ with $\cos \omega T - j \sin \omega T$, Equation 8.371 reduces to

$$e_{|G|} = (5 - 4 \cos \omega T)^{1/2} - 1 \quad (8.372)$$

An asymptotic formula for $e_{|G|}$ is (see Exercise 8.50)

$$e_{|G|} \approx (\omega T)^2, \quad \omega T \ll 1 \quad (8.373)$$

The phase error in extrapolator frequency response is

$$e_{\angle G} = \text{Arg}\{G(e^{j\omega T})\} - \text{Arg}\{G_I(j\omega)\} \quad (8.374)$$

$$= \text{Arg}\{2 - e^{-j\omega T}\} - \omega T \quad (8.375)$$

An asymptotic formula for $e_{\angle G}$ is (Howe 1995)

$$e_{\angle G} \approx -(\omega T)^3, \quad \omega T \ll 1 \quad (8.376)$$

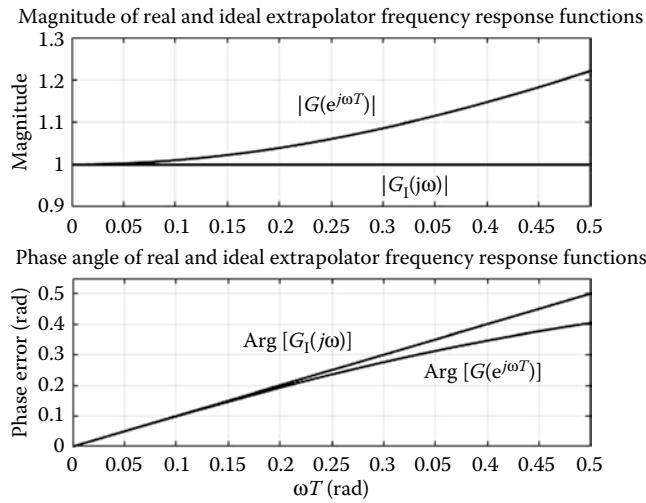


FIGURE 8.79 Magnitude and phase plots for first-order and ideal extrapolator.

Magnitude and phase angle plots of a real and ideal extrapolator are shown in [Figure 8.79](#) for $0 \leq \omega T \leq 0.5$ rad. The graphs are in agreement with Equations 8.373 and 8.376, which imply that the magnitude error is more significant than the phase angle error.

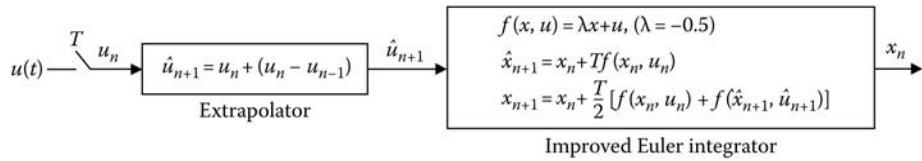
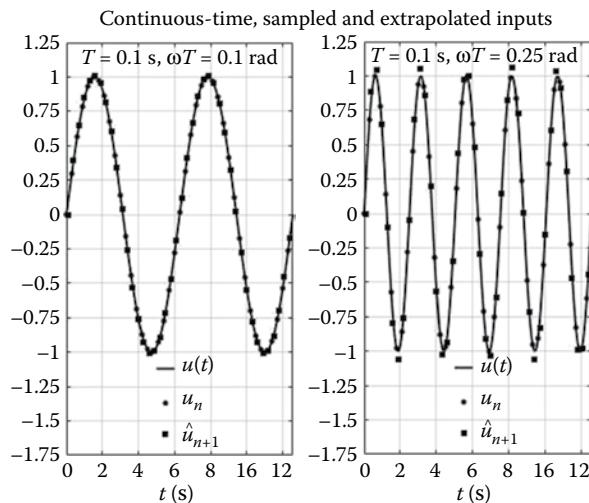
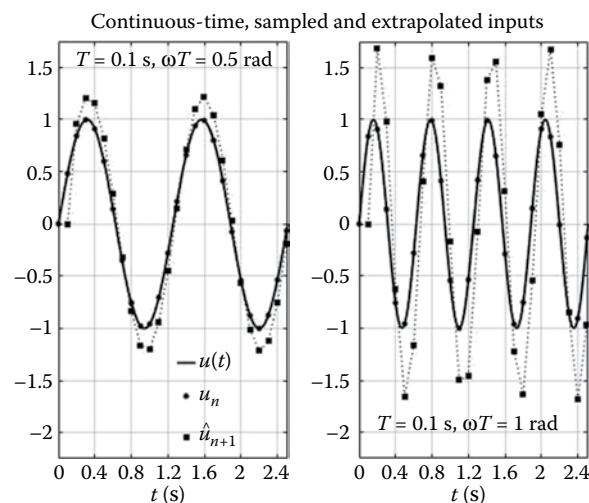
EXAMPLE 8.10

An input signal $u(t) = \sin \omega t$, $t \geq 0$ is sampled every $T = 0.1$ s, and the resulting discrete-time signal u_n , $n = 0, 1, 2, \dots$ is input to an extrapolator governed by Equation 8.358.

- Graph the continuous-time signal $u(t)$, discrete-time signal u_n , and the extrapolator output $\hat{u}_{n+1} = 0, 1, 2, 3, \dots$ for the following cases:
 - $\omega T = 0.1$ rad
 - $\omega T = 0.25$ rad
 - $\omega T = 0.5$ rad
 - $\omega T = 1$ rad
- An improved Euler integrator with step size $T = 0.1$ s is used to simulate the response of the first-order system in Equation 8.339 to the sinusoidal input $u(t) = \sin \omega t$, $t \geq 0$. In order to simulate the real-time response, the input is extrapolated as shown in [Figure 8.80](#) before being numerically integrated. Find the exact and simulated responses for the four cases in part (a) and plot the results.
 - The signals $u(t)$, u_n , and \hat{u}_{n+1} are generated in the script file “Ch8_Ex8_10.m” and the results are shown in [Figures 8.81](#) and [8.82](#). For purposes of clarity, not all discrete points are plotted. The extrapolator gain error is first noticeable at $\omega T = 0.25$ rad, becoming progressively worse at $\omega T = 0.5$ rad and $\omega T = 1$ rad, respectively.
 - The analytical solution for the response is obtained by Laplace transformation of the differential equation $\dot{x} = \lambda x + u$ with sinusoidal input $u = \sin \omega t$. The Laplace transform of $x(t)$ is

$$X(s) = \frac{\omega}{(s - \lambda)(s^2 + \omega^2)} \quad (8.377)$$

which is easily inverted by partial fractions to give

**FIGURE 8.80** Real-time simulation of first-order system dynamic response.**FIGURE 8.81** Continuous-time, sampled and extrapolated inputs ($\omega T = 0.1, 0.25$ rad).**FIGURE 8.82** Continuous-time, sampled and extrapolated inputs ($\omega T = 0.5, 1$ rad).

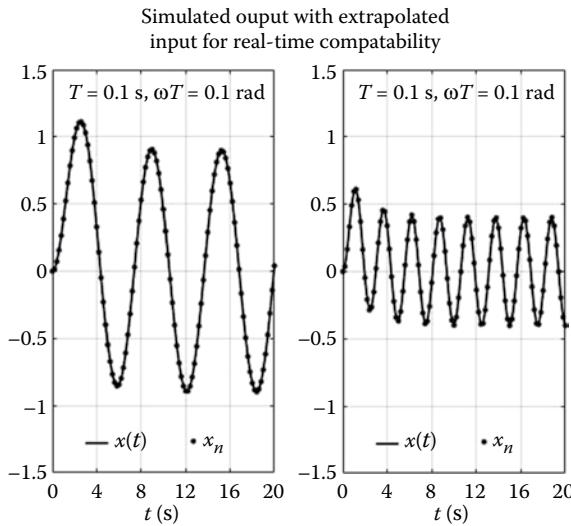


FIGURE 8.83 Exact and simulated (improved Euler) responses ($\omega T = 0.1, 0.25 \text{ rad}$).

$$x(t) = \frac{\omega}{\lambda^2 + \omega^2} \left(e^{\lambda t} - \cos \omega t - \frac{\lambda}{\omega} \sin \omega t \right) \quad (8.378)$$

$$= \frac{\omega}{(\lambda^2 + \omega^2)} e^{\lambda t} - \frac{1}{(\lambda^2 + \omega^2)^{1/2}} \sin(\omega t + \varphi), \quad \varphi = \pi + \tan^{-1}\left(\frac{\omega}{\lambda}\right) \quad (8.379)$$

The exact response $x(t)$ and the simulated responses x_n for the cases when $\omega T = 0.1 \text{ rad}$ and $\omega T = 0.25 \text{ rad}$ are plotted in Figure 8.83. Results for the remaining two cases, $\omega T = 0.5 \text{ rad}$ and $\omega T = 1 \text{ rad}$, are shown in Figure 8.84. Error in the simulated response due to extrapolator gain error is significant at input frequencies $\omega T = 0.5 \text{ rad}$ and

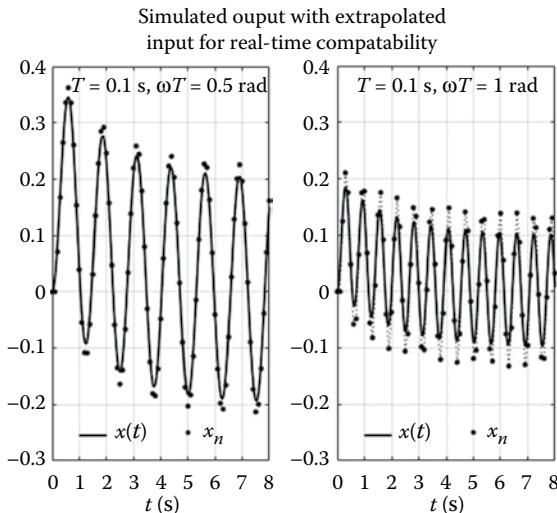


FIGURE 8.84 Exact and simulated (improved Euler) responses ($\omega T = 0.5, 1 \text{ rad}$).

$\omega T = 1$ rad where the asymptotic approximations in Equations 8.373 and 8.376 are no longer valid.

8.5.11 ALTERNATE APPROACH TO REAL-TIME COMPATIBILITY: INPUT DELAY

When numerical integrators are not compatible with real-time simulation, it is because the input(s) are required at points in time prior to their occurrence. One solution to this dilemma is to use input values previously sampled in place of the input data required by the formula in the numerical integration algorithm. Refer to Figure 8.85, which shows an input $u(t)$ and delayed versions $u(t - T/2)$, $u(t - T)$.

Let us assume once again that improved Euler, a second-order, two-pass RK integrator incompatible with real-time simulation, is to be used. Starting at time t_n , the first stage is an Euler prediction of the state at t_{n+1} . However, instead of using the current input u_n , suppose the input from one-half a time step in the past is used, namely, $u_{n-1/2}$. That is, \hat{x}_{n+1} is computed from

$$\hat{x}_{n+1} = x_n + Tf(x_n, u_{n-1/2}) \quad (8.380)$$

Starting at time $t_{n+1/2}$, the second pass to compute the new state is

$$x_{n+1} = x_n + \frac{T}{2}[f(x_n, u_{n+1/2}) + f(\hat{x}_{n+1}, u_{n+1/2})] \quad (8.381)$$

Assuming Equation 8.381 requires approximately $T/2$ units of time to execute, the updated state x_{n+1} is available at time t_{n+1} . Hence, by using $u_{n-1/2}$ in place of u_n and $u_{n+1/2}$ instead of u_{n+1} , the improved Euler integrator is running in real time. Equations 8.380 and 8.381 applied to the first-order system $dx/dt = \lambda x + u$ lead to the difference equation

$$x_{n+1} = \left[1 + \lambda T + \frac{(\lambda T)^2}{2}\right]x_n + \frac{T}{2}(1 + \lambda T)u_{n-1/2} + \frac{T}{2}u_{n+1/2} \quad (8.382)$$

Equation 8.382 is similar to the difference equation for simulation of the first-order system using classical improved Euler integration except for the presence of the delayed input, that is, u_n is replaced by $u_{n-1/2}$ and u_{n+1} is replaced by $u_{n+1/2}$.

There is of course a penalty incurred as a result of using “old” values from the delayed input $u(t)$. To illustrate, consider the case where sampled values are obtained from the input delayed a full time step T as shown in Figure 8.86.

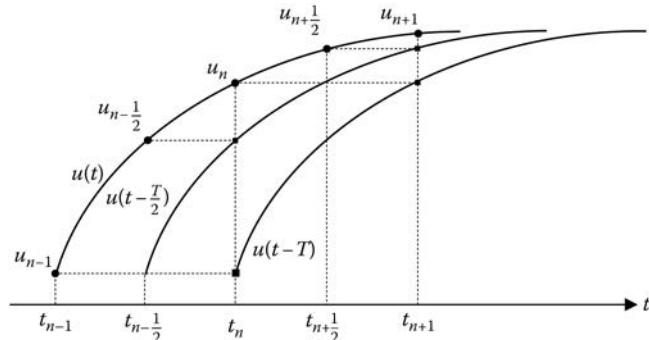


FIGURE 8.85 Use of delayed input to make numerical integrator real-time compatible.

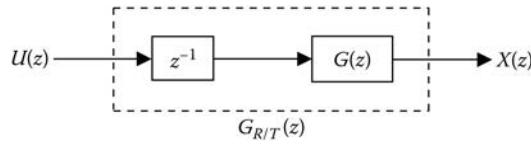


FIGURE 8.86 z -Domain transfer function for real-time implementation.

Simulation of the system $\frac{dx}{dt} = \lambda x + u$ with real-time, improved Euler integration leads to a discrete-time system with z -domain transfer function

$$G_{R/T}(z) = \frac{X(z)}{U(z)} = z^{-1}G(z) = \frac{b_1 z + b_0}{z(z + a_0)} \quad (8.383)$$

where

$$a_0 = 1 + \lambda T + \frac{(\lambda T)^2}{2}, b_0 = \frac{T}{2}(1 + \lambda T), b_1 = \frac{T}{2} \quad (8.384)$$

The continuous-time system transfer function is

$$G(s) = \frac{1}{s - \lambda} \quad (8.385)$$

The dynamic errors in the discrete-time frequency response functions are

$$e_G = \frac{G(z)|_{z \leftarrow e^{j\omega T}} - G(s)|_{s \leftarrow j\omega}}{G(s)|_{s \leftarrow j\omega}} \quad (8.386)$$

$$= \frac{(b_1 e^{j\omega T} + b_0)/(e^{j\omega T} + a_0) - 1}{1/(j\omega - \lambda)} \quad (8.387)$$

$$= \frac{(j\omega - \lambda)(b_1 e^{j\omega T} + b_0)}{e^{j\omega T} + a_0} - 1 \quad (8.388)$$

$$e_{G_{R/T}} = \frac{G_{R/T}(z)|_{z \leftarrow e^{j\omega T}} - G(s)|_{s \leftarrow j\omega}}{G(s)|_{s \leftarrow j\omega}} \quad (8.389)$$

$$= \frac{(b_1 e^{j\omega T} + b_0)/(e^{j\omega T}(e^{j\omega T} - a_0)) - 1}{1/(j\omega - \lambda)} \quad (8.390)$$

$$= \frac{(j\omega - \lambda)(b_1 e^{j\omega T} + b_0)}{e^{j\omega T}(e^{j\omega T} - a_0)} - 1 \quad (8.391)$$

The fraction gain errors are

$$e_{|G|} = \frac{|G(e^{j\omega T})| - |G(j\omega)|}{|G(j\omega)|} \quad (8.392)$$

$$= \frac{|(b_1 e^{j\omega T} + b_0)/(e^{j\omega T} - a_0)|}{|1/(j\omega - \lambda)|} - 1 \quad (8.393)$$

$$e_{|G_{R/T}|} = \frac{|G_{R/T}(e^{j\omega T})| - |G(j\omega)|}{|G(j\omega)|} \quad (8.394)$$

$$= \frac{|(b_1 e^{j\omega T} + b_0)/[(e^{j\omega T}(e^{j\omega T} - a_0))|}{|1/(j\omega - \lambda)|} - 1 \quad (8.395)$$

$$= \frac{|(b_1 e^{j\omega T} + b_0)/[(e^{j\omega T} - a_0)]|}{|1/(j\omega - \lambda)|} - 1 \quad (8.396)$$

$$= e_{|G|} \quad (8.397)$$

The phase error are

$$e_{\angle G} = \angle G(e^{j\omega T}) - \angle G(j\omega) \quad (8.398)$$

$$= \text{Arg}\left(\frac{b_1 e^{j\omega T} + b_0}{e^{j\omega T} - a_0}\right) - \text{Arg}\left(\frac{1}{j\omega - \lambda}\right) \quad (8.399)$$

$$e_{\angle G_{R/T}} = L G_{R/T}(e^{j\omega T}) - \angle G(j\omega) \quad (8.400)$$

$$= \text{Arg}\left(\frac{b_1 e^{j\omega T} + b_0}{e^{j\omega T}(e^{j\omega T} - a_0)}\right) - \text{Arg}\left(\frac{1}{j\omega - \lambda}\right) \quad (8.401)$$

$$= \text{Arg}\left(\frac{b_1 e^{j\omega T} + b_0}{(e^{j\omega T} - a_0)}\right) - \omega T - \text{Arg}\left(\frac{1}{j\omega - \lambda}\right) \quad (8.402)$$

$$= e_{\angle G} - \omega T \quad (8.403)$$

The fractional gain and phase errors for the classical and real-time, improved Euler integrators are graphed in [Figure 8.87](#) for the case when $\lambda = -0.5$. As expected from Equation 8.397, the fractional gain errors are equal and from Equation 8.403, the real-time, improved Euler integrator introduces an additional phase lag of ωT rad. Note that the fractional gain error varies from zero to approximately -2% over the interval $0 \leq \omega T \leq 0.5$ rad.

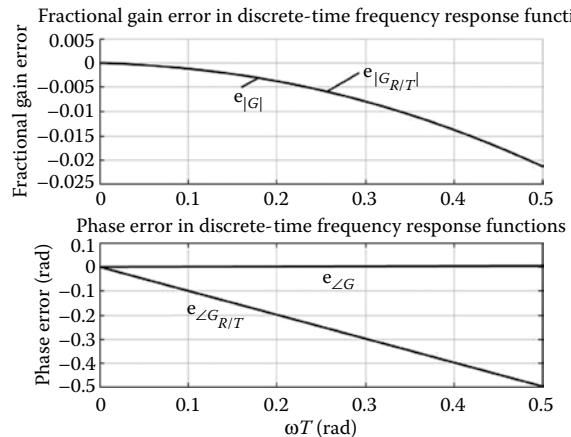


FIGURE 8.87 Dynamic errors from simulation of $dx/dt = -0.5x + u$ with classical and real-time, improved Euler integration ($T = 0.1$ s).

Also, note that $e_{\angle G} \approx 0$ for $0 \leq \omega T \leq 0.5$ rad. Hence, the classical improved Euler integrator contributes essentially zero phase shift with respect to the continuous-time frequency response.

The phase angles (in deg) of the two discrete-time and the continuous-time frequency response functions are shown in Figure 8.88. As expected from Equation 8.403, the separation between the top two plots $\angle G(e^{j\omega T})$ and $\angle G(j\omega)$ and the bottom plot $\angle G_{R/T}(e^{j\omega T})$ is ωT rad. For example, at $\omega T = 0.3$ rad, $\angle G(e^{j\omega T}) = \angle G(j\omega) = -1.4031$ rad (-80.3914 deg) and $\angle G_{R/T}(e^{j\omega T}) = -1.7031$ rad (-97.5801 deg).

EXAMPLE 8.11

An object with thermal capacitance C and thermal resistance R shown in Figure 8.89 is exposed to a surrounding temperature that varies according to $T_0(t) = \bar{T}_0 + \Delta T_0 \sin 2\pi f_0 t, t \geq 0$. The

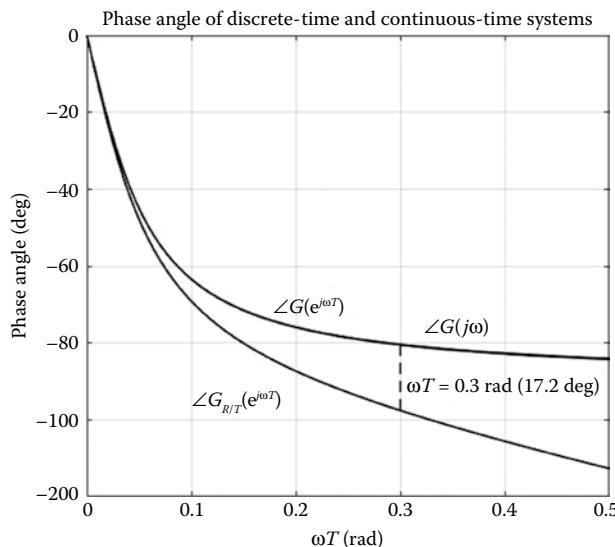
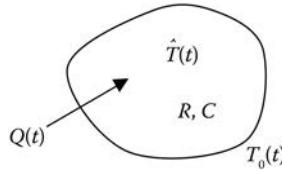


FIGURE 8.88 Phase of discrete-time and continuous frequency response functions.

**FIGURE 8.89** Thermal system for Example 8.11.

mathematical model governing $\hat{T}(t)$, the temperature of the object, consists of Equations 8.404 and 8.405.

Baseline system parameter values are

$$C = 200 \text{ Btu}/^{\circ}\text{F}, R = 0.005 ^{\circ}\text{F/Btu/h},$$

$$\bar{T}_0 = 50^{\circ}\text{F}, \Delta T_0 = 20^{\circ}\text{F}, f_0 = 1 \text{ cycle every } 24 \text{ h}, T(0) = 50^{\circ}\text{F}$$

Find the difference equations for simulating the temperature response using

- Improved Euler integration
- Real-time, improved Euler integration using a one-step delayed version of the input
- Find the analytical solution for $\hat{T}(t)$.
- Simulate the temperature response over two cycles in $T_0(t)$ by recursive solution of the difference equations in parts (a) and (b). Choose the time step T , so that $T/RC = 0.25$. Plot the analytical and numerical solutions on the same graph.
- Repeat part (d) for $f_0 = 1$ cycle every 3 h.

$$C \frac{d\hat{T}}{dt} = Q(t) \quad (8.404)$$

$$Q(t) = \frac{1}{R} [T_0(t) - \hat{T}(t)] \quad (8.405)$$

- Combining Equations 8.404 and 8.405 leads to the differential equation of the system.

$$\tau \frac{d\hat{T}}{dt} + \hat{T}(t) = T_0(t), \quad \tau = RC \quad (8.406)$$

The state derivative function is

$$f(\hat{T}, T_0) = \frac{d\hat{T}}{dt} = \frac{1}{\tau} (T_0 - \hat{T}) \quad (8.407)$$

and the difference equation for implementing standard improved Euler integration is

$$\hat{T}_{n+1} = \left[1 - \frac{T}{\tau} + \frac{1}{2} \left(\frac{T}{\tau} \right)^2 \right] \hat{T}_n + \frac{T}{2\tau} \left(1 - \frac{T}{\tau} \right) T_{0,n} + \left(\frac{T}{2\tau} \right) T_{0,n+1}, \quad n = 0, 1, 2, \dots \quad (8.408)$$

where

$$T_{0,n} = \bar{T} + \Delta T_0 \sin \omega n T, \quad n = 0, 1, 2, \dots (\omega = 2\pi f_0) \quad (8.409)$$

- b. Delaying the input $T_0(t)$ by T h before sampling leads to the difference equation

$$\hat{T}_{n+1} = \left[1 - \frac{T}{\tau} + \left(\frac{T}{\tau} \right)^2 \right] \hat{T}_n + \frac{T}{2\tau} \left(1 - \frac{T}{\tau} \right) T_{0,n-1} + \left(\frac{T}{2\tau} \right) T_{0,n}, \quad n = 0, 1, 2, \dots \quad (8.410)$$

- c. The analytical solution for $\hat{T}(t)$ is obtained by Laplace transforming Equation 8.406 followed by inverse Laplace transformation of the expression for $\hat{T}(s)$. The steps are left for an exercise. The result is

$$\bar{T}_0 + \left[\hat{T}(0) - \bar{T}_0 \frac{\tau \omega \Delta T_0}{1 + (\tau \omega)^2} \right] e^{-t/\tau} + \frac{\Delta T_0}{1 + (\tau \omega)^2} [\sin \omega t - (\tau \omega) \cos \omega t] \quad (8.411)$$

- d. The simulated responses are determined by recursive solution of the appropriate difference equation in "Ch8_Ex8_11.m." The step size is determined from

$$T = 0.25RC = 0.25(0.005^{\circ}\text{F/Btu/h})(200\text{Btu}/^{\circ}\text{F}) = 0.25\text{h}$$

The continuous-time input $T_0(t)$ is shown in the top half of Figure 8.90. The discrete-time input $T_{0,n}$, $n = 0, 1, 2, 3, \dots$ are the sampled values at 0.25 h intervals; however, only the sampled values at the end of each hour are shown in Figure 8.90. The lower half of Figure 8.90 shows the continuous-time output $\hat{T}(t)$ and the discrete-time outputs at the end of each hour, that is, every fourth value.

The continuous-time response $\hat{T}(t)$ and the simulated response \hat{T}_n generated by improved Euler integration are indistinguishable from each other at the end of the integration steps. The discrete-time response $\hat{T}_{R/T,n}$ is simply \hat{T}_n delayed by $T = 0.25$ h. There is close agreement between the simulated and analytical responses because the dynamic errors are very small when $\omega T = 0.065$ (see Figure 8.87).

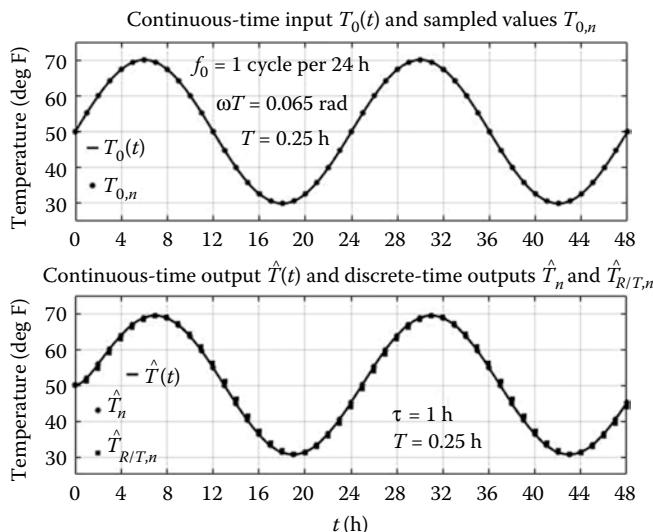


FIGURE 8.90 Continuous- and discrete-time inputs and outputs ($f_0 = 1$ cycle/24 h).

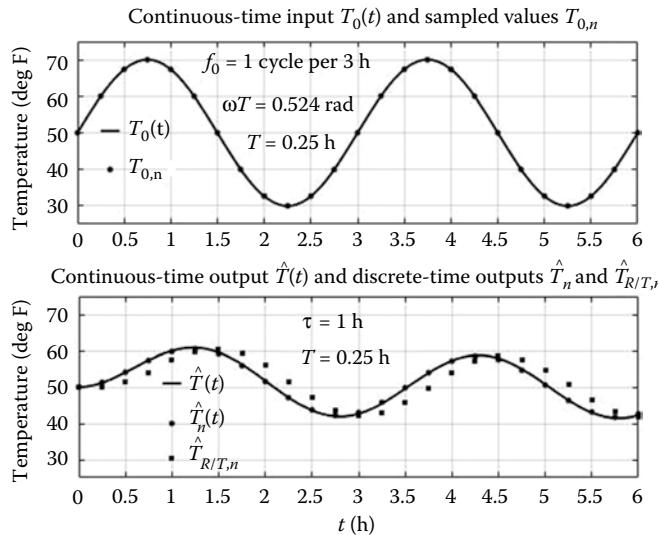


FIGURE 8.91 Continuous- and discrete-time inputs and outputs ($f_0 = 1$ cycle/3 h).

- e. The period of input temperature fluctuations is reduced from 24 to 3 h. The new radian frequency is $\omega = 2\pi f_0 = 2\pi (1/3) = 2.094$ rad/h and $\omega T = 0.524$ rad. A slight difference between the simulated response \hat{T}_n , and the continuous-time response is now evident as shown in Figure 8.91. According to Figure 8.87, the two are in phase and the fractional gain error is approximately -0.02 (-2%).

The real-time, improved Euler temperature response $\hat{T}_{R/T,n}$ is once again a delayed version of \hat{T}_n , the delay being $T = 0.25$ h. There is a significant difference between the analytical solution and the real-time, improved Euler response.

EXERCISES

- 8.47 Rework Example 8.9 using two half steps of RK-2 to generate the starting value for the real-time predictor–corrector.
- 8.48 Use the real-time predictor–corrector to simulate the response of the first-order system in Example 8.9 for $\lambda T = 0.1$ and a sinusoidal input $u(t) = \sin \omega t$, $t \geq 0$. Write a MATLAB script file that accepts values for the radian frequency in the range $0.1 \omega_{BW} \leq \omega < 10 \omega_{BW}$, where ω_{BW} is the system bandwidth and plots the simulated response and exact solution.
- 8.49 Suppose a zero-order extrapolator $y_n = \hat{u}_{n+1} - u_n$, $n = 0, 1, 2, \dots$ is used instead of the first-order extrapolator in Equation 8.358.
- Find the z-transform $G(z) = Y(z)/U(z)$ of this extrapolator.
 - Find an expression for the fractional error in the frequency response function e_G .
 - Find expressions for the fraction error in gain $e_{|G|}$ and the error in phase $e_{\angle G}$.
 - Find asymptotic formulas for the errors in part (c).
 - Plot the magnitude and phase of the zero order and ideal extrapolator.
- 8.50 Derive the asymptotic expression in Equation 8.373 for the fractional error in extrapolator frequency response gain.
- 8.51 Estimate the fractional gain and phase errors from the graph in Example 8.10 (Figure 8.79) for the case when $\omega T = 0.5$ rad. Compare the results with the exact values given in Equations 8.372 and 8.375.
- Hint:* Run “Ch8_Ex8_10.m” and enlarge the plots to facilitate the measurements needed to estimate the respective errors.

8.52 Repeat Example 8.10 part (b) using the second-order system

$$\frac{d^2x}{dt^2} + 2\xi\omega_n \frac{dx}{dt} + \omega_n^2 x = K\omega_n^2 u \quad (K = 2, \omega_n = 10 \text{ rad/s})$$

in place of the first-order system. Plot the exact and simulated responses for $\zeta = 0.1$, $\zeta = 0.707$, and $\zeta = 2$ when $\omega T = 0.1$ rad, $\omega T = 0.25$ rad, $\omega T = 0.5$ rad, and $\omega T = 1$ rad. Note that there are a total of 12 distinct combinations of ζ and ωT .

8.53 Derive the analytical expression for $\hat{T}(t)$ in Equation 8.411.

8.54 Run the M-file “Ch8_Ex8_11.m.”

- Zoom in the bottom graph in Figure 8.91 in order to accurately measure the peak amplitudes (with respect to $T_0 = 50^\circ \text{ F}$) after the transient response has died out. Calculate the fractional error in $|G(e^{j\omega})|$ and compare to the value estimated from Figure 8.87.
- Measure the time phase shift in $\hat{T}(t)$ and \hat{T}_n with respect to the input $T_0(t)$, and convert the value to degrees. Compare your answer with the phase angle estimated from Figure 8.87.

8.55 Rework Example 8.11 and include the real-time, modified Euler integrator given in Equations 8.323 and 8.324.

8.56 The classic RK-4 integrator introduced in Section 6.2 is incompatible with real-time simulation. Choose a step size of $T = 0.01$ s for the RK-4 integrator given by

$$\begin{aligned} k_1 &= f(x_n, u_n), \quad x_{n+1/2} = x_n + 0.5Tk_1 \\ k_2 &= f(x_{n+1/2}, \hat{u}_{n+1/2}), \quad \hat{x}_{n+1/2} = x_n + 0.5Tk_2 \\ k_3 &= f(\hat{x}_{n+1/2}, u_{n+1/2}), \quad \hat{x}_{n+1/2} = x_n + 0.5Tk_3 \\ k_4 &= f(\hat{x}_{n+1}, \hat{u}_{n+1}) \end{aligned}$$

$$x_{n+1} = x_n + \frac{T}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

to simulate the response of the system $dx/dt = x + u$ when the input is given by $u = u(t) = \sin 25t$, $t \geq 0$. The initial condition $x(0) = 0$. Compute $\hat{u}_{n+1/2}$ and \hat{u}_{n+1} based on linear extrapolation through the points (t_{n-1}, u_{n-1}) and (t_n, u_n) . Plot the exact solution and the simulated response on the same graph.

8.6 ADDITIONAL METHODS OF APPROXIMATING CONTINUOUS-TIME SYSTEM MODELS

Several additional methods for simulating the dynamics of continuous-time systems are presented in this section. Explanations of each are followed by the application of the methods to a linear continuous-time system to produce the z -domain transfer function, difference equations, and frequency response functions of the resulting discrete-time systems.

8.6.1 SAMPLING AND SIGNAL RECONSTRUCTION

A special case of this method was introduced briefly in Exercise 4.74. A discrete-time system to approximate an LTI continuous-time system can be synthesized by sampling the continuous-time input and then reconstituting the input using a reconstruction process. The reconstructed signal is applied to the LTI continuous-time system. Finally, the output is sampled to produce a discrete-time signal. The process is illustrated in Figure 8.92.

The sampled values u_k , $k = 0, 1, 2, \dots$ can be used to reconstruct a piecewise continuous approximation to $u(t)$ in different ways. The simplest approach is to use a zero-order hold (ZOH) circuit,

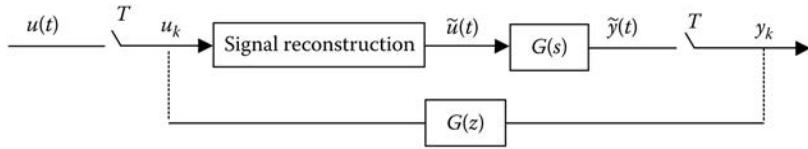


FIGURE 8.92 Sampling and signal reconstruction to approximate a linear time-invariant continuous-time system.

which generates a zero-order polynomial fit through the sampled values to produce the piecewise constant staircase function $\tilde{u}(t)$ shown in Figure 8.93. A single value of u_k , $k = 0, 1, 2, \dots$ in each interval is all that is required to reconstruct the continuous-time signal approximation for that interval.

The piecewise constant function $\tilde{u}(t)$ can be decomposed into a series of rectangular pulses as shown in Figure 8.94.

Expressing $\tilde{u}(t)$ in terms of the unit step function $u(t)$,

$$\tilde{u}(t) = u_0[1 - \hat{u}(t - T)] + u_1[\hat{u}(t - T) - \hat{u}(t - 2T)] + u_2[\hat{u}(t - 2T) - \hat{u}(t - 3T)] + \dots \quad (8.412)$$

Laplace transforming Equation 8.412 gives

$$\tilde{U}(s) = u_0 \left[\frac{1}{s} - \frac{e^{-Ts}}{s} \right] + u_1 \left[\frac{e^{-Ts}}{s} - \frac{e^{-2Ts}}{s} \right] + u_2 \left[\frac{e^{-2Ts}}{s} - \frac{e^{-3Ts}}{s} \right] + \dots \quad (8.413)$$

The output of the continuous-time system with transfer function $G(s)$ is

$$\tilde{y}(t) = \mathcal{L}^{-1}\{\tilde{Y}(s)\} = \mathcal{L}^{-1}\{G(s)\tilde{U}(s)\} \quad (8.414)$$

$$= \mathcal{L}^{-1} \left\{ G(s)u_0[(1 - e^{-Ts}) + u_1 e^{-Ts}(1 - e^{-Ts}) + u_2 e^{-2Ts}(1 - e^{-Ts}) + \dots] \frac{1}{s} \right\} \quad (8.415)$$

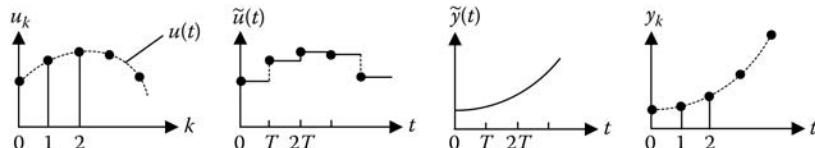


FIGURE 8.93 Representative signals in Figure 8.92 using a ZOH reconstruction device.

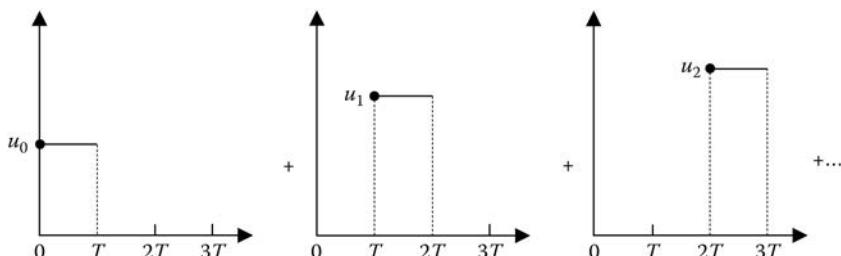


FIGURE 8.94 ZOH output $\tilde{u}(t)$ shown as a sum of rectangular pulses.

The discrete-time output y_k consists of the sampled values $\tilde{y}(t)|_{t=kT}, k = 0, 1, 2, \dots$. $Y(z)$ is obtained by z -transforming Equation 8.415 after setting $z = e^{Ts}$, resulting in

$$Y(z) = z \left\{ [u_0(1-z^{-1}) + u_1z^{-1}(1-z^{-1}) + u_2z^{-2}(1-z^{-1}) + \dots] \mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} \right\} \quad (8.416)$$

$$= z \left\{ [u_0 + u_1z^{-1} + u_2z^{-2} + \dots] (1-z^{-1}) \mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} \right\} \quad (8.417)$$

The inside bracketed expression is recognized as $U(z) = z\{u_k\}$. Hence,

$$Y(z) = U(z)(1-z^{-1})z \left\{ \mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} \right\} \quad (8.418)$$

where $z\{\mathcal{L}^{-1}\{G(s)/s\}\}$ represents the z -transform of the discrete-time signal obtained from uniform sampling of the continuous-time signal $\mathcal{L}^{-1}\{G(s)/s\}$. The z -domain transfer function resulting from the sampling and ZOH reconstruction method illustrated in Figure 8.92 is given by

$$G(z) = \frac{Y(z)}{U(z)} = (1-z^{-1})z \left\{ \mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} \right\} \quad (8.419)$$

The errors resulting from the use of Equation 8.419 are related to the signal reconstruction process. As you might expect, properties of the input $u(t)$, sampling interval T , and the method of reconstructing the input from the sampled values $u_k, k = 0, 1, 2, \dots$ play a central role in the process.

We now illustrate the application of Equation 8.419 in finding a discrete-time system approximation of a second-order continuous-time system.

EXAMPLE 8.12

Consider an underdamped second-order system with damping ratio $\zeta = 1/\sqrt{10}$, natural frequency $\omega_n = \sqrt{10}$, rad/s, and steady-state gain of unity.

- Find the z -domain transfer function and difference equation of the discrete-time system approximation. Leave your answer in terms of the sampling period T .
- Input to the continuous-time system is $u(t) = 5(1 - e^{-2t}), t \geq 0$. Find the continuous-time system response $y(t), t \geq 0$.
- Plot the continuous-time system response $y(t)$ and the discrete-time approximation $y_k, k = 0, 1, 2, \dots$ for $T = 0.05, 0.1, 0.25, 0.5$ s.

- The transfer function of the continuous-time system is

$$G(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{10}{s^2 + 2s + 10} \quad (8.420)$$

$$\frac{G(s)}{s} = \frac{10}{s(s^2 + 2s + 10)} = \frac{1}{s} - \frac{s+1}{s^2 + 2s + 10} \quad (8.421)$$

$$\mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} = 1 - e^{-t} \left(\cos 3t + \frac{1}{3} \sin 3t \right) \quad (8.422)$$

From Table 4.4,

$$z[1] = \frac{z}{z-1} \quad (8.423)$$

$$z\{e^{-kt} \cos 3kT\} = \frac{z^2 - (e^{-T} \cos 3T)z}{z^2 - (2e^{-T} \cos 3T)z + e^{-2T}} \quad (8.424)$$

$$z\{e^{-kt} \sin 3kT\} = \frac{(e^{-T} \sin 3T)z}{z^2 - (2e^{-T} \cos 3T)z + e^{-2T}} \quad (8.425)$$

Using Equations 8.423 through 8.425 in Equation 8.419 for $G(z)$ results in (after simplification)

$$G(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} \quad (8.426)$$

$$b_1 = 1 - e^{-T} \left(\cos 3T + \frac{1}{3} \sin 3T \right), \quad b_2 = e^{-2T} - e^{-T} \left(\cos 3T + \frac{1}{3} \sin 3T \right) \quad (8.427)$$

$$a_1 = -2e^{-T} \cos 3T, \quad a_2 = e^{-2T} \quad (8.428)$$

Equation 8.426 leads to the difference equation of the discrete-time system

$$y_k + a_1 y_{k-1} + a_2 y_{k-2} = b_1 u_k + b_2 u_{k-1} \quad (8.429)$$

b. The continuous-time system response to the input $u(t)$ is obtained from

$$y(t) = \mathcal{L}^{-1}\{G(s)U(s)\} = \mathcal{L}^{-1}\left\{\frac{10}{s^2 + 2s + 10} \cdot 5\left(\frac{1}{s} - \frac{1}{s+2}\right)\right\} \quad (8.430)$$

Partial fraction expansion of the terms in brackets followed by inverse Laplace transformation leads to

$$y(t) = 5 - 5e^{-2t} - \frac{10}{3}e^{-t} \sin 3t, \quad t \geq 0 \quad (8.431)$$

c. The MATLAB M-file “Ch8_Ex8_12.m” includes statements to solve Equation 8.429 in recursive fashion. [Figure 8.95](#) shows the continuous-time system response and the discrete-time response when $T = 0.05, 0.1, 0.25, 0.5$ s.

For signals that are not band limited such as the input $u(t) = 5(1 - e^{-2t})$, a good rule of thumb is to sample 10 times faster than the shortest time constant ($\tau = 0.5$ s in this case). The top left graph in [Figure 8.95](#) corresponds to $T = \tau/10 = 0.05$ s, and the agreement between the continuous-time and discrete-time responses is excellent.

The outputs of the ZOH for the two extremes ($T = 0.05$ and 0.5 s) are shown in [Figure 8.96](#), illustrating the importance of the sampling process.

The ZOH has characteristics similar to a low-pass filter. To see this, suppose the first sampler in [Figure 8.92](#) produces a train of impulses of strength $u(kT)$ at the sampling instants kT , $k = 0, 1, 2, \dots$ instead of the discrete-time signal $u_k = u(kT)$, $k = 0, 1, 2, \dots$. Knowing the output of the

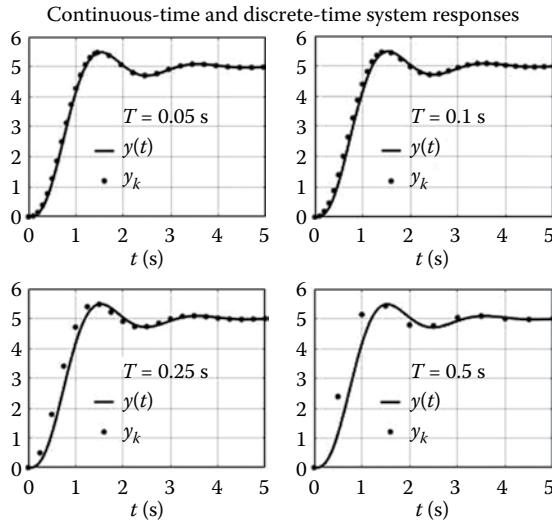


FIGURE 8.95 Illustration of “sample and ZOH reconstruction” method.

ZOH is u_k , $kT \leq t \leq (k+1)T$ implies that the ZOH is effectively integrating the k th impulse for $kT \leq t < (k+1)T$. The situation is portrayed in Figure 8.97. The transfer function of the ZOH is therefore

$$G_{\text{ZOH}}(s) = \frac{1 - e^{-Ts}}{s} \quad (8.432)$$

Keep in mind that the impulse sampler is a mathematical fiction that allows the zero-order hold to be modeled by the continuous-time transfer function in Equation 8.432.

The frequency response function is obtained by replacing s with $j\omega$ in Equation 8.432.

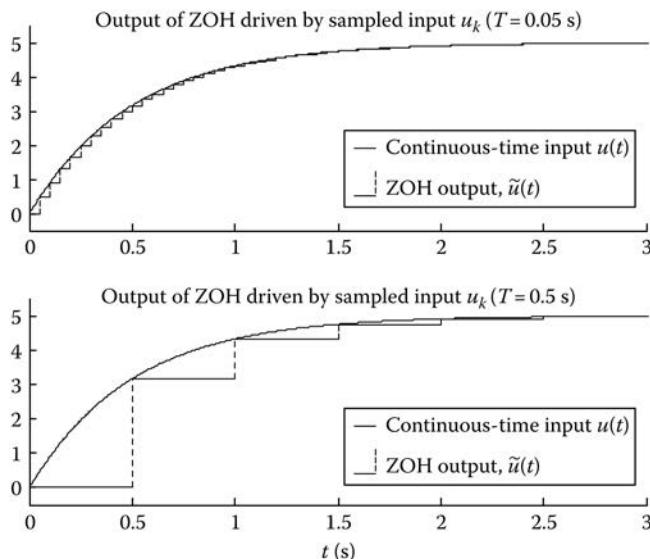


FIGURE 8.96 Effect of sampling rate on ZOH reconstruction of input $u(t)$.

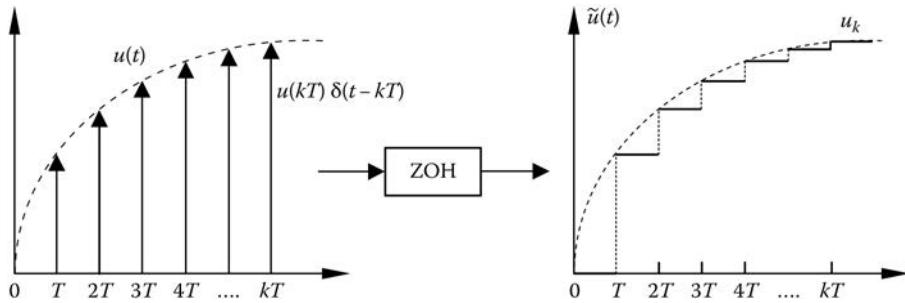


FIGURE 8.97 Impulse sampler feeding ZOH device.

$$G_{\text{ZOH}}(j\omega) = \frac{1 - e^{-j\omega T}}{j\omega} \quad (8.433)$$

Equation 8.432 can be manipulated into the form (Kuo 1980)

$$G_{\text{ZOH}}(j\omega) = T \frac{\sin(\omega T/2)}{\omega T/2} e^{-j(\omega T/2)} \quad (8.434)$$

$$= \left(\frac{2\pi}{\omega_s} \right) \frac{\sin \pi(\omega/\omega_s)}{\pi(\omega/\omega_s)} e^{-j\pi(\omega/\omega_s)} \quad (8.435)$$

where $\omega_s = 2\pi/T$ is the sampling frequency. Equation 8.434 reveals that the ZOH introduces a half sample period ($T/2$) delay, which explains the need for choosing T small when the input contains significant high-frequency components.

The magnitude and phase of $G_{\text{ZOH}}(j\omega)$ are shown in Figure 8.98 for the case where $T = 0.05$ s and $\omega_s = 2\pi/T = 125.67$ rad/s. Note the DC gain $|G_{\text{ZOH}}(j0)| = T$.

For band-limited inputs with cut-off frequency ω_0 , the minimum sampling frequency is $\omega_s = 2\omega_0$. The actual sampling period should be chosen to minimize the attenuation of $G_{\text{ZOH}}(j\omega)$.

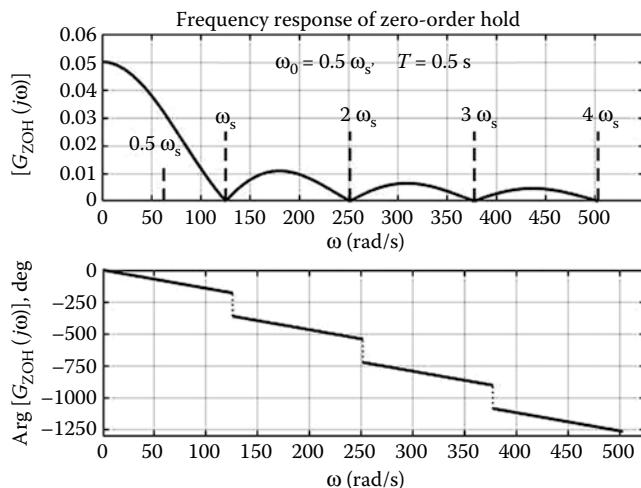


FIGURE 8.98 Frequency response of $G_{\text{ZOH}}(j\omega)$.

over the information band $(0, \omega_0)$. Furthermore, additive noise components above the cutoff frequency will also be passed, since there is no sharp drop in attenuation at ω_0 .

The “c2d” function in the MATLAB control system toolbox introduced in Section 4.10 supports sampling and ZOH signal reconstruction to find the z-domain transfer function given in Equation 8.419. The syntax for calling the “c2d” function using ZOH approximation is `sysd = c2d(sysc,T,'zoh')` where “`sysc`” is created using the control system toolbox command “`tf`” to represent the continuous-time transfer function.

8.6.2 FIRST-ORDER HOLD SIGNAL RECONSTRUCTION

More accurate signal reconstruction methods are possible using polynomial fits through several data points, resulting in different expressions for the z-domain transfer function $G(z)$. The output of a first-order hold circuit that approximates the sampled continuous-time signal by a sequence of linear functions is shown in Figure 8.99.

The analytical expression for the piecewise continuous output of the first-order hold is given by

$$\tilde{u}(t) = u_n + \frac{u_n - u_{n-1}}{T} (t - nT), \quad nT \leq t < (n+1)T \quad (n = 0, 1, 2, \dots) \quad (8.436)$$

where u_{-1} is assumed to be zero. A derivation of $G(z)$ based on a first-order hold approximation is possible using a similar approach to the derivation leading to the z-domain transfer function in Equation 8.419 using the zero-order hold approximation. However, it is quite laborious and unnecessary, since the “c2d” function includes the first-order hold approximation method. The approximation is invoked by issuing also the command `sysd = c2d(sysc,T,'föh')`.

8.6.3 MATCHED POLE-ZERO METHOD

Another approach to developing a discrete-time approximation to a continuous-time system is by the process of matching the z -plane poles and zeros to their s -plane counterparts. This method can be applied to any asymptotically stable, LTI system with nonzero steady-state gain.

Consider an n th-order, stable, LTI system with transfer function $G(s)$. Uniform sampling every T s of the system’s impulse response produces a discrete-time signal from an equivalent n th-order discrete-time system with z -domain transfer function $G(z)$. The n poles of $G(z)$ are obtained by a mapping of the s -plane poles according to

$$z_i = e^{s_i T}, \quad i = 1, 2, \dots, n \quad (8.437)$$

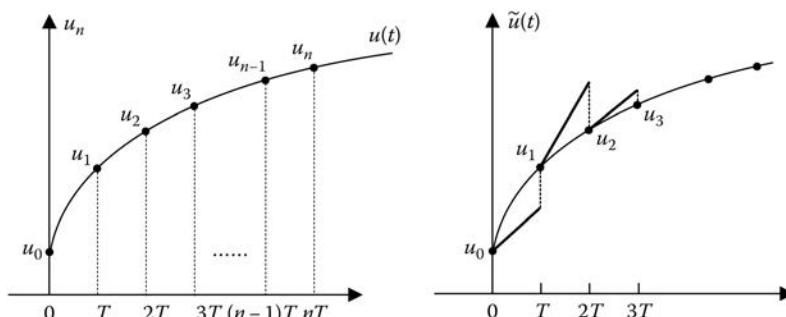


FIGURE 8.99 First-order hold reconstruction of a sampled continuous-time signal.

Two examples of this are

$$G(s) = \frac{1}{s+a} \Rightarrow g(t) = \mathcal{L}^{-1}\{G(s)\} = e^{-at} \quad (8.438)$$

$$gk = g(kT) = e^{-akT} \Rightarrow G(z) = z\{gk\} = \frac{z}{z - e^{-aT}} \quad (8.439)$$

$$G(s) = \frac{s+\alpha}{(s+\alpha)^2 + \beta^2} = \frac{s+\alpha}{[s-(\alpha+j\beta)][s-(\alpha-j\beta)]} \quad (8.440)$$

$$g(t) = \mathcal{L}^{-1}\{G(s)\} = e^{-\alpha T} \cos \beta T \quad (8.441)$$

$$g_k = g(kT) = e^{-\alpha T} \cos \beta kT \quad (8.442)$$

$$G(z) = \frac{z - e^{-\alpha T} \cos \beta T}{z^2 - (e^{-\alpha T} \cos \beta T)z + e^{-2\alpha T}} \quad (8.443)$$

$$= \frac{z - e^{-\alpha T} \cos \beta T}{[z - e^{-\alpha+j\beta} T][z - e^{-\alpha+j\beta} T]} \quad (8.444)$$

When zeros of $G(s)$ are present as in Equation 8.440, they are not mapped into zeros of $G(z)$ according to Equation 8.437. However, in the matched pole-zero method, a discrete-time transfer function is created with the poles and zeros of $G(z)$ determined from Equation 8.437.

Two additional steps complete the process. First, the term z^{n-m} , where m is the order of the numerator polynomial of $G(s)$, is inserted in the numerator of $G(z)$ (Smith 1987). An alternative approach inserts the term $(z+1)^{n-m}$ in the numerator of $G(z)$. Second, the gains of the two transfer functions are matched at some frequency by appropriate choice of a gain term in $G(z)$.

The matched pole-zero method is illustrated for the second-order system in Example 8.12. The poles of $G(s)$ in Equation 8.420 are $s_{1,2} = \alpha \pm j\beta$, ($\alpha = -1$, $\beta = 3$). Since there are no zeros of $G(s)$, $m = 0$ and the z -domain transfer function $G(z)$ is of the form

$$G(z) = K' \frac{z^2}{(z - e^{s_1 T})(z - e^{s_2 T})} \quad (8.445)$$

$$= K' \frac{z^2}{[z - e^{(\alpha+j\beta)T}][z - e^{(\alpha-j\beta)T}]} \quad (8.446)$$

$$= K' \frac{z^2}{z^2 - 2(e^{\alpha T} \cos \beta T)z + e^{2\alpha T}} \quad (8.447)$$

Substituting the given values of α and β into Equation 8.447 results in

$$G(z) = K' \frac{z^2}{z^2 - 2(e^{-T} \cos 3T)z + e^{-2T}} \quad (8.448)$$

The DC gains of $G(s)$ and $G(z)$ are

$$G(s)|_{s=0} = \frac{10}{s^2 + 2s + 10}|_{s=0} = 1 \quad (8.449)$$

$$G(z)|_{z=1} = K' \frac{z^2}{z^2 - 2(e^{-T}\cos 3T)z + e^{-2T}}|_{z=1} \quad (8.450)$$

$$= K' \frac{1}{1 - 2e^{-T}\cos 3T + e^{-2T}} \quad (8.451)$$

Equating the DC gains gives

$$K' = 1 - 2e^{-T}\cos 3T + e^{-2T} \quad (8.452)$$

Substituting K' in Equation 8.452 into Equation 8.448 gives

$$G(z) = \frac{(1 - 2e^{-T}\cos 3T + e^{-2T})z^2}{z^2 - 2(e^{-T}\cos 3T)z + e^{-2T}} \quad (8.453)$$

A frequency response plot of the continuous-time system transfer function $G(s)|_{s=j\omega}$ and the approximating discrete-time system transfer functions $G(z)|_{z=e^{j\omega T}}$ based on the two methods are shown in Figure 8.100 for sampling times of $T = 0.05$ s and $T = 0.25$ s, respectively. $G_1(e^{j\omega T})$ refers to the discrete-time transfer function in Equation 8.426 arrived at by using the ZOH method, and $G_2(e^{j\omega T})$ corresponds to the one in Equation 8.453 obtained using the matched pole-zero method.

The plots extend from zero (DC) to the Nyquist frequency (π/T), which is 62.83 rad/s for $T = 0.05$ s and 12.57 rad/s when $T = 0.25$ s. An accurate (magnitude and phase) approximation of

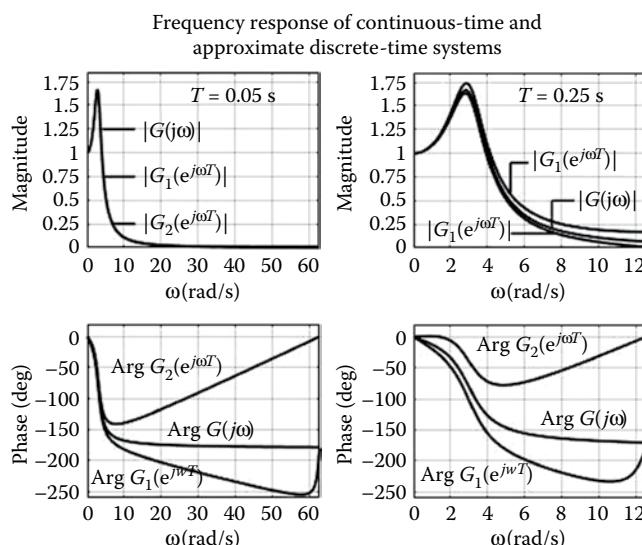


FIGURE 8.100 Frequency response of continuous-time and approximate discrete-time systems.

the continuous-time system frequency response characteristics is possible using the ZOH approximation method or the matched pole-zero technique with $T = 0.05$ s for frequencies up to around 5 rad/s. The magnitude functions for both discrete-time systems and the continuous-time system are nearly identical over the entire range of frequencies shown for $T = 0.05$ s.

The “c2d” function in the MATLAB control system toolbox implements a “modified matched pole-zero” approximation. A $(z + 1)^{(n-m)-1}$ term is inserted in the numerator where m and n are the orders of the numerator and denominator of $G(s)$. The resulting $G(z)$ will contain an $(n - 1)$ st-order polynomial in the numerator. The current output of the n th-order discrete-time system y_k depends on outputs $y_{k-1}, y_{k-2}, \dots, y_{k-n}$ and most importantly only on the past inputs $u_{k-1}, u_{k-2}, \dots, u_{k-n}$. With an n th-order term in the numerator of $G(z)$, y_k will depend on the current input u_k as well. In real-time applications, the current output would have to wait for an A/D read, implementation of the difference equation followed by a D/A write to hardware, all performed in theoretically zero time. The problem is mitigated to a large extent when these operations consume a small fraction of the sample time T .

The matched pole-zero and modified matched pole-zero methods are applied to the continuous-time transfer function in Equation 8.420 in “Ch8_matched_pole.m” with a sampling time of $T = 0.05$ s. Results are as follows:

$$\text{Matched pole-zero: } G(z) = \frac{0.0237 z^2}{z^2 - 1.8811 z - 0.9048} \quad (8.454)$$

$$\text{Modified matched pole-zero: } G(z) = \frac{0.01187(z+1)}{z^2 - 1.8811 z + 0.9048} \quad (8.455)$$

An important property of the ZOH approximation and matched pole-zero methods is related to the stability of the resulting discrete-time systems. Note that the characteristic polynomials of the transfer functions $G(z)$ in Equations 8.426, 8.454, and 8.455 are identical, namely, $z^2 - 2(e^{-T} \cos 3T)z + e^{-2T}$. The continuous-time system poles are mapped to the z -plane according to Equation 8.437 in each case. Consequently, continuous-time system poles in the left-hand plane are mapped to the interior of the Unit Circle in the z -plane and, therefore, produce stable discrete-time modes as well.

8.6.4 BILINEAR TRANSFORM WITH PREWARPING

The use of trapezoidal integration to discretize a continuous-time system with transfer function $G(s)$ was discussed in Section 4.7. The z -domain transfer function of the discrete-time system approximation was shown to be

$$G(z) = G(s) \Big|_{s \leftarrow \frac{2(z-1)}{T(z+1)}} \quad (8.456)$$

An alternate derivation of Equation 8.456 is based on the transformation $z = e^{Ts}$, which can be written in terms of a pair of infinite series expansions according to

$$z = \frac{e^{(T/2)s}}{e^{(-T/2)s}} = \frac{1 + (Ts/2) + (1/2!)(Ts/2)^2 + (1/3!)(Ts/2)^3 + \dots}{1 - (Ts/2) + (1/2!)(Ts/2)^2 - (1/3!)(Ts/2)^3 + \dots} \quad (8.457)$$

Truncating both series after the linear term gives

$$z = \frac{1 + (T/2)s}{1 - (T/2)s} \quad (8.458)$$

Solving for s in Equation 8.458 gives

$$s = \frac{2}{T} \left(\frac{z-1}{z+1} \right) \quad (8.459)$$

Equation 8.459 is known as the bilinear transform, and the process for obtaining the discrete-time approximation is commonly referred to as Tustin's method. The left half of the s -plane consisting of points $s = \sigma + j\omega$, $-\infty < \sigma < 0$ is mapped into the interior of the Unit Circle, $|z| < 1$. Consequently, the method produces stable discrete-time systems regardless of the step size T provided the continuous-time system is stable. For this reason, it is among the most popular methods for simulation of continuous-time systems.

The frequency response of discretized systems obtained using the bilinear transform in Equation 8.459 is examined by considering the image of points along the $j\omega$ axis, that is, $s = j\omega$ $-\infty < \omega < \infty$. From Equation 8.458 with $s = j\omega$,

$$z = \frac{1 + (T/2)j\omega}{1 - (T/2)j\omega} = 1e^{j\theta} \quad (8.460)$$

where

$$\theta = 2 \tan^{-1} \left(\frac{\omega T}{2} \right), \quad -\infty < \omega < \infty \quad (8.461)$$

The entire length of the $j\omega$ axis from $-j\infty$ (pt A) to $j\infty$ (pt C) is mapped one-to-one into the Unit Circle starting at $\theta = -\pi$ (pt A') to $\theta = \pi$ (pt C') (see Figure 8.101).

Compressing the $j\omega$ axis into the Unit Circle according to Equation 8.461 results in a warping of the frequency response. This can be overcome by prewarping a critical frequency, say ω_0 , in the s -plane before applying the bilinear transform to the continuous-time transfer function $H(s)$.

Frequency response of the resulting z -domain transfer function $\hat{H}(z)$ and the continuous-time transfer function $H(s)$ will agree at the selected critical frequency, that is,

$$H(s)|_{s=j\omega_0} = \hat{H}(z)|_{z=e^{j\omega_0 T}} \quad (8.462)$$

$$\hat{\omega}_0 = \frac{2}{T} \tan \left(\frac{\omega_0 T}{2} \right) \quad (8.463)$$

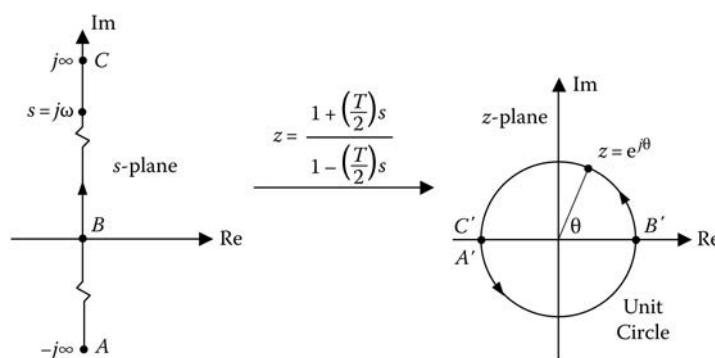


FIGURE 8.101 Bilinear transform mapping of the imaginary axis in the s -plane.

The following example illustrates the process of prewarping a second-order continuous-time filter transfer function to force agreement in the frequency response functions at the natural frequency of the filter.

EXAMPLE 8.13

An analog filter is described by

$$H(s) = \frac{s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (\zeta = 0.25, \omega_n = 1000 \text{ rad/s}) \quad (8.464)$$

- a. Find $H(z)$ using the bilinear transform with a sampling time of $T = 0.001$ s.
- b. Find the transfer function $\hat{H}(s)$ resulting from prewarping the natural frequency ω_n .
- c. Find $\hat{H}(z)$ using the bilinear transform on the prewarped transfer function $\hat{H}(s)$.
- d. Plot the magnitude and phase of $H(s)$, $H(z)$, and $\hat{H}(z)$ on the same graph and comment on the results.
- a. Substituting the filter parameter values ζ and ω_n into Equation 8.464 gives

$$H(s) = \frac{s + 10^6}{s^2 + 500s + 10^6} \quad (8.465)$$

$H(z)$ is obtained by replacing s with the right-hand side of Equation 8.459. The MATLAB control system toolbox functions “BILINEAR” and “c2d” are both designed to facilitate implementation of the bilinear transform. One form of the function “BILINEAR,” which is applicable in this case, is

```
[NUMd, DENd] = BILINEAR (NUM, DEN, FS)
```

where the parameters “NUM” and “DEN” are row vectors describing the numerator and denominator of $H(s)$ in descending powers of s , and “FS” is the sampling frequency in Hz. The numerator and denominator of $H(z)$ are specified in the output arrays “NUMd” and “DENd.”

The M-file “Ch8_Ex8_13.m” contains the call to the “BILINEAR” function and the result is

```
NUMd = 0.1670 0.333 0.1663
DENd = 1.000 -1.0000 0.6667
```

Invoking the “c2d” function with “sysd = c2d (sync, T, ‘tustin’)” results in

```
Transfer function:
0.167z^2 + 0.3333z + 0.1663
-----
z^2-z + 0.6667
sampling time = 0.001 sec
```

in agreement with the results obtained using the “BILINEAR” command.

- b. Prewarping the natural frequency using Equation 8.463 yields

$$\hat{\omega}_n = \frac{2}{T} \tan\left(\frac{\omega_n T}{2}\right) = \frac{2}{0.001} \tan\left(\frac{1000(0.001)}{2}\right) = 1092.6 \text{ rad/s} \quad (8.466)$$

The prewarped transfer function is therefore

$$\hat{H}(s) = \frac{s + \hat{\omega}_n^2}{s^2 + 2\zeta\hat{\omega}_n s + \hat{\omega}_n^2} = \frac{s + 1.1938 \times 10^6}{s^2 + 546.3s + 1.1938 \times 10^6} \quad (8.467)$$

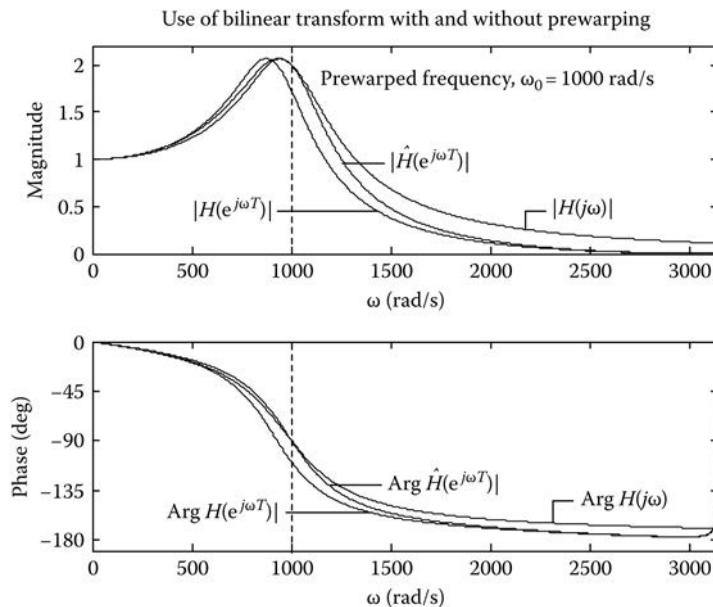


FIGURE 8.102 Illustration of prewarping critical frequency prior to bilinear transform.

- c. $\hat{H}(z)$ results from the bilinear transformation applied to the transfer function in Equation 8.467. Equivalently, the MATLAB statement

```
Sysd_prewarp = c2d(sysc, T, 'prewarp', w_crit)
```

can be found in "Ch8_Ex8_13.m" with "w_crit" set equal to the natural frequency $\omega_n = 1000$ rad/s. The resulting z-domain transfer function appears as

$$\frac{0.1902 z^2 + 0.3798z + 0.1896}{z^2 - 0.8928z + 0.6524}$$

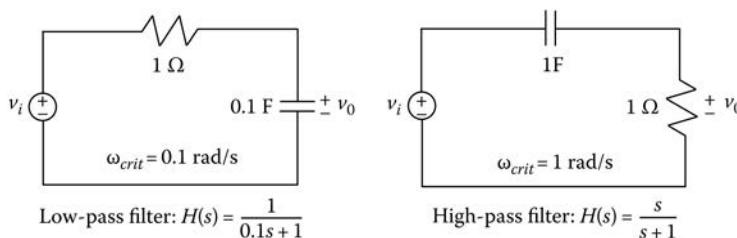
- d. The magnitude and phase plots for the continuous-time system frequency response $H(j\omega)$ and the two discrete-time systems (with and without prewarping the natural frequency $\omega_n = 1000$ rad/s) are shown in Figure 8.102. As expected, Equation 8.462 is verified at the critical frequency of 1000 rad/s.

There is one additional method included in the "c2d" function, which requires the Signal Processing Toolbox for converting continuous-time models to discrete-time models. It is called the impulse invariant method. It is predicated on making the discrete-time system impulse response proportional to the sampled values of the continuous-time system impulse response function. The syntax for implementing this method is "sysd = c2d (sync,T, 'imp')."

EXERCISES

- 8.57 Implement the "c2d" function using the zero-order hold approximation in Example 8.12 and show that the results are consistent with Equations 8.426 through 8.428 when $T = 0.05$ s.
- 8.58 Derive the expression for $G_{FOH}(s)$, the transfer function of a first-order hold driven by an impulse sampler. Plot the frequency response for the case when $T = 0.05$ s, and compare the result with the frequency response plot of a ZOH with $T = 0.05$ s shown in Figure 8.98.
- 8.59 Redo Example 8.12 using
 - The first-order hold approximation and compare the results with the ZOH approximation method

- b. The bilinear transform method and compare the results with the ZOH approximation method
- 8.60 Apply the bilinear transform to the prewarped continuous-time transfer function $\hat{H}(s)$ in Equation 8.46 and compare the result with z -domain transfer function $\hat{H}(z)$ given in part (c) of Example 8.13.
- 8.61 The circuits in below figure are low- and high-pass filters.



- a. Find the z -domain transfer function of the approximating discrete-time filters based on the bilinear transform with sample time $T = 0.005\text{ s}$ for the low-pass filter and $T = 0.05\text{ s}$ for the high-pass filter.
- b. Repeat part (a) after first prewarping the critical frequencies of each filter.
- c. Compare the frequency responses of the continuous-time and discrete-time filters.
- d. Find and plot the unit step responses of the low-pass filter and its discrete-time approximations.
- e. Repeat part (d) for the high-pass filter.
- 8.62 For the transfer function $G(s) = 10/(s^2 + 2s + 10)$ in Example 8.12,
- Convert to state-space form $\dot{x} = Ax + Bu$, $y = Cx + Du$ using the MATLAB function “tf2ss.”
 - Convert the continuous-time state-space model to discrete-time form using the MATLAB function “c2dm.” Choose the sample time $T = 0.05\text{ s}$ and specify “zoh” as the method of approximation.
 - Obtain the unit step response by solving the discrete-time state model equations recursively and plot the results.
- 8.63 For the filters in Exercise 8.62,
- Obtain discrete-time filter approximations to each using the impulse invariant method.
 - Compare the frequency response functions of the continuous- and discrete-time filters.
 - Compare the impulse response functions of the continuous- and discrete-time filters.

8.7 CASE STUDY: LEGO MINDSTORMS™ NXT

8.7.1 INTRODUCTION

In the November 2008 issue of *Mechanical Engineering*, the American Society of Mechanical Engineers surveyed its members for the trend they thought would have the most significant impact 10 years hence. In second place, with 26% of the responses, was Mechatronics—the integration of mechanical and electronic design. Incidentally, first place (28%) went to nanotechnology and micro-electromechanical systems—devices that are demanded by mechatronicians.

The discipline of Mechatronics is a nexus of four technical sub-disciplines: mechanical systems, electronics systems, control systems, and computers. The intersection of mechanical and electronic systems is electromechanics; electronic and control systems intersect at control electronics; control systems and computers combine to form digital control systems; and finally, computers and mechanical systems form mechanical computer-aided design (CAD). Figure 8.103 displays these relationships graphically.

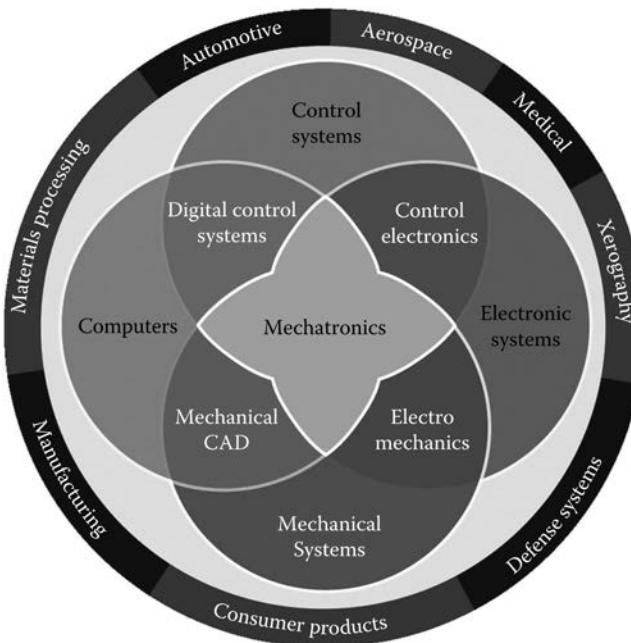


FIGURE 8.103 Mechatronics: mechanical, electronics, and control systems, and computers.

From a feedback control systems perspective, mechatronics is the implementation or realization of a controller design. A feedback control system involves either tracking a changing input or regulating a constant reference input. The block diagram for a generic feedback control system is given in [Figure 8.104](#).

The feedback control system diagram begins with the reference input, which is fed into a comparator. The comparator measures the difference between the reference input and the feedback signal, generating an error signal. In order for the computer to process the error signal, it must first be converted by an A2D converter. Once the control signal is processed, it is converted by a D2A converter. It is this signal, fed into the actuator, that controls the process. Note that this signal may need to be amplified in order to actuate the controlling hardware. A sensor is connected to the process in order to measure the performance of the system. The sensor provides the feedback signal, which is used in the comparator. This loop is continually repeated for the process to be controlled. Simply stated, control design synthesis involves mathematically modeling and analyzing a physical process (e.g., missile airframe) and then designing a controller (e.g., acceleration autopilot). Simulink's graphical environment facilitates this “model-based design” approach. These steps of controller design synthesis are usually performed on the same host development platform, that is, a personal computer.

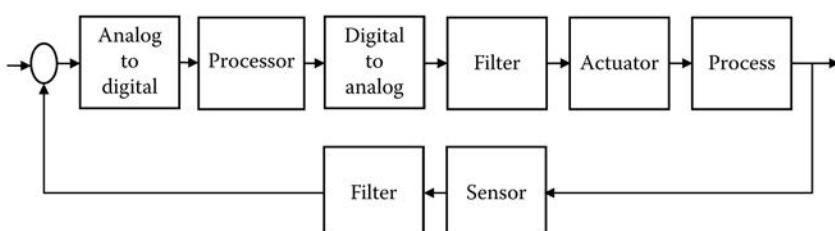


FIGURE 8.104 Feedback control system block diagram.

Beyond modeling, analysis, and design, mechatronics adds the implementation step. In this step, the controller design is realized on the actual (e.g., flight) hardware. Generated source code is compiled and assembled for a particular microprocessor that is executed for the digital controller design. Source code interfaces for actuators and sensors, called device drivers, are typically provided by vendors that manufacture these various pieces of hardware. An application that facilitates this extended development process is called an Integrated Development Environment (IDE). By adding MATLAB's Real-Time Workshop to the host's suite of tools, code generation, compilation, and assembly for a specific microprocessor are enabled. The repetitive process of making changes to the controller design in the Simulink model, generating, compiling, assembling, downloading, and running code on the microprocessor is known as rapid prototyping. This allows the engineer to build a little and test a little, thereby rooting out errors early in the development process and potentially avoiding the larger costs associated with redesigning the system late in the development process.

By combining the popular Lego MindstormsTM NXT (henceforth referred to NXT) robotics platform with MATLAB's Simulink and Real-Time Worshop tools, this development process can be demonstrated end to end. Therefore, the remainder of this section is devoted to

- Product requirements, software download, and installation
- Creating a Simulink model that provides a noisy input signal and then running the “unfiltered” model on the NXT to observe how the noisy signal affects the physical motor
- Modifying the Simulink model by adding a discrete-time Kalman filter (DTKF) (Section 5.12), which filters the noisy input signal and then running the “filtered” model on the NXT observing the effect of the filtered signal on the physical motor

8.7.2 REQUIREMENTS AND INSTALLATION

The software and hardware requirements are

- MATLAB, Simulink, Real-Time Workshop, and Real-Time Workshop's Embedded Coder available from The Mathworks
- CygwinTM and the GNU ARMTM compiler available as a download
- NXT hardware and corresponding device drivers available from Lego

Once The Mathworks software is installed and functioning properly, the third-party software download and installation (CygwinTM and the GNU ARMTM compiler) is facilitated by a MATLAB m-file script. CygwinTM is a UNIX shell environment that runs on Windows and the GNU ARMTM compiler compiles C source code for the ARM processor that runs on the NXT. The script is available from The Mathworks' Web site at <http://www.mathworks.com> by searching for “ECRobot Installer.” (ECRobot is an abbreviation for Embedded Controller Robot.) Locate the hyperlink “Download the ECRobot installer” to download (ecrobot_installer_v1_2.zip at the time). Extract the files and then follow the README.pdf instructions. The ECRobotInstaller contains three MATLAB m-file scripts:

- A script “download_ecrobot_tools” to download all the necessary software. *Note:* Before running the script in Step 1: Automated Download of the README file, it may need to be edited to accommodate the current version of nxtOSEK.
- A script “install_ecrobot_tools” to configure and install the necessary software.
- A script “update_nxt_firmware” that updates the firmware on the NXT to run ARM binary files

Note: At the time of this writing, sg.exe had been removed from nxtOSEK. Therefore, sg.exe is obtained by downloading and extracting osek_os-1.1.1zh for nxtOSEK from the Web site <http://lejos-osek.sourceforge.net/download.htm>. Copy/toppers_osek/sg/sg.exe to the nxtOSEK/toppers_osek/sg directory.

At this point, follow Step 5: Verify that everything works as outlined in the README file. Note that the README file contains answers to commonly asked questions. If you have additional questions, please e-mail: mindstorms@mathworks.com.

8.7.3 NOISY MODEL

In this section, a Simulink model is created that generates a noisy input signal, which drives an NXT motor. First, the model is built and simulated to view the noisy signal. Then, C source code is generated, compiled, assembled, downloaded, and run on the NXT to observe how the noisy signal affects the physical motor.

Knowing ahead of time that C source code will be generated with MATLAB's Real-Time Workshop, the model is architected such that the portion of the model that is generated into C source code exists within a function—where the function is driven by a scheduler. Upon starting Simulink, a new block set has been installed and added to the Simulink Library Browser called "ECRobot NXT Blockset." In this blockset, there is a block called "ExpFcnCalls Scheduler." This block generates function-call events according to the rate specified within the block parameters. For the model shown in Figure 8.105, this block generates function calls at the rate of 100 ms. The function-call scheduler expects to be connected to a demux block in case there are multiple functions being called by the scheduler. Even though there is only one function, a demux block is still necessary. The output of the demux block is the input into a subsystem block—which contains the function. However, at this top-level, a servo motor interface block (from the ECRobot NXT Blockset) is connected to a scope, so the (noisy) output can be viewed. The blocks nxtbuild('Noisy', 'build') and nxtbuild('Noisy', 'rxflash') are annotation blocks with call-back functions enabled to execute the corresponding command in MATLAB.

By double clicking on the subsystem block named "Noisy," the function-call subsystem (from the standard Simulink library blockset) is seen as in Figure 8.106.

By double clicking on the function-call subsystem, the model that generates the noisy signal is seen as in Figure 8.107.

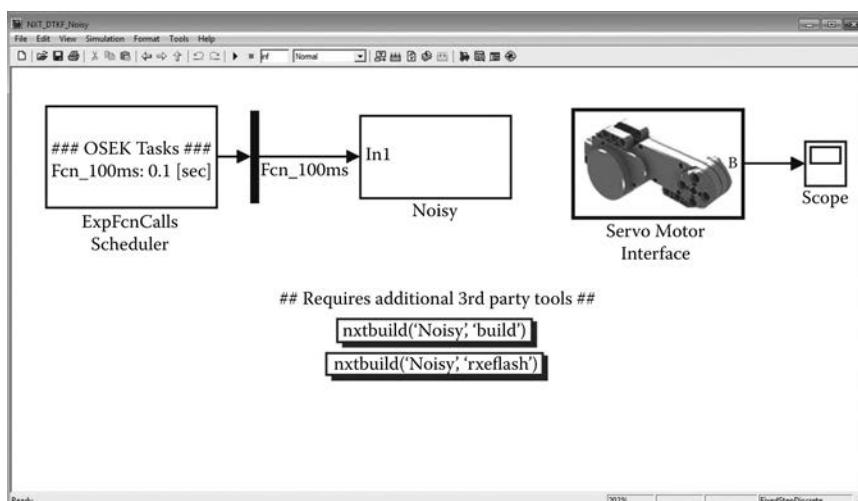


FIGURE 8.105 Top-level block diagram of the noisy model.

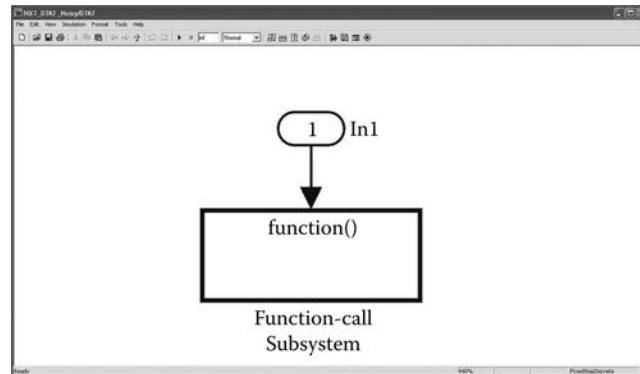


FIGURE 8.106 Function-call subsystem.

The creation of the function-call sSubsystem automatically places the f() block in this subsystem to indicate that the included elemental blocks are part of the function. A random number block with a mean of 32 and a variance of 32^2 generates the random signal. The saturation block limits possible signals to ± 100 as these are the limits of the NXT motor signals. The data type conversion is set to int8 to represent the signed 8-bit integer, that is, -128 to 127. (The random number, saturation, and data type conversion blocks are all part of the standard Simulink library blockset.) Finally, the Servo Motor Write block (from the ECRobot NXT blockset) is connected to port B. Port B is the second output (top/left) from the Lego “brick” as seen in [Figure 8.108](#).

Upon running a Simulink simulation, the noisy data may be viewed from the scope block as seen in [Figure 8.109](#).

Alternatively, the noisy data are available in the MATLAB Workspace as a variable “structure with time” named “noisy.” A plot of this noisy data is shown in [Figure 8.110](#).

The next part of this exercise is to generate, compile, assemble, download, and run C source code for the “noisy” function on the NXT to observe how the noisy signal affects the physical motor.

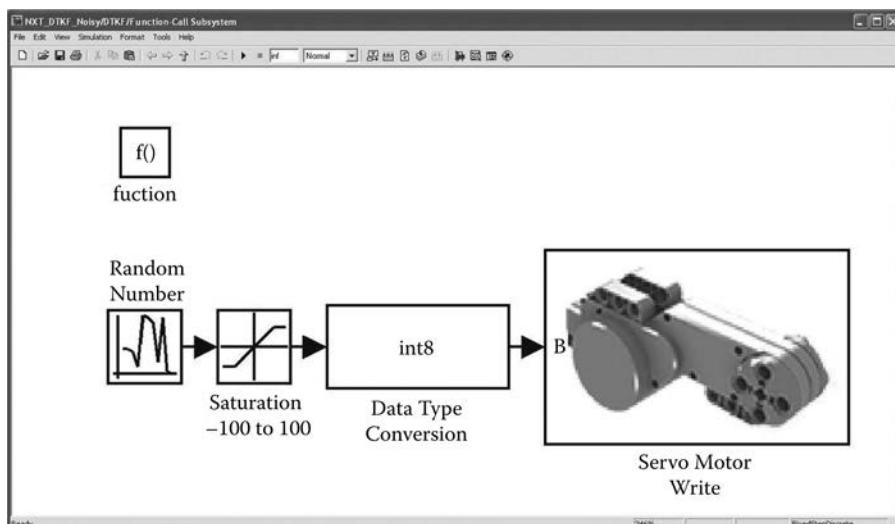


FIGURE 8.107 Noisy signal model.



FIGURE 8.108 Lego Mind-storms™ NXT “brick.” (LEGO® and LEGO® Mind-storms® NXT™ are trademarks of the LEGO® Group, which does not sponsor nor endorse this book. This photo of the LEGO® Mind-storms® NXT™ brick is used here with permission. © 2010 The LEGO® Group.)

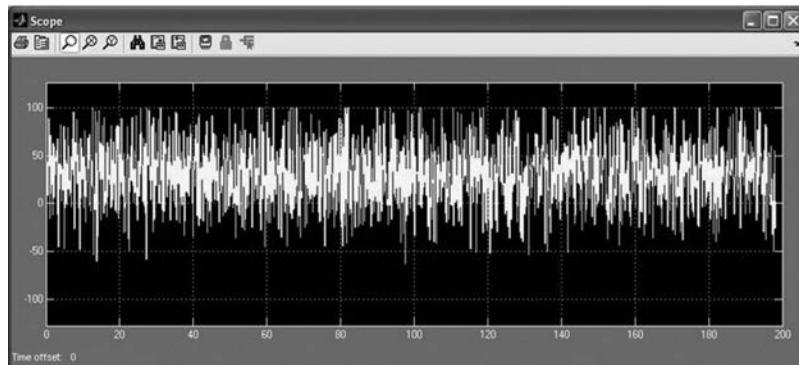


FIGURE 8.109 Noisy output viewed from the Scope Block.

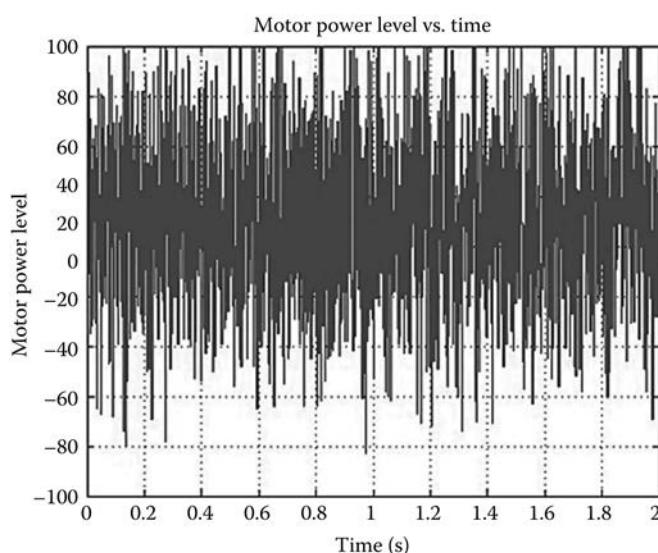


FIGURE 8.110 MATLAB plot of the noisy data.

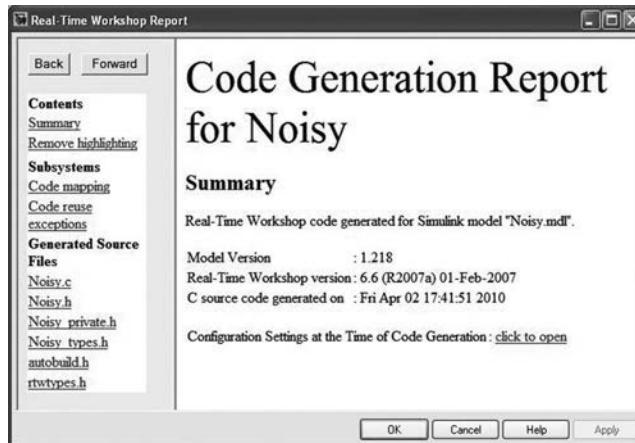


FIGURE 8.111 Real-time workshop report.

By clicking on the annotation block `nxtbuild` ('Noisy', 'build'), the Real-Time Workshop code generator is invoked, which creates C source code and corresponding header files from the Simulink function. The following text appears in MATLAB's Command Window.

```
### Starting Real-Time Workshop build procedure for model: Noisy
### Successful completion of Real-Time Workshop build procedure for
model: Noisy
### Generating ECRobot NXT scheduler file(s) for model: Noisy
### Successful completion of ECRobot NXT scheduler file (s) generation
for model: Noisy
### Executing GNU-ARM toolchain for building executable ...
```

Successful C source code and header file generation result in the Real-Time Workshop Report appearing as shown in [Figure 8.111](#).

On the left side of the Real-Time Workshop Report window, hyperlinks indicate the various sections of the C source code related to the function from the Simulink model. In particular, by clicking on "Noisy.c," one can view portions of the code that correspond directly with the elemental blocks that constitute the function of the Simulink model. The rest of the messages in MATLAB's Command Window correspond to the build portion of compiling and assembling the binary image file named "Noisy.rxe."

```
.
.
.
(many messages corresponding to the build process, i.e., compiling and
assembling)
```

```
.
.
.
Generating binary image file: Noisy_rom.bin
Generating binary image file: Noisy_ram.bin
Generating binary image file: Noisy.rxe.
```

Once the binary image file "Noisy.rxe" has been created, click on the annotation block `nxtbuild` ("Noisy," "rxfash") to load the binary image into the flash memory of the NXT. For this part of the procedure, MATLAB's Command Window shows the following:

```
### Execute NeXTTool for uploading a program to the enhanced NXT standard
firmware:./nxtprj/Noisy.rxe
Executing NeXTTool to upload Noisy.rxe ...
```

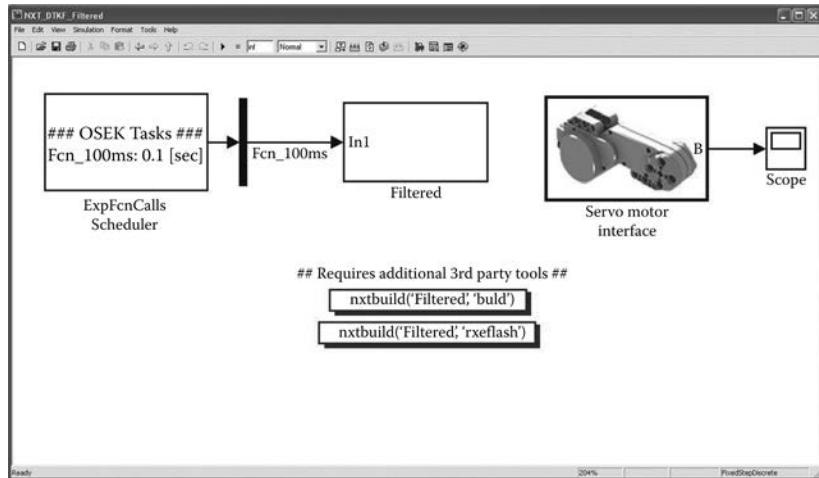


FIGURE 8.112 Top-level block diagram for the filtered model.

Noisy.rxe = 26144

NeXTTool is terminated.

Note: NeXTTool is a utility that transfers files from the PC to the NXT.

At this time, the NXT is ready to run the noisy motor program. Be certain there is a motor connected to Port B on the brick. Upon running this program, the motor indeed runs erratically, exhibiting its response to the noisy input.

8.7.4 FILTERED MODEL

In this section, the noisy Simulink model is modified by adding a DTKF (Section 5.12), which filters the noisy input signal. The model is simulated in Simulink to view the filtered signal. Then, as before, C source code is generated, compiled, assembled, downloaded, and run on the NXT to observe how the physical motor responds to the filtered signal.

As shown in Figure 8.112, the top-level block for the filtered model is similar to that of the noisy model, except the name of the subsystem block has been changed to “Filtered” and the annotation blocks have been updated as well.

By double clicking on the subsystem block named “Filtered,” the function-call subsystem (from the standard Simulink library blockset) is seen as in Figure 8.113.

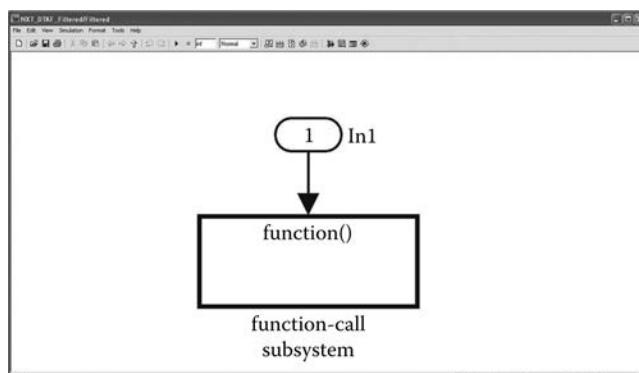


FIGURE 8.113 Function-call subsystem.

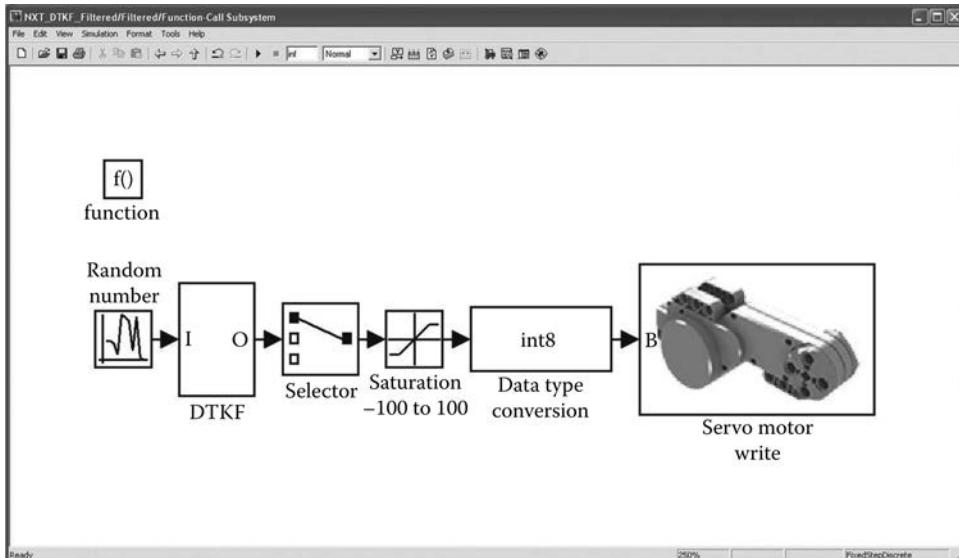


FIGURE 8.114 Filtered signal model.

By double clicking on the function-call subsystem, the model that generates the filtered signal is seen as in Figure 8.114.

A subsystem block named “DTKF” has been added to the model in order to filter the noisy signal. This is the same DTKF that was developed in Section 5.12 for the meteorite. However, rather than setting the variables in MATLAB (as in Section 5.12), the parameters are set directly in the Simulink blocks. Also, while the DTKF had three states: position, velocity, and acceleration, the position of the meteorite was of primary interest in that example. Therefore, the first output of the DTKF (corresponding to position) is selected as the input to the saturation block. The major subsystems of the DTKF are the a priori and a posteriori calculations of the state and covariance matrix updates as seen in Figure 8.115.

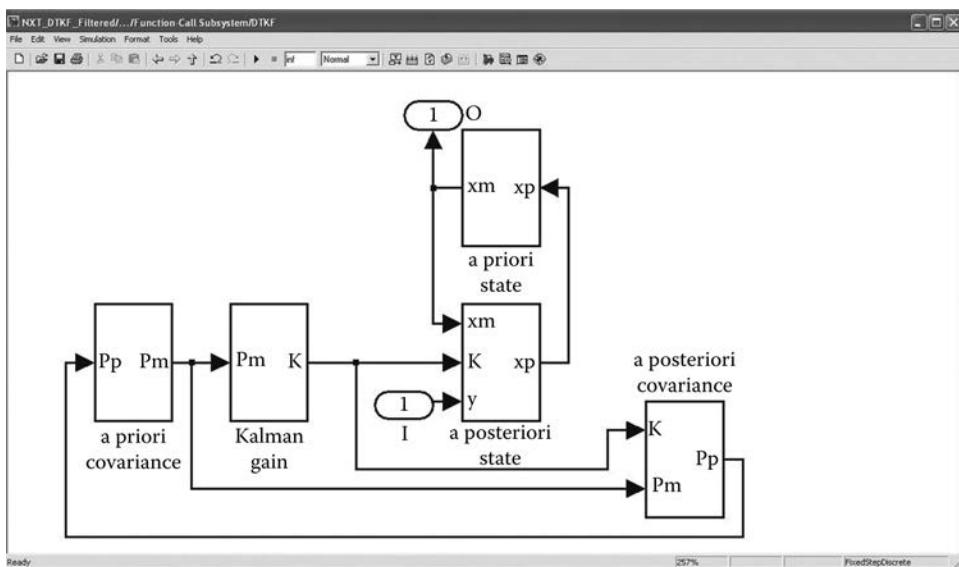


FIGURE 8.115 Discrete-time Kalman filter subsystems.

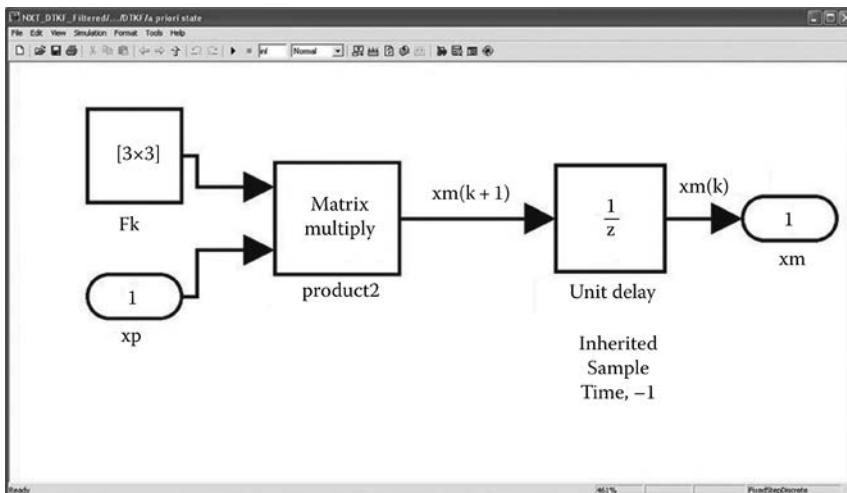


FIGURE 8.116 “A priori” state.

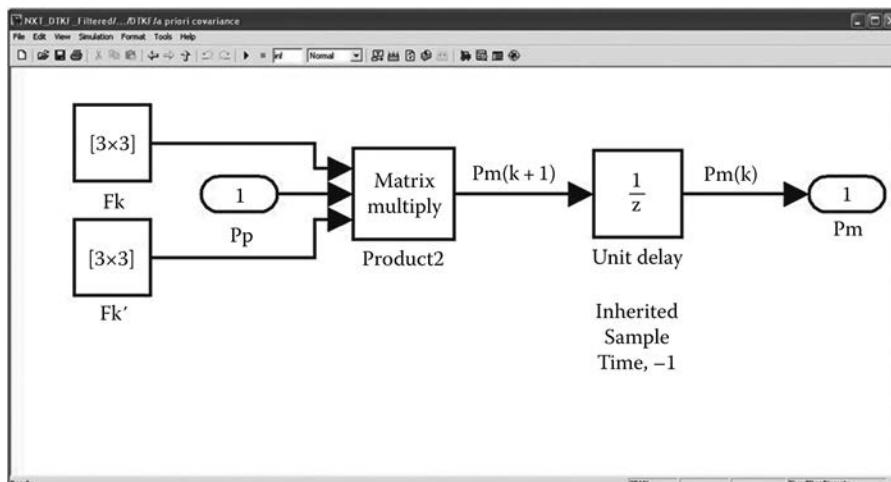


FIGURE 8.117 “A priori” covariance.

The details of the DTKF are shown in Figures 8.116 through 8.120. Notice in Figure 8.116 (a priori state) and 8.117 (a priori covariance) that the unit delay block inherits the sample time, that is, the function-call scheduler time, by setting sample time equal to -1 in the block properties.

Upon running a Simulink simulation, the filtered data may be viewed from the Scope Block as seen in Figure 8.121. Notice that the filtered value appears to be approximately 32, which was the mean of the random number block.

Alternatively, the filtered data are available in the MATLAB Workspace as a variable “structure with time” named “filtered.” A plot of the filtered data is shown in Figure 8.122.

By clicking on the annotation block `nxtbuild("Filtered," "build")`, the Real-Time Workshop code generator is invoked, which creates C source code and corresponding header files from the Simulink function. The following text appears in MATLAB’s Command Window.

```
### Starting Real-Time Workshop build procedure for model: Filtered
### Successful completion of Real-Time Workshop build procedure for
model: Filtered
```

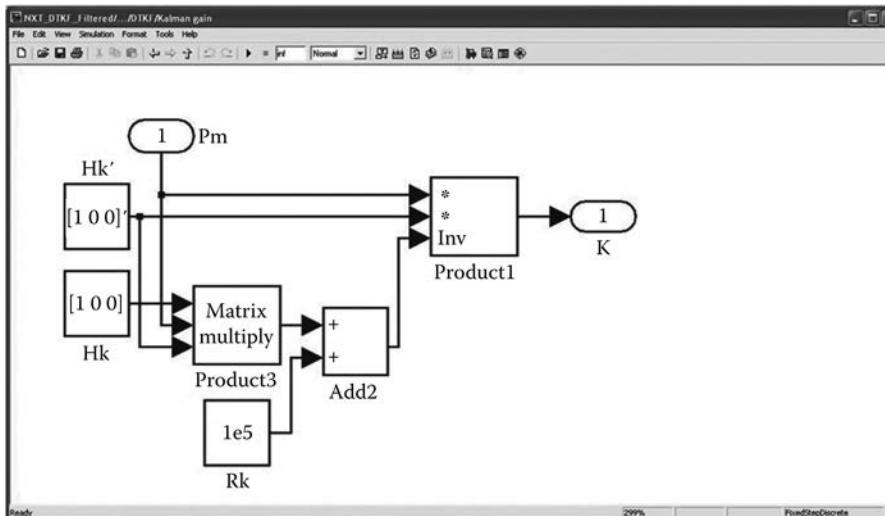


FIGURE 8.118 Kalman gain.

```
### Generating ECRobot NXT scheduler file(s) for model: Filtered
### Successful completion of ECRobot NXT scheduler file(s) generation for
model: Filtered
### Executing GNU-ARM toolchain for building executable ...
```

Successful C source code and header file generation result in the Real-Time Workshop Report appearing as shown in [Figure 8.123](#).

On the left side of the Real-Time Workshop Report window, hyperlinks indicate the various sections of the C source code related to the function from the Simulink model. In particular, by clicking on “Filtered.c,” one can view portions of the code that correspond directly with the elemental blocks and the DTKF blocks that constitute the function of the Simulink model. The rest of the messages in MATLAB’s Command Window correspond to the build portion of compiling and assembling the binary image file named “Filtered.rxe.”

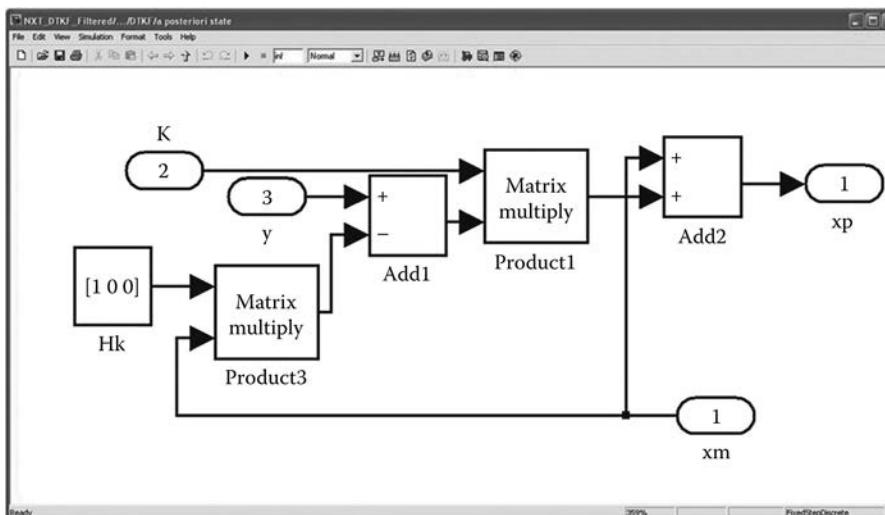
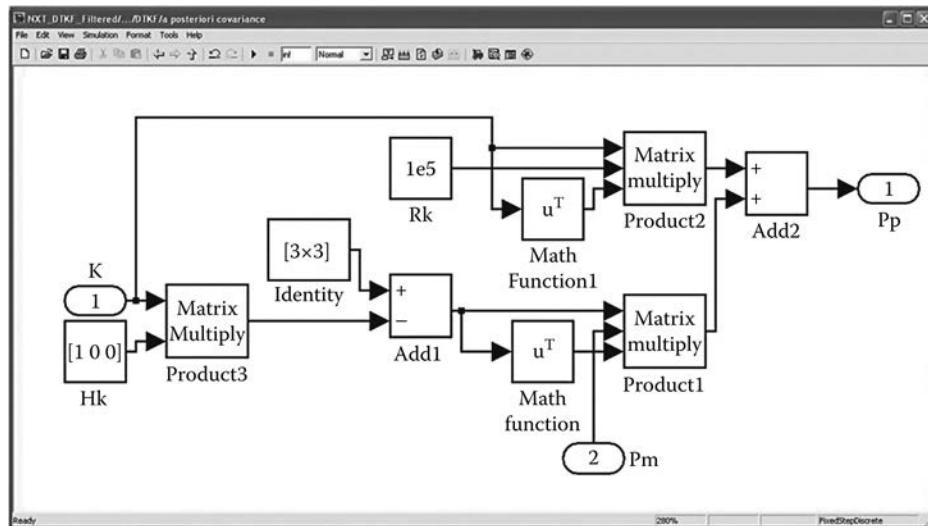
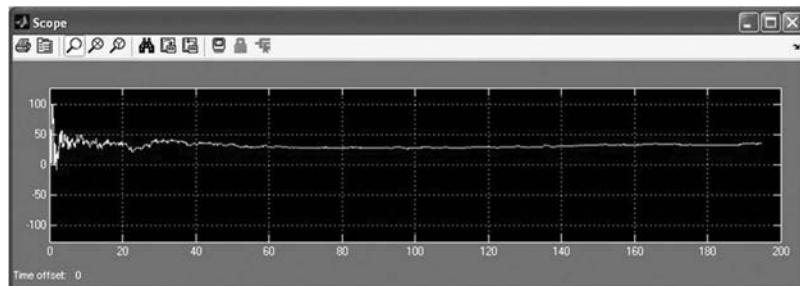
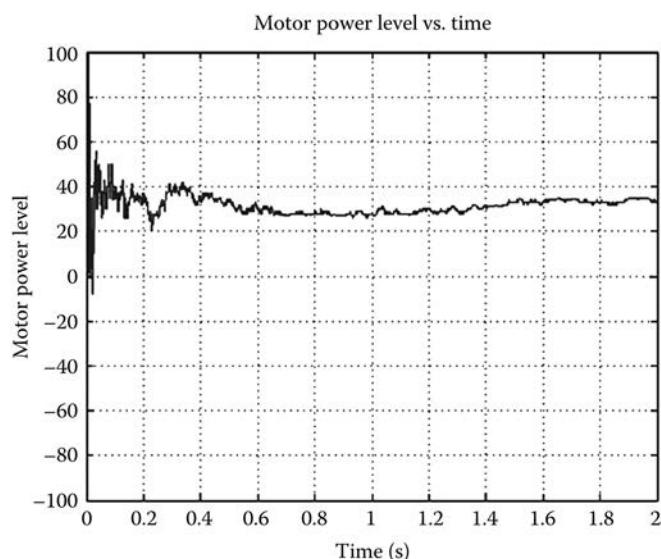


FIGURE 8.119 “A posteriori” state.

**FIGURE 8.120** “A posteriori” covariance.**FIGURE 8.121** Filtered output viewed from the Scope Block.**FIGURE 8.122** MATLAB plot of the filtered data.

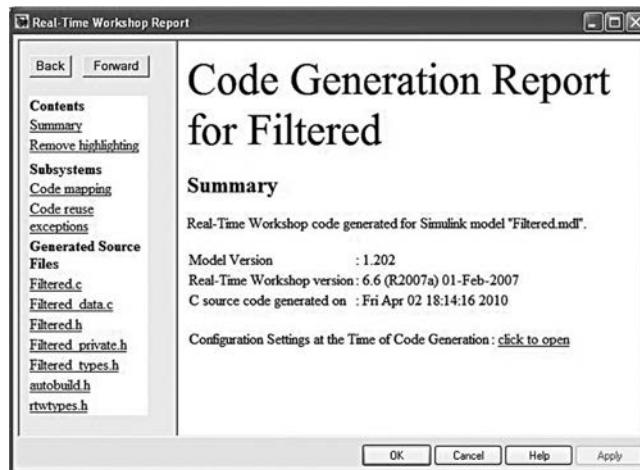


FIGURE 8.123 Real-time workshop report.

(many messages corresponding to the build process, i.e., compiling and assembling)

Generating binary image file: Filtered_rom.bin
 Generating binary image file: Filtered_ram.bin
 Generating binary image file: Filtered.rxe

Once the binary image file “Filtered.rxe” has been created, click on the annotation block nxt-build (“Filtered,” “rxfetch”) to load the binary image into the flash memory of the NXT. For this part of the procedure, MATLAB’s Command Window shows the following:

```
### Execute NeXTTool for uploading a program to the enhanced NXT standard
firmware:./nxtprj/Filtered.rxe
Executing NeXTTool to upload Filtered.rxe ...
Filtered.rxe = 28720
NeXTTool is terminated.
```

At this time, the NXT is ready to run the filtered motor program. As before, be sure there is a motor connected to Port B on the NXT brick. Upon running this program the motor indeed runs smoothly, exhibiting its response to the filtered input.

8.7.5 SUMMARY

In this Case Study, the build-a-little/test-a-little rapid prototyping development process was facilitated by an IDE. The technology (software and hardware) that enabled the IDE was made available by tools from The Mathworks (MATLAB, Simulink, Real-Time Workshop, and RTW’s Embedded Coder), Cygwin™, GNU ARM™, and Lego (Mindstorms™ NXT).

Once the IDE is enabled with the technology, the intent was to demonstrate how easy it is to rapidly change the model from within Simulink, and then with two mouse clicks: generate, compile, assemble, download, and run the model on the NXT.

It is left as an exercise for the student to now unleash his/her creativity in developing various applications on this platform.

EXERCISES

- 8.64 While a noisy input signal was generated from within Simulink, modify the model such that a sensor provides the input. Examine the various sensors that are available physically, as well as from the ECRobot NXT Blockset in the Simulink Library Browser. For additional assistance, see the samples that were part of the software installation, for example, TestUltrasonicSensor.mdl.

References

- Akai, T. J., *Applied Numerical Methods*, John Wiley & Sons, New York, 1994.
- Allen, R. W. and T. Rosenthal, Systems technology/requirements for vehicle dynamics simulation models, Society of Automotive Engineers, SAE 941075, 1994.
- Aycin, M. and R. Benekohal, Stability and performance of car-following models in congested traffic, *Journal of Transportation Engineering*, 127, 2–12, 2001.
- Banks, J., J. S. Carson II et al., *Discrete-Event System Simulation*, 4th edn., Pearson Prentice-Hall, Upper Saddle River, NJ, 2005.
- Baruh, H., *Analytical Dynamics*, WCB/McGraw-Hill, Boston, MA, 1999.
- Beltrami, E., *Mathematical Models in the Social and Biological Sciences*, Jones and Bartlett, Boston, MA, 1993.
- Bender, J. G. and R. E. Fenton, A study of automatic car following, *IEEE Transactions on Vehicular Technology*, VT-18, 134–140, 1966.
- Borse, G. J., *Numerical Methods with MATLAB*, PWS Publishing, Boston, MA, 1997.
- Bracewell, R., *The Fourier Transform and Its Applications*, McGraw-Hill, New York, 1986.
- Braun, M., *Differential Equations and Their Applications*, Springer-Verlag, New York, 1978.
- Brown, D. and P. Rothery, *Models in Biology: Mathematics, Statistics and Computing*, John Wiley & Sons, West Sussex, U.K., 1993.
- Bryson, A. E., *Dynamic Optimization*, Addison-Wesley, Menlo Park, CA, 1999.
- Buckley, P., *Techniques of Process Control*, John Wiley & Sons, New York, 1964.
- Burns, R. S., *Advanced Control Engineering*, Butterworth Heinemann, Oxford, U.K., 2001.
- Cadzow, J. A., *Discrete-Time Systems—An Introduction with Interdisciplinary Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- Canova, B. S., P. H. Christensen, M. D. Lee, B. R. Tripp, M. H. Pack, and D. L. Pack, Simulation to support operational testing: A practical approach, in *Proceedings of the 1999 Winter Simulation Conference*, pp. 1071–1078, 1999.
- Chapra, S. and R. Canale, *Numerical Methods for Engineers with Software Programming Applications*, 4th edn., McGraw-Hill, New York, 2002.
- Close, C. M., *Modeling and Analysis of Dynamic Systems*, 3rd edn., John Wiley & Sons, New York, 2002.
- Converse, A. O., *Optimization*, Holt, Rinehart & Winston, New York, 1970.
- Coutinho, F. A. B., L. F. Lopez, M. N. Burattini, and E. Massad, Modeling the natural history of HIV infection in individuals and its epidemiological implications, *Bulletin of Mathematical Biology*, 63, 1041–1062, 2001.
- Culshaw, R. V. and S. Ruan, A delay differential equation model of HIV infection of CD4+ T cells, *Mathematical Biosciences*, 165, 27–39, 2000.
- Dabney, J. B. and T. L. Harman, *Mastering Simulink 4*, Prentice Hall, Upper Saddle River, NJ, 2001.
- Daniels, R. W., *An Introduction to Numerical Methods and Optimization Techniques*, Elsevier/North Holland, New York, 1978.
- D'azzo, J. J. and C. H. Houpis, *Linear Control System Analysis and Design*, 4th edn., McGraw-Hill, New York, 1995.
- Dorf, R. C. and R. H. Bishop, *Modern Control Systems*, 10th edn., Pearson/Prentice-Hall, Upper Saddle River, NJ, 2005.
- Edelstein-Keshet, L., *Mathematical Models in Biology*, McGraw-Hill, New York, 1988.
- Eguchi, H., K. Obana, and M. Kamiya, Hardware-in-the-loop missile simulation facility, *Proceedings of SPIE*, 3368, 2–9, 1998.
- Etkin, B., *Dynamics of Flight*, John Wiley & Sons, New York, 1982.
- Farlow, S. J., *An Introduction to Differential Equations and Their Applications*, McGraw-Hill, New York, 1994.
- Fausett, L. V., *Numerical Methods—Algorithms and Applications*, Prentice-Hall, Upper Saddle River, NJ, 2003.
- Fishwick, P. A., *Simulation Model Design and Execution—Building Digital Worlds*, Prentice-Hall, Upper Saddle River, NJ, 1995.

- Franklin, G. F., J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th edn., Prentice-Hall, Upper Saddle River, NJ, 2002.
- Gawthrop, P. and L. Smith, *METAMODELLING: Bond Graphs and Dynamic Systems*, Prentice-Hall, London, U.K., 1996.
- Gear, W. C., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1971.
- Gerald, J. Theusen and W.J. Fabrycky, Engineering Economy (9th Edition), Prentice-Hall, 2000.
- Gordon, G., *System Simulation*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
- Green, R. and K. Jackson, The design drive—Advanced HITL simulation systems for automotive controllers, *Modern Simulation and Training Journal*, 56–58, 1997.
- Haberman, R., *Mathematical Models—Mechanical Vibrations, Population Dynamics and Traffic Flow*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- Hannan, B. and R. Matthias, *Dynamic Modeling with STELLA II*, Springer-Verlag, New York, 1994.
- Hanselmann, H. and K. Smith, Real-time simulation replaces test drives, *Test & Measurement World Magazine*, 35–40, February 15, 1996.
- Haraldsdottir, A. and R. Howe, Multiple frame rate integration, *Flight Simulation Technologies Conference*, Atlanta, GA, September 7–9, 1988, Technical Paper (A88-53626 23-09), 1988.
- Hartley, T. T., *Digital Simulation of Dynamic Systems—A Control Theory Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
- Hasdorff, L., *Gradient Optimization and Nonlinear Control*, John Wiley & Sons, New York, 1976.
- Hay, J. L., R. E. Crosbie, and R. I. Chaplin, Integration routines for systems with discontinuities, *The Computer Journal*, 17, 275–279, 1973.
- Hethcote, H., Qualitative analyses of communicable disease models, *Mathematical Biosciences*, 28, 335–356, 1976.
- Hoffman, J. D., *Numerical Methods for Engineers and Scientists*, Marcel Dekker, New York, 1992.
- Hostetter, G. H., M. S. Santina, and Paul D'Carpio-Montalvo, *Analytical, Numerical and Computational Methods for Science and Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- Howe, R., Transfer function and characteristic root errors for fixed-step integration algorithms, *Transactions of SCS*, 2, 293–320, 1986.
- Howe, R., *Dynamics of Real-Time Digital Simulation*, Applied Dynamics International, Ann Arbor, MI, 1995.
- Hultquist, P. F., *Numerical Methods for Engineers and Computer Scientists*, Benjamin Cummings, Menlo Park, CA, 1988.
- Huntsinger, R., Personal notes.
- Hutton, D. V., *Fundamentals of Finite Element Analysis*, McGraw-Hill, New York, 2004.
- Isham, V., Mathematical modeling of the transmission dynamics of HIV infection and AIDS, *Journal of the Royal Statistical Society*, 151, 5–30, 1988.
- Jackson, L. B., *Signals, Systems and Transforms*, Addison-Wesley, Reading, MA, 1991.
- Jacquot, R., *Modern Digital Control Systems*, Marcel Dekker, New York, NY, 1981.
- Kailath, T., *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- Karayannakis, N., *Computer-Assisted Simulation of Dynamic Systems with Block Diagram Languages*, CRC Press, Boca Raton, FL, 1993.
- Karnopp, D. C., D. L. Margolis, and R. C. Rosenberg, *System Dynamics—Modeling and Simulation of Mechatronic Systems*, 4th edn., John Wiley & Sons, New York, 2000.
- Keen, R. E. and J. D. Spain, *Computer Simulation in Biology—A Basic Introduction*, John Wiley & Sons, New York, 1992.
- Kelton, W. D., R. P. Sadowski, and D. A. Sadowski, *Simulation with Arena*, McGraw-Hill, New York, 1997.
- Kermack, W. D. and A. D. McKendrick, A contribution to the mathematical theory of epidemics, *Proceedings of the Royal Society of London*, 115, 700–721, 1927.
- Korn, G. A. and J. V. Wait, *Digital Continuous-System Simulation*, Prentice-Hall, Englewood Cliffs, NJ, 1978.
- Kraniauskas, P., *Transforms in Signals and Systems*, Addison-Wesley, Wokingham, U.K., 1992.
- Kuo, B., *Digital Control Systems*, Holt, Rinehart & Winston, New York, 1980.
- Ledin, J., *Simulation Engineering*, CMP Books, Lawrence, KS, 2001.
- Linz, P. and R. L. C. Wang, *Exploring Numerical Methods—An Introduction to Scientific Computing Using MATLAB*, Jones and Bartlett, Boston, MA, 2003.
- Mathews, J. H. and K. D. Fink, *Numerical Methods Using MATLAB*, 3rd edn., Prentice-Hall, Upper Saddle River, NJ, 1999.
- McClamroch, N. H., *State Models of Dynamic Systems*, Springer-Verlag, New York, 1980.
- McLeod, J., PHYSBE ... A physiological simulation benchmark experiment, *SIMULATION*, 7, 324–329, 1966.

- Meerschaert, M. M., *Mathematical Modeling*, 2nd edn., Academic Press, San Diego, CA, 1999.
- Mesterton-Gibbons, M., *A Concrete Approach to Mathematical Modeling*, Addison-Wesley, Redwood City, CA, 1988.
- Miller, K. S., *Partial Differential Equations in Engineering Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- Miller, R. E., *Optimization Foundations and Applications*, John Wiley & Sons, New York, 2000.
- Mokhtari, M. and M. Marie, *Engineering Applications of MATLAB 5.3 and SIMULINK 3*, Springer-Verlag, London, U.K., 2000.
- Natke, H. G., *Introduction to Multi-Disciplinary Model-Building*, WIT Press, Southampton, U.K., 2003.
- Nekoogar, F. and G. Moriarty, *Digital Control Using Digital Signal Processing*, Prentice-Hall, Upper Saddle River, NJ, 1999.
- Nise, N. S., *Control Systems Engineering*, 2nd edn., Benjamin Cummings, Redwood City, CA, 1995.
- Ogata, K., *Discrete-Time Control Systems*, 2nd edn., Prentice-Hall, Englewood Cliffs, NJ, 1995.
- Ogata, K., *System Dynamics*, 3rd edn., Prentice-Hall, Upper Saddle River, NJ, 1998.
- Ogata, K., *Modern Control Engineering*, 4th edn., Prentice-Hall, 2002.
- O'Neil, P. V., *Advanced Engineering Mathematics*, Wadsworth, Belmont, CA, 1983.
- Oppenheim, A. V., R. W. Schafer, and R. J. Buck, *Discrete-Time Signal Processing*, 2nd edn., Prentice-Hall, Eaglewood Cliffs, NJ, 1999.
- Orfanidis, S., *Introduction to Signal Processing*, Prentice-Hall, Upper Saddle River, NJ, 1996.
- Palm, W. J., *Modeling, Analysis and Control of Dynamic Systems*, John Wiley & Sons, New York, 1983.
- Palusinski, O. A., Simulation of dynamic systems using multirate integration techniques, *Transactions of the Society for Computer Simulation*, 2, 257–273, 1986.
- Papoulis, A., *The Fourier Integral and Its Applications*, McGraw-Hill, New York, 1962.
- Parks, T. W. and C. S. Burrus, *Digital Filter Design* (Topics in Digital Signal Processing), John Wiley & Sons, New York, 1987.
- Perelson, A., Dynamics of HIV infection of CD4+ T cells, *Mathematical Biosciences*, 114, 81–125, 1993.
- Ralston, A. and H. S. Wilf, *Mathematical Methods for Digital Computers*, John Wiley & Sons, New York, 1965.
- Rao, S. S., *Applied Numerical Methods for Engineers and Scientists*, Prentice-Hall, Upper Saddle River, NJ, 2002.
- Recktenwald, G., *Numerical Methods with MATLAB—Implementation and Application*, Prentice-Hall, Upper Saddle River, NJ, 2000.
- Reseck, J., *SCUBA, Safe and Simple*, Simon and Schuster, New York, 1990.
- Richmond, B., *An Introduction to Systems Thinking: STELLA Software*, High Performance Systems Inc., Hanover, NH, 2001.
- Riggs, D. S., *Control Theory and Physiological Feedback Mechanisms*, The Williams & Wilkins Co., Baltimore, MD, 1970.
- Rohrs, C. E., J. L. Melsa, and D. G. Schultz, *Linear Control Systems*, McGraw-Hill, New York, 1993.
- Schilling, R. J. and S. L. Harris, *Applied Numerical Methods for Engineers Using MATLAB and C*, Brooks/Cole, Pacific Grove, CA, 2000.
- Shampine, L., *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994.
- Shearer, J. L., *Dynamic Modeling and Control of Engineering Systems*, Prentice-Hall, Upper Saddle River, NJ, 1997.
- Shevell, R. S., *Fundamentals of Flight*, 2nd edn., Prentice-Hall, Englewood Cliffs, NJ, 1989.
- Shier, D. R., *Applied Mathematical Modeling*, CRC Press, Boca Raton, FL, 2000.
- Smith, W. A., *Elementary Numerical Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1986.
- Smith, J. M., *Mathematical Modeling and Digital Simulation for Engineers and Scientists*, 2nd edn., John Wiley & Sons, New York, 1987.
- Speckhart, F. H., *A Guide to Using CSMP—The Continuous System Modeling Program*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
- Theusen, G. J. and W. J. Fabrycky, *Engineering Economy*, Prentice-Hall, 1971.
- Theusen, G. J. and W. J. Fabrycky, *Engineering Economy*, 9th edn., Prentice-Hall, Upper Saddle River, NJ, 2001.
- Tse, I. E., F. S. Hinkle, and R. T. Marse, *Mechanical Vibrations: Theory and Applications*, Allyn and Bacon, 1963.
- Wellstead, P. E., *Introduction to Physical System Modelling*, Academic Press, London, U.K., 1979.
- Wilde, D. J., *Optimum Seeking Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1964.
- Woods, R. L. and K. L. Lawrence, *Modeling and Simulation of Dynamic Systems*, Prentice-Hall, Upper Saddle River, NJ, 1997.



Taylor & Francis
Taylor & Francis Group
<http://taylorandfrancis.com>

Index

A

AB- m integrator, 515, 516
ACSL, *see* Advanced Continuous Simulation Language
Adams–Bashforth numerical integrators
 characteristic root error formula, 715
 method, 513–514
 stability boundaries, 717–720
 stability condition, 716
 undamped second-order system, 719–722
 z -domain transfer function, 714–715
Adams–Moulton implicit integrators, 519
 chemical concentration, 724–726
 stability boundaries, 723–724
 z -domain transfer functions, 723
Adaptive step size, 505
Adaptive techniques
 adaptive step size, 505
 constant step size, 505
 repeated Runge–Kutta with interval halving, 500–505
 Runge–Kutta–Fehlberg method, 505–510
Advanced Continuous Simulation Language (ACSL), 349
Advanced numerical integration
 continuous-time system models
 bilinear transform, 799–801
 first-order hold signal reconstruction, 796
 matched pole-zero method, 796–799
 sampling and signal reconstruction, 790–792
 dynamic errors
 asymptotic formulas, 704–708
 characteristic root errors, 687–688
 definition, 683
 discrete-time and equivalent continuous-time systems, 684–687
 linear system simulation, 708–711
 transfer function errors, 697–704
 types, 683–684
Lego Mindstorms™ NXT
 feedback control systems, 804
 filtered model, 810–815
 IDE, 805
 installation, 805–806
 mechatronics, 803
 noisy model, 806–810
 software and hardware requirements, 805
multirate integration
 aircraft pitch control system, 739
 airframe dynamics, 739
 analytical solution, state variables, 748–750
 frame ratio, 741
 master routine, 741
 nonlinear dual speed second-order system, 753–760
 simulation trade-offs, 763–764
 slave routine, 741
 slow and fast states procedure, 742–743
 slow and fast subsystem interaction, 741
 step size selection, 743–745

stiff system, 738
two-tank system, 760–763
real-time simulation
 extrapolation, 776–783
 high-fidelity-driving simulator, 769
 HIL, 767–768
 input delay, 783–786
 predictor–corrector method, 772–776
 RK-1 (explicit Euler), 770–771
 RK-2 (improved Euler), 771
 RK-2 (modified Euler), 771
 RK-3 (real-time compatible), 772
 RK-3 (real-time incompatible), 771–772
 RK-4 (real-time incompatible), 772
 two-pass numerical integration method, 769–770
 vehicle ABS system, 768
stability
 Adams–Bashforth numerical integrators, 714–720
 Adams–Moulton implicit integrators, 722–726
 Runge–Kutta (RK) integration, 726–736
Aircraft, longitudinal control, 319
 altitude control system, 330
 altitude from steady-state flight conditions, 328
 angle of attack and forces, 321
 body axis coordinates and Euler angles, 320
 elevator response for open-and closed-loop, 332
 linearized aircraft pitch response, 325
 open-and closed-loop altitude response vs. time, 329
 partial fraction expansion, 327
 primary control surface, 321
 short period and phugoid modes, 324–326
 transfer function, 323, 327, 330
Aircraft longitudinal dynamics, digital simulation of, 333–335
Aircraft pitch control system
 block diagram of, 738–739
 multiple integration of, 750–753
 simulation diagram for, 56, 744
Algebraic constraint blocks, 378
Algebraic equations, 375–378
 algebraic constraint blocks, 378
 first-order autonomous system, 376
Algebraic loops, 371–373
 circular nature, 372
 eliminating, 373–375
 equations, 375–378
 Memory block, 373–375
 submarine dynamics transfer function, 374
Algebraic manipulation, 373
AM- m integrator, 516, 732
Armature-controlled DC motor, 535
Asymptotic stability, 197
Autonomous nonlinear system, 80

B

Backlash, 72, 73
Backlash block, 389

Backward rectangular integration, 100
 BIBO, *see* Bounded input-bounded output
 Bode plot, 210
 closed-loop frequency response functions, 213
 for control system, 314
 of discrete-time systems, 287, 298
 for first-order system, 710
 of frequency response function, 284
 for marginally stable system, 219
 of open-loop transfer function, 217–219
 for second-order systems, 214–215
 third-order Butterworth low-pass filter, 211
 Bounded input-bounded output (BIBO), 197, 267–268, 271, 273–274

C

Car-following
 models, 380, 381, 384
 subsystem, 396–398
 Cascaded tanks with flow logic control, 121–126
 Centralized integration, 409–412
 Characteristic root errors, dynamic errors
 asymptotic formula, 689–690
 complex pole relationship, 691–692
 continuous and discrete-time unit step responses, 695–697
 damping ratio error, 693, 694
 equivalent system natural frequency, 695
 exact and asymptotic fractional errors, 692–693
 fractional error, 687–688
 impulse responses, 697
 responses, 690
 step response, 695–696
 trapezoidal integration, 690
 z -domain transfer function, 690
 Closed-loop depth rate control system, 363
 Closed-loop transfer function, 362
 Constant forces, physical properties and, 563–569
 Constant step size, 505
 Continuous-/discrete-time system
 conversion, 309–311
 poles, 276
 Continuous System Modeling Program (CSMP), 349
 Continuous-time first-order system
 discrete-time system approximation, 92
 exact and approximate solution, 95–96
 first-order, continuous-time systems, 92–93
 improved Euler integration, 727
 using trapezoidal integration, 288–293
 Continuous-time Kalman filter, 453–454
 Continuous-time system
 bilinear transform
 frequency response, 800–801
 mapping, 800
 prewarped transfer function, 801–802
 dynamic systems with, 349
 first-order continuous system, 92–93
 n distinct integrations, 91–92
 object's velocity, 115
 state derivative vector, 92
 first-order hold signal reconstruction, 796

first-order systems
 description, 29–30
 step response, 30–36
 higher-order systems
 aircraft pitch control system, 56
 feedback control system, 55
 railroad cars, 57, 65
 linear time invariant
 frequency response, 206–216
 stability, 194–206
 matched pole-zero method
 DC gains, 798
 frequency response, 798–799
 nonlinear systems
 applied force *vs.* time, 69–70
 backlash, 72, 73
 coulomb friction, 68
 dead zone, 71
 first-order systems, 66
 friction force *vs.* applied force, 68–69
 hysteresis, 72–74
 linear model approximation, 67
 mechanical system, 81
 progressive, 68
 quantization, 76–77
 saturation, 71–72
 sustained oscillations and limit cycles, 77–80
 temperature response, 74–76
 time constant, 75
 valve flow *vs.* current, 72
 with polynomial solutions, 488–490
 sampling and signal reconstruction
 continuous-time system response, 790–792
 frequency response function, 794–795
 illustration, 791
 piecewise constant function, 791
 transfer function, 791
 z -domain transfer function, 792
 second-order systems
 description, 36
 first-order equation conversion, 41–42
 mechanical system, 39–41
 two-tank mixing system, 42–44
 unit step response, 36–39
 simulation diagrams
 aircraft pitch control system, 56
 description, 45
 first-order system, 45–47
 heat flows and temperatures, two-room building, 51–52
 room temperature model, 51–52
 second-order system, 53–54, 58–59
 state variables
 dynamic system, 58, 61
 interacting tank system, 61–62
 linear state variable form conversion, 62–63
 spring-mass-damper system, 57
 state equations, 61–62
 transition matrix, 63
 submarine depth control system
 block diagram, 85
 controller and stern plane actuator, 89–90
 difference equations, 88

- discrete-time approximation, 89
 - simulation diagram, 86
 - state equations, 86–87
 - unit step response of, 277
 - Continuous-time system models, advanced numerical integration
 - bilinear transform, 799–801
 - first-order hold signal reconstruction, 796
 - matched pole-zero method, 796–799
 - sampling and signal reconstruction, 790–792
 - Continuous-time system simulation languages (CSSLs), 349
 - Control systems, 29
 - aircraft pitch, 739
 - components, 523
 - continuous-and discrete-time, 276
 - higher-order systems
 - aircraft pitch, 56
 - feedback, 55
 - Lego Mindstorms™ NXT, 804
 - linear, 216–219
 - Runge–Kutta (RK) method, *see* Runge–Kutta (RK) method
 - ship heading, *see* Ship heading control system
 - stiff systems, *see* Stiff systems
 - toolbox, 300
 - continuous-/discrete-time system conversion, 309–311
 - frequency response, 311–313
 - root locus, 313–316
 - state-space models, 302–303
 - state-space/transfer function conversion, 303–305
 - system interconnections, 305–307
 - system response, 307–309
 - transfer function models, 301–302
 - unity, 212
 - Coulomb friction, 68
 - CSMP, *see* Continuous System Modeling Program
 - CSSLs, *see* Continuous-time system simulation languages
 - Cylinder node temperatures, 550
- D**
- Data logging of scope signals, 355
 - Dead zone block, 387–389
 - Dead zone nonlinearity, 71
 - Decompression Sickness (DCS), 146
 - Digital control system, for chamber temperature, 432
 - Digital filters, 293–297, 412–414
 - Digital simulation, of aircraft longitudinal dynamics, 333–335
 - Discontinuity functions, 385–386, 559, 568
 - Discrete event models, 3
 - Discrete state equations, 116, 126
 - discrete step response of circuit, 120
 - lead-lag network, 117–118
 - linear state equations, 116–117
 - predator-prey ecosystem, 125–126
 - simulation diagram for *RC* lead-lag network, 119
 - steady-state response, 121–122
 - tank level responses, discrete and continuous, 123–125
 - using explicit Euler integration, 117, 121
 - Discrete-time frequency response function, 282
 - Discrete-time impulse function, 226–228
 - Discrete-time signal, 222–226
 - Discrete-time systems, 402–403
 - block diagram of, 275
 - centralized integration, 409–412
 - digital filters, 412–414
 - impulse responses for, 274
 - integrators, 406–409
 - Kalman filter, 454–455
 - mathematical modeling
 - exact vs. approximate solutions, 13–14
 - inherent, 19–22
 - liquid tank continuous-time system, 19
 - liquid tank discrete-time system, 19
 - step size, 16–18
 - matrices, 132–146
 - damped natural frequency, 140
 - discrete and continuous responses, 136–139
 - discrete system matrix, 139
 - explicit numerical integrators, 133
 - improved or modified Euler, 133
 - interacting tanks, 134–136
 - nonlinear pendulum with damping, 141–143
 - nonlinear second-order system, 141
 - quasi exact solution, 142
 - step responses of a second-order system, 140
 - transition matrix, 132
 - output, 260
 - simulation of inherently, 403–406
 - transfer function, 414–418
 - Distributed parameter systems, 546–550
 - Dynamic errors
 - asymptotic formulas
 - Euler integrator, 705–707
 - numerical integrators, 704–705
 - z*-domain transfer function, 704
 - characteristic root errors
 - asymptotic formula, 689–690
 - complex pole relationship, 691–692
 - continuous and discrete-time unit step responses, 695–697
 - damping ratio error, 693, 694
 - equivalent system natural frequency, 695
 - exact and asymptotic fractional errors, 692–693
 - fractional error, 687–688
 - impulse responses, 697
 - responses, 690
 - step response, 695–696
 - trapezoidal integration, 690
 - z*-domain transfer function, 690
 - definition, 683
 - discrete-time and equivalent continuous-time system
 - characteristic root, 688
 - continuous-time integrator, 686
 - step response, 685–686
 - linear system simulation
 - frequency response function, 708–709
 - RC circuit, 709–711
 - transfer function errors
 - continuous-and discrete-time integration, 702
 - explicit Euler and continuous-time integrator outputs, 703
 - fractional error, 697–699

- Dynamic errors (*Continued*)
 - frequency response functions, 697
 - phase angle plots, 701
 - time delay, 703–704
 - types, 683–684
- E**
- ECRobot NXT Blockset, 806, 807
- Elementary numerical integration, 91–92
 - discrete state equations, 116, 126
 - discrete step response of circuit, 120
 - lead-lag network, 117–118
 - linear state equations, 116–117
 - predator-prey ecosystem, 125–126
 - simulation diagram for RC lead-lag network, 119
 - steady-state response, 121–122
 - tank level responses, discrete and
 - continuous, 123–125
 - using explicit Euler integration, 117, 121
 - discrete-time system matrices, 132–146
 - damped natural frequency, 140
 - discrete and continuous responses, 136–139
 - discrete system matrix, 139
 - explicit numerical integrators, 133
 - improved or modified Euler, 133
 - interacting tanks, 134–136
 - nonlinear pendulum with damping, 141–143
 - nonlinear second-order system, 141
 - quasi exact solution, 142
 - step responses of a second-order system, 140
 - transition matrix, 132
 - discrete-time system, of continuous first-order system, 92–98
 - Euler integration, *see* Euler integration
 - improved Euler integration, 127–131
 - modified Euler integration, 131–132
 - nonlinear first-order systems, discrete approximation of, 112–117
 - trapezoidal integration, 104–111
 - area approximation, 104–105
 - continuous integrators, 105–106, 108
 - difference equation based on, 105, 107
 - discrete and continuous responses, 108–109, 110, 111
 - dynamics of sinking drum, 109–110
 - for first-order system, 106–107
 - integration step size, 104, 111
 - quadratic function, 108
 - vertical ascent of diver, 146–154
- Epidemic model
 - baseline conditions, 575–577
 - fatal disease, 573
 - immigration and inoculation profiles, 574–575
 - sensitivity analysis, 578–579
 - S-I-R models, 573
 - state transition diagram, 574
 - symptoms, 573
- Euler integration, 410, 539
 - area approximation, 98
 - comparison of explicit and implicit, 101
 - continuous-time signal, 103
 - discrete-time integrator, 102
 - explicit, *see* Explicit Euler integration
 - implicit, *see* Implicit Euler integration
 - improvements to
 - accuracy, 128–131
 - improved state estimate, 128
 - new state using forward Euler integration, 127–128
 - inherent weakness of, 127
 - modified, 131–132
 - RC circuit, 102
 - tank flow, 103
- Euler integrator (RK-1), 353, 479–481, 483, 486
- Explicit Euler integration, 99–100, 133, 242, 244, 248, 283, 334, 411
 - damped pendulum response using, 141–143
 - discrete state equations, 117, 121, 142
 - numerical integrator, 100
 - undamped pendulum response using, 144
- Explicit methods, 513–515, 519
- F**
- Fcn blocks, 398–401
- Feedback control system
 - block diagram, 200
 - characteristic polynomial, 201
 - closed-loop system
 - properties, 202
 - transfer function, 200
 - inverse Laplace transform, 202
 - ship heading response, 203–204
 - stability of linear, frequency response
 - block diagram, 217
 - Bode plot of open-loop transfer function, 217–218
 - closed-loop transfer function, 219
- First-order autonomous system, 376
- First-order differential equations, 490
- First-order discrete-time system, low-pass filter in, 262–265
- First-order systems
 - block diagram, 46
 - continuous-time models, 92–93
 - description, 29–30
 - difference equations, 10
 - exact vs. approximate solution, 13–15
 - LTI continuous-time systems, 208–210
 - nonlinear system, 66
 - simulation diagram
 - linear tank, 47
 - RC circuit, 47–48
 - step response of
 - graphs of, 30, 31
 - liquid storage tank model, 32, 34–35
 - RC circuit, 32–34
 - rule of thumb, 31
 - stiffness property in, 524–526
 - temperature-controlled chamber, 35–36
 - trapezoidal integration for, 106–107
- Fishery system dynamics
 - block diagram, 593
 - equilibrium states, 593–594
 - growth rate and equilibrium points, 592
 - Simulink® diagram, 591
 - state derivative function, 589–590
 - state responses, 592
- Forward rectangular integration, 100
- Fourier coefficients, 423, 424–426, 428

- Fourier Series expansion, 423–424
 Frequency response
 control system toolbox, 311–313
 function, 287, 540
 LTI continuous-time systems
 Bode plot for second-order systems, 214–215
 circuit with high-pass filter transfer function, 216
 closed-loop frequency response functions, 213
 first-order system, 208–210
 Fourier integral, 207
 linear feedback control systems, 216–219
 step responses for second-order systems, 215–216
 third-order Butterworth low-pass filter, 211
 unity feedback control system, 212
 LTI discrete-time systems, 280
 digital filters, 293–297
 properties of, 282–283
 sampling theorem, 287–288
 steady-state sinusoidal response, 280–282
 Friction, 386–387
- G**
- Global truncation error, 479
 Gradient search algorithm, 611–619
 Gradient vector, 605–607
 Graphical user interfaces (GUIs), 349
- H**
- Hardware-in-the-loop (HIL) simulation, 767–768
 Hemispherical tank-filling simulation
 gradient search algorithm, flow chart, 616
 objective function contours, 618–619
 objective function surface, 614–615, 627–628
 Simulink diagram, 615
 Heun's method, 128
 Higher-order systems, 490–496
 Higher-order systems, continuous-time system
 aircraft pitch control system, 56
 feedback control system, 55
 railroad cars, 57, 65
 High-order Runge–Kutta methods, 484–485
 HIL simulation, *see* Hardware-in-the-loop (HIL) simulation
 Human circulatory system, 395
 Hybrid systems, continuous-and discrete-time
 components, 431–433
 Hysteresis, 389–391
- I**
- IDE, *see* Integrated development environment
 Implicit Euler integration, 100–102, 133
 of continuous model, 115
 difference equation based on, 116
 numerical integrator, 100
 Implicit methods, multistep methods, 515–518
 Impulse response, LTI systems, 175–179
 spring-mass-damper system
 differential equation model of, 175–176
 Laplace transform, 177–178
 and transfer function, 179–182
 Impulse responses
 for discrete-time systems, 274
- function, 261–265
 graphs of, 274
 Inherently discrete-time system, 403–406
 Integrated development environment (IDE), 805
 Integration step, 475
 Intermediate numerical integration, 475
 adaptive techniques, 500
 adaptive step size, 505
 constant step size, 505
 repeated Runge–Kutta with interval halving, 500–505
 Runge–Kutta–Fehlberg method, 505–510
 epidemic model
 baseline conditions, 575–577
 fatal disease, 573
 immigration and inoculation profiles, 574–575
 sensitivity analysis, 578–579
 S-I-R models, 573
 state transition diagram, 574
 symptoms, 573
 lumped parameter approximation, 546–550
 nonlinear distributed parameter system, 550–555
 multistep methods, 512–513
 explicit methods, 513–515
 implicit methods, 515–518
 predictor–corrector methods, 518–522
 Runge–Kutta one-step methods, 475–476
 continuous-time models with polynomial solutions, 488–490
 higher-order systems, 490–496
 high-order Runge–Kutta methods, 484–485
 linear system models, 486–488
 second-order Runge–Kutta method, 477–479
 Taylor Series method, 476–477
 truncation errors, 479–484
 stiff systems, 523–524
 lower-order nonstiff system models, 529–542
 stiffness property in first-order system, 524–526
 stiff second-order system, 526–529
 systems with discontinuities, 555–563
 case study, 573–578
 physical properties and constant forces, 563–569
 Internal heat flows, 547
 Interval halving, repeated Runge–Kutta with, 500–505
 Inverse Laplace transform, 163–164
 Inverse z-transform, 232–233, 239–240
 Inverted pendulum, algebraic loop, 374
 Iodine distribution, human body
 block diagram, 185
 compartmental model for, 184
 state equations, 184–185
 state variable model, 190–192
 steady-state iodine levels, 186–187
 step response, 187–188
 transfer function, 185–187
- K**
- Kalman filtering, 453
 continuous-time, 453–454, 457
 discrete-time, 454–455
 Simulink simulations, *see* Simulink simulations
 steady-state, 454
 Kinetic friction, 386

L

- Laplace transform, 524–525
 inverse, 163–164
 one-sided, 155
 pairs for elementary continuous-time signals, 157
 partial fraction expansion, 166–172
 properties of, 156–163
 region of convergence, 156
 spring-mass-damper system, 175–177
 of system response, 164–166
- Lego Mindstorms™ NXT
 feedback control systems, 804
 filtered model
 block diagram, 811
 discrete-time Kalman filter subsystems, 811
 filtered data, 812, 814
 function-call subsystem, 810
 MATLAB plot, 814
 real-time workshop report, 809, 813
 signal generation, 811
- IDE, 805
 installation, 805–806
 mechatronics, 803
 noisy model
 block diagram, 806
 ECRobot NXT Blockset, 806
 function-call subsystem, 807
 MATLABO plot, 808
 noisy data, 808
 real-time workshop report, 809
 signal generation, 809
- software and hardware requirements, 805
- Linear discrete-time state equations, 256–261
- Linear second-order system, 491, 493–494
- Linear system analysis
 aircraft, longitudinal control of
 altitude control system, 330
 altitude from steady-state flight conditions, 328
 angle of attack and forces, 321
 body axis coordinates and Euler angles, 320
 elevator response for open-and closed-loop, 332
 linearized aircraft pitch response, 325
 open-and closed-loop altitude response vs.
 time, 329
 partial fraction expansion, 327
 primary control surface, 321
 short period and phugoid modes, 324–326
 transfer function, 323, 327, 330
- control system toolbox, 300
 continuous-/discrete-time system conversion,
 309–311
 frequency response, 311–313
 root locus, 313–316
 state-space models, 302–303
 state-space/transfer function conversion, 303–305
 system interconnections, 305–307
 system response, 307–309
 transfer function models, 301–302
- frequency response, LTI continuous-time systems
 Bode plot for second-order systems, 204–215
 circuit with high-pass filter transfer function, 216
 closed-loop frequency response functions, 213
 first-order system, 208–210
- Fourier integral, 207
 linear feedback control systems, 216–219
 step responses for second-order systems, 215–216
 third-order Butterworth low-pass filter, 211
 unity feedback control system, 212
- frequency response, LTI discrete-time systems
 digital filters, 293–297
 properties, 282–283
 sampling theorem, 287–288
 steady-state sinusoidal response, 280–282
- Laplace transform
 inverse, 163–164
 one-sided, 155
 pairs for elementary continuous-time signals, 157
 partial fraction expansion, 166–172
 properties of, 156–163
 region of convergence, 156
 of system response, 164–166
- models, 486–488
- notch filter for electrocardiograph waveform
 magnitude function, 339
 magnitude squared function, 339
 multinoth filters, 339–346
- stability
 LTI continuous-time system, 195–206
 LTI discrete-time system, 267–280
- transfer function
 impulse function, 173
 impulse response, 175–179
 and impulse response, relationship, 179–182
 multiple inputs and outputs, 182–189
 transformation from state variable model to,
 190–194
 unit step and unit impulse function, 173–175
- z*-domain transfer function
 approximating continuous-time system transfer
 functions, 245–247
 definition, 242
 Euler integration, 242–244
 linear discrete-time state equations, 256–261
 monetary fund, 257–258
 nonzero initial conditions, 243–244
 relationship of impulse response to, 264
 simulation diagrams and state variables, 250–256
 trapezoidal integration, 249–251
 weighting sequence (impulse response function),
 261–265
- z*-transform
 discrete-time impulse function, 226–228
 discrete-time signal, 222–226
 inverse, 232–233, 239–240
 Laplace and, 227
 partial fraction expansion, 233–234
 properties of, 229
 table for inverting, 236
- Linear systems simulation
 state-space block, 363–370
 Transfer Fcn block, 357–363
- Linear time invariant (LTI), continuous-time systems
 frequency response
 Bode plot for second-order systems, 214–215
 circuit with high-pass filter transfer
 function, 216
 closed-loop frequency response functions, 213

- first-order system, 208–210
- Fourier integral, 207
- linear feedback control systems, 216–219
- step responses for second-order systems, 215–216
- third-order Butterworth low-pass filter, 211
- unity feedback control system, 212
- stability
 - feedback control system, 200–206
 - polynomial characteristic, 195–200
- Linear time invariant (LTI), discrete-time systems
 - frequency response
 - digital filters, 293–297
 - properties, 282
 - sampling theorem, 287–288
 - steady-state sinusoidal response, 280–282
 - stability
 - BIBO, 267
 - complex poles of $H(z)$, 271–273
 - impulse response, 268
 - z -domain transfer function, 267–268
- Local truncation error, 479, 500–501, 515
- Logistic population growth model, 144
- Lookup Table block parameters, 383
- Lower-order dynamics model, 534
- Lower-order nonstiff system models
 - RK-4 integrator, 531–532
 - second-order system, 529, 532
 - sensor dynamics, 530
 - Simulink diagram, 531
 - step response, 531–532
 - step size vs. step number, 530
 - third-order system, 530
- Low-pass digital filters, 293–297
- Lumped parameter approximation, 546–550
 - nonlinear distributed parameter system, 550–555
- Lumped parameter system model, 2, 550
- LUNGS subsystem, 395

- M**
- Matched pole-zero method, 796–799
- Mathematical modeling
 - derivation, open tank
 - dynamic behavior, 4
 - flow between tanks, 8–9
 - fluid resistance, 7
 - volume, liquid flow, 5–6
 - difference equations, 10–12
 - discrete-time systems
 - exact vs. approximate solutions, 13–14
 - inherent, 19–22
 - liquid tank continuous-time system, 19
 - liquid tank discrete-time system, 19
 - step size, 16–18
 - dynamic systems, 555
 - lumped parameter model, 2
 - population dynamics
 - discrete-time model, 24
 - logistic growth population, 26–28
 - observed, discrete-time and continuous-time populations, 25
 - population data, 22, 23
 - simulation models, 3
 - stochastic models, 3
- MATLAB, 422–428, 436, 457
 - control system, 309
 - Fourier Series, 422–423
 - function, 531, 559
 - optimization toolbox, 599–600, 630
 - second-order system, 356, 426
 - truncated Fourier Series, 424–425
 - Workspace, 354, 356, 383
- Memory block, algebraic loops, 373–375
- MIMO, *see* Multiple input–multiple output
- Modified Euler integration, 131–132, 133–134
- Monte Carlo simulation, 435–439, 629
 - hospital occupancy, 623–624
 - mathematical model, 439–445
- Multinotch filters, 339–346
 - input and output of, 341, 342, 346
 - magnitude function, 340, 341, 344, 345
 - magnitude squared function, 339, 340
 - for removing fundamental frequency, 345
- Multiple input–multiple output (MIMO) system, 363
 - electric circuit, 182
 - iodine distribution, human
 - block diagram, 185
 - compartmental model for, 184
 - state equations, 184–185
 - state variable model, 190–192
 - steady-state iodine levels, 186–187
 - step response, 187–188
 - transfer function, 185–187
 - Multirate integration
 - aircraft pitch control system, 739
 - analytical, Simulink, and multirate responses, 751
 - Simulink and multirate integration, 751
 - airframe dynamics, 739
 - analytical solution, state variables
 - advantage, 748
 - total elevator deflection and its components, 750
 - total pitch response and its components, 749
 - frame ratio, 741
 - master routine, 741
 - nonlinear dual speed second-order system
 - air pressure, 754
 - coefficient matrix, 756
 - eigenvalues, 756–757
 - `linmod` function, 757–758
 - Simulink diagram, 758
 - two tank system, 753–754
 - procedure, slow and fast states, 742–743
 - simulation trade-offs
 - cpu time, 763–764
 - total execution time, 763
 - slave routine, 741
 - slow and fast subsystem interaction, 741
 - step size selection
 - dynamic accuracy, 745–748
 - stability, 743–745
 - stiff system, 738
 - two-tank system, 760–763
 - Multistep methods, 512–513
 - explicit methods, 513–515
 - implicit method, 515–518
 - predictor–corrector methods, 518–522

N

- Nonlinear algebraic equations, 377
 Nonlinear distributed parameter system, 550–555
 Nonlinear dual speed second-order system
 air pressure, 754
 coefficient matrix, 756
 eigenvalues, 756–757
 linmod function, 757–758
 Simulink diagram, 758
 two tank system, 753–754
 Nonlinear first-order systems, discrete approximation of, 112
 continuous model for sinking drum, 113–116
 exact solution for depth, 114
 implicit numerical integrators, 112–113
 object falling in a viscous medium, 116
 Nonlinear systems
 continuous-time systems
 applied force vs. time, 69–70
 backlash, 72, 73
 coulomb friction, 68
 dead zone, 71
 first-order systems, 66
 friction force vs. applied force, 68–69
 hysteresis, 72–74
 linear model approximation, 68
 mechanical system, 81
 progressive, 68
 quantization, 76–77
 saturation, 71–72
 sustained oscillations and limit cycles, 77–80
 temperature response, 74–76
 time constant, 75
 valve flow vs. current, 72
 Nonstiff control system models, step response, 533
 Notch filter, for electrocardiograph waveform, 338
 input and output of, 345
 magnitude function, 339, 344
 magnitude squared function, 339
 multinotch filters, 339–346
 input and output of, 341, 342, 346
 magnitude function, 340, 341, 344, 345
 magnitude squared function, 339, 340
 for removing fundamental frequency, 345
 square wave noise
 components of ECG signal, 342
 noise-corrupted ECG signal, 343
 Numerical integration methods, 520
 Nyquist frequency, 288

O

- One-sided Laplace transform, 155
 One-step methods, 475–476, 515
 Optimization, Simulink
 discrete-time system models, 620–625
 gradient vector, 605–607
 ground vehicle performance, 596
 MATLAB optimization, 599–600, 630
 minimum separation, 604
 multiparameter objective functions, 607–610
 optimum firing angle, 600–601
 parameter identification, 610–611
 projectile firing angle, 598–599

- separation distance vs. time, 603
 simple gradient search, 611–619
 target and projectile system, 597–598
 target speed sensitivity analysis, 602

P

- Parameter Estimation, 581
 Parenthesis, 547
 Partial differential equation models, 1–2
 Partial fraction expansion
 coefficients, 260
 Laplace transform
 complex roots, 169–172
 real and at least one multiple root, 167–169
 real and distinct roots, 166–167
 z-transform, 233–234
 Pendulum bob dynamics, 564, 565
 drag force, 566
 physical properties and constant forces, 563–569
 simulation of, 560
 velocity, 565, 566
 Periodic signals, 158
 PHYSBE, 395–396
 Physical models, 1
 Pilot ejection, 448–452
 diagram, 448
 Simulink diagram, 451
 trajectory of, 449
 Pitch control system transfer function, 746
 Plot of discontinuity functions, 567
 Polynomial characteristic
 asymptotic stability, 197
 bounded input-bounded output, 197
 higher order LTI system, 198
 MIMO systems, 198–199
 poles, natural modes, and stability, 197
 stability of second-order linear system, 196
 Population dynamics
 discrete-time model, 24
 logistic growth population, 26–28
 observed, discrete-time and continuous-time
 populations, 25
 population data, 22, 23
 Posteriori covariance subsystem, 466
 Posteriori state subsystem, 466
 Predator-prey
 ecosystem, 125–126
 model, 583–584, 595–596
 Predictor–corrector methods, 518–522
 Priori covariance subsystem, 465
 Priori state subsystem, 464
 Progressive nonlinearity, 68
 Public safety organizations, 1

Q

- Quadratic interpolation, 562
 Quantization block, 391–392, 391–394
 Quantization nonlinearity, 76–77

R

- Real-time HIL simulation, 767–768
 Real-time predictor–corrector method, 774–776

- Real-time simulation
 extrapolation
 fractional error, 779
 ideal extrapolator, 778–779
 linear, 778
 magnitude and phase plots, 780
 uses, 777
 input delay
 fraction gain error, 785
 phase angles, 786
 phase error, 785
 thermal system, 786–789
 uses, 783
 z -domain transfer function, 784
- Repeated Runge–Kutta with interval halving, 500–505
- Response Optimization, 581
- RK–Fehlberg method, 505–510
 boat crossing, 507–510
- RK-1 integrators, 479–481, 483, 486
- RK-2 integrators, 479–481, 483, 486, 528
- RK-3 integrators, 484–485, 486
- RK-4 integrators, 485, 486, 491, 505–506, 525–526, 567
- RK-5 integrators, 486, 505–506
- RK-6 integrators, 486
- RK method, *see* Runge–Kutta (RK) method
- Root locus, control system toolbox, 313–316
- Runge–Kutta integration, 358
- Runge–Kutta (RK) method
 characteristic root errors, 730–731
 modified Euler integration, 727–728
 one-step methods, 475–476
 continuous-time models with polynomial solutions, 488–490
 higher-order systems, 490–496
 high-order Runge–Kutta methods, 484–485
 linear system models, 486–488
 second-order Runge–Kutta method, 477–479
 Taylor Series method, 476–477
 truncation errors, 479–484
- polynomials, 730
- speed control system
 analytical and RK-2 simulation, 734
 analytical and RK-4 simulation, 735
 analytical step response and RK-3 simulated response, 736
 block diagram, 733
 RK-3 stability boundary, 734, 735
 Simulink diagram, 734
 z -domain transfer function, 729–730
- S**
- Sampled sinusoid, aliasing of, 288
- Sampling theorem, 287–288
- Saturation block, 387–389
- Saturation nonlinearity, 71–72
- Second-order continuous-time, P-I control of, 275
- Second-order RLC circuit, 527
- Second-order Runge–Kutta method, 477–479
- Second-order systems, 526–529, 555
 Adams–Bashforth numerical integrators, 720–722
 Bode plot, 214–215
 characteristic polynomial, 196
 description, 36
- first-order equation conversion, 41–42
- mechanical system
 block diagram, 39
 damping ratio and natural frequency, 39–40, 42
 position and velocity response, 41
 steady-state gain, 39, 42–43
 transient period, 40
- nonlinear dual speed
 air pressure, 754
 coefficient matrix, 756
 eigenvalues, 756–757
 linmod function, 757–758
 Simulink diagram, 758
 steady-state operating levels, 757
 two tank system, 753–754
- oscillatory step response, 38
- phase angle term, 37
- poles, natural modes, and stability, 197
- response, 355
- simulation diagrams, 53, 58–59
- step responses, 215–216, 352
- two-tank mixing system, 42–45
- unit step response, 36–39
- z -domain transfer function, 273–277
- Second-order truncated Taylor Series method, 479
- Ship heading control system
 block diagram, 608
 control parameters, 608
 feedback control system, 200–206
 objective function, 608–609
 optimal parameter settings, 611
 Simulink block diagram, 610
- Simulated response, 569
 using Euler integration, 539
- Simulation diagrams
 airframe dynamics, 739
 continuous-time systems
 aircraft pitch control system, 56
 description, 45
 first-order system, 45–48
 heat flows and temperatures, two-room building, 51–52
 room temperature model, 51–52
 second-order system, 48–49, 53–54, 58–59
- fast subsystem, 741
- n th-order continuous-time system, 252, 253
- for RC lead-lag network, 119
- second-order system
 trapezoidal integration, 251
 state variables and, 250–256
- third-order system, 180
- Simulation models, 3
- Simulation tools
 iterative procedure, 582
 optimization, Simulink
 discrete-time system models, 620–625
 gradient vector, 605–607
 ground vehicle performance, 596
 MATLAB optimization toolbox, 599–600, 630
 minimum separation, 604
 multiparameter objective functions, 607–610
 optimum firing angle, 600–601, 625–627
 parameter identification, 610–611
 projectile firing angle, 598–599

- Simulation tools (*Continued*)
 - separation distance vs. time, 603
 - simple gradient search, 611–619
 - target and projectile system, 597–598
 - target speed sensitivity analysis, 602
 - steady-state solver
 - equilibrium point, nonautonomous system, 586–589
 - nonlinear state model, 582
 - predator-prey model, 583–584
 - trim function, 584–586
 - Simulink, 349
 - algebraic loops, *see* Algebraic loops
 - blocks, 380–385
 - acceleration response, 381
 - backlash, 389
 - car-following models, 380, 381–382, 384
 - dead zone and saturation, 387–389
 - discontinuities, 385–386
 - friction, 386–387
 - hysteresis, 389–391
 - lead and following vehicles, 380, 384–385
 - Lookup Table block parameters, 383
 - quantization, 391–392, 391–394
 - continuous-and discrete-time components, 431–433
 - diagram, 508
 - arrow and target simulation, 441
 - capacitive transducer, 588
 - car-following system, 383, 397, 398
 - cascaded tanks, 471
 - closed-loop depth rate control system, 363
 - continuous-time Kalman filter, 457
 - digital control system for chamber temperature, 432
 - explicit Euler integration, 411
 - first-and second-order models, 532
 - fishery system dynamics, 591
 - hemispherical tank-filling simulation, 615
 - hospital occupancy, 621
 - inverted pendulum, 400, 644
 - loan repayment, 405
 - low-pass filters, 417
 - lumped parameter system model, 550
 - nonlinear two-tank system, 652
 - nonlinear vs. linearized models, 646
 - notch filter, 413
 - pendulum dynamics, 564
 - PHYSBE model, 396
 - pilot ejection, 451
 - Relay block for thermostat, 391
 - second-order system, 357, 410, 532
 - ship heading step response, 610
 - simulating stiff control system dynamics, 531
 - for simulation of nonlinear and linearized system, 636
 - solving algebraic equations, 376
 - sub depth control, 358
 - submarine depth rate., 361
 - third-order control systems, 532
 - truncated Fourier Series, 424
 - vehicle response traveling, 367
 - vehicle rolling down incline, 408
 - discrete-time systems, 402–403
 - centralized integration, 409–412
 - digital filters, 412–414
 - integrators, 406–409
 - simulation of inherently, 403–406
 - transfer function, 414–418
 - interface, 422–428
 - Kalman filtering, 453
 - continuous-time, 453–454
 - discrete-time, 454–455
 - Simulink simulations, *see* Simulink simulations
 - steady-state, 454
 - MATLAB, *see* MATLAB
 - model, 349, 353–355, 357
 - data logging of scope signals, 355
 - dialog box for configuring, 353
 - Euler integrator, 353
 - inverted pendulum with "Memory" block, 374
 - for RLC circuit, 528
 - running Simulink, 353–355
 - scope output, 354
 - screen capture, 354
 - second-order system response, 355
 - simulating coffee pot, 554
 - Simulink library, 349–353
 - model optimization
 - discrete-time system models, 620–625
 - gradient vector, 605–607
 - ground vehicle performance, 596
 - MATLAB optimization toolbox, 599–600
 - minimum separation, 604
 - multiparameter objective functions, 607–610
 - optimum firing angle, 600–601, 625–627
 - parameter identification, 610–611
 - projectile firing angle, 598–599
 - separation distance vs. time, 603
 - simple gradient search, 611–619
 - target and projectile system, 597–598
 - target speed sensitivity analysis, 602
 - Monte Carlo simulation, 435–439
 - mathematical model, 439–445
 - pilot ejection, 448–452
 - simulation of linear systems, 357
 - state-space block, 363–370
 - Transfer Fcn block, 357–363
 - subsystems, 394–395
 - car-following, 396–398
 - Fcn block, 398–401
 - PHYSBE, 395–396
 - Simulink library
 - blocks, 349–350
 - Browser, 350, 385
 - Discontinuities, 387
 - second-order system step response, 352
 - step response of second-order system, 352
 - Simulink optimization, hospital-patient occupancy
 - block diagram, 621
 - daily arrivals and departures, 620
 - daily net patient input, 621, 623
 - input and output relationship, 620
 - Monte Carlo simulation, 623–624, 629
 - objective function, 624
 - patient profiles, 621–622
 - Simulink simulations, 455–468
 - actual subsystem, 456
 - continuous-time Kalman filter, 456–457
 - discrete-time Kalman filter, 462, 464

- Kalman gain subsystem, 465
plot of
 acceleration, 459, 462, 468
 range, 458
 range error vs. time, 459, 463, 468
 range estimates, 461, 467
 velocity, 458, 462, 467
 velocity error vs. time., 460, 463, 469
posteriori covariance subsystem, 466
posteriori state subsystem, 466
prior covariance subsystem, 465
prior state subsystem, 464
 steady-state Kalman filter algorithm, 461
Simulink's stiff integrators, 526
Single input-single output (SISO), 363
Spring-mass-damper system, 57
 differential equation model of, 175–176
 impulse response, 177–178
Stability, linear time invariant
 continuous-time system
 characteristic polynomial, 195–200
 feedback control system, 200–206
 discrete-time systems
 BIBO, 267
 complex poles of $H(z)$, 271–273
 impulse response, 268
 z-domain transfer function, 267–268
 linear feedback control systems, 216–219
State derivative function, 475
State-space block, 363–370
 moving vehicle and suspension system model, 365
 vehicle cab displacement, 368
State-space models, 302–303
State variable model, simulation of, 335–337
State variables, simulation diagrams and, 250–256
Steady-state Kalman filter, 454
Steady-state solver
 equilibrium point, nonautonomous system, 586–589
 nonlinear state model, 582
 predator-prey model, 583–584
 trim function, 584–586
Step response of second-order system, 352
Stiff control system models, step response, 533
Stiff integrators, 529
Stiffness property in first-order system, 524–526
Stiff second-order system, 526–529
Stiff systems, 523–524
 lower-order nonstiff system models, 529–542
 stiffness property in first-order system, 524–526
 stiff second-order system, 526–529
Stochastic models, 3
Submarine depth control system, 358
 block diagram, 85
 closed-loop transfer function, 362
 controller and stern plane actuator, 89–90
 difference equations, 88
 discrete-time approximation, 89
 simulation diagram, 86
 state equations, 86–87
 state-space models, 303–305
Submarine dynamics transfer function, 374
Subsystems, 394–395
 car-following, 396–398
 Fc_n block, 398–401
 PHYSBE, 395–396
 Tire Model, 395
 vehicle dynamics model, 395
System interconnections, 305–307
System response, 307–309
Systems with discontinuities, 555–563
 case study, 573–578
 physical properties and constant forces, 563–569
- T**
- Taylor Series method, 476–477, 479, 480, 483, 488–490
Tire Model, 395
Transfer Fcn block, 373
 command and actual submarine depth rates, 359
 second-order system, 357, 358
 submarine depth rate control system, 358, 360
Transfer function, 414–418
 conversion, 303–305
 errors
 continuous-and discrete-time integration, 702
 explicit Euler and continuous-time integrator outputs, 703
 fractional error, 697–699
 frequency response functions, 697
 phase angle plots, 701
 time delay, 703–704
 of linear systems analysis
 impulse function, 173
 impulse response, 175–179
 and impulse response, relationship, 179–182
 multiple inputs and outputs, 182–189
 transformation from state variable model to, 190–194
 unit step and unit impulse function, 173–175
models, 301
Trapezoidal integration, 104–111, 249–251, 254, 255
area approximation, 104–105
continuous-and discrete-time, 251
continuous integrators, 105–106, 108
continuous-time first-order system in, 288–293
difference equation based on, 105, 107
discrete and continuous responses, 108–109, 110, 111
discrete integrators, 105–106, 108
dynamics of sinking drum, 109–110
for first-order system, 106–107
integration step size, 104, 111
of nonlinear time-varying system, 107–108
of second-order system, 255
state equations for, 255
of underdamped second-order system, 256
Trim function, 584–586
Truncated Fourier Series, 424–425
Twente University of Technology Simulator (TUTW), 349
- U**
- Undamped pendulum response, using explicit Euler, 144
Unit impulse function, 173–175
Unit step function, 173–175
Unit step responses
 first-and second-order system models, 538
 unstable second-order model, 539
 weighting sequences and, 264

V

Variable capacitance transducer
 circuit diagram, 586
 dynamic system with equilibrium conditions, 589
 mathematical model, 586–587
 Simulink diagram, 588

Vehicle dynamics model, 767–768

Vehicle response traveling, 367

Vertical ascent of diver, 146
 air, 146
 cable forces, 146
 discrete differential pressure responses, 152
 discrete-time system
 equilibrium state, 149
 outputs, 148, 151
 state equation matrices, 148
 state variables, 150
 diver's internal body pressure, 146
 drag force, 146
 dynamic system, 146–147
 initial cable force, 148–149
 maximum cable force, 152–153
 net cable force, 147
 second-order differential equation, 147, 153
 third order linear dynamic system, 147

W

Weighting sequence (impulse response function), 261–265

Z

z -domain transfer function
 approximating continuous-time system transfer functions, 245–247
 definition, 242
 Euler integration, 242–244
 linear discrete-time state equations, 256–261
 monetary fund, 257–258
 nonzero initial conditions, 243–244
 relationship of impulse response to, 264
 simulation diagrams and state variables, 250–256
 trapezoidal integration, 249–251
 weighting sequence (impulse response function), 261–265

z -transform
 discrete-time impulse function, 226–228
 discrete-time signal, 222–226
 inverse, 232–233, 239–240
 Laplace and, 227
 partial fraction expansion, 233–234
 properties of, 229
 table for inverting, 236