SYSTEM DYNAMICS

Fourth edition

William J. Palm III

Mc Graw Hill

System Dynamics

Fourth Edition

William J. Palm III
University of Rhode Island

Mc Graw Hill

SYSTEM DYNAMICS, FOURTH EDITION

This book is printed on acid-free paper.

1 2 3 4 5 6 7 8 9 LWI 24 23 22 21 20

ISBN 978-0-07-814005-1 (bound edition)
MHID 0-07-814005-6 (bound edition)
ISBN 978-1-260-44398-1 (loose-leaf edition)
MHID 1-260-44398-1 (loose-leaf edition)

Product Developers: *Tina Bower & Megan Platt*
Marketing Manager: *Shannon O'Donnell*
Content Project Managers: *Laura Bies, Samantha Donisi-Hamm, and Sandra Schnee*
Buyer: *Sandy Ludovissy*
Design: *Matt Diamond*
Content Licensing Specialist: *Abbey Jones*
Cover Image: ©*Chesky/Shutterstock*
Compositor: *Aptara®, Inc*

All credits appearing on page or at the end of the book are considered to be an extension of the copyright page.

Library of Congress Cataloging-in-Publication Data

Names: Palm, William J., III (William John), 1944- author.
Title: System dynamics / William J. Palm, III, University of Rhode Island.
Description: Fourth edition. | Dubuque : McGraw-Hill Education, 2020. | Includes index.
Identifiers: LCCN 2019029119 (print) | LCCN 2019029120 (ebook) | ISBN 9780078140051 (hardcover) | ISBN 9781260443981 (spiral bound) | ISBN 9781260443967 (ebook) | ISBN 9781260443943 (ebook other)
Subjects: LCSH: Automatic control–Mathematical models. | Dynamics–Mathematical models. | System analysis.
Classification: LCC TJ213 .P228 2020 (print) | LCC TJ213 (ebook) | DDC 620.1/04015118–dc23
LC record available at https://lccn.loc.gov/2019029119
LC ebook record available at https://lccn.loc.gov/2019029120

mheducation.com/highered

To my wife, Mary Louise; and to my children, Aileene, Bill, and Andrew.

CONTENTS

Preface vii

About the Author xii

CHAPTER 1
Introduction 1

1.1 Introduction to System Dynamics 2
1.2 Units 9
1.3 Developing Linear Models 11
1.4 Introduction to Differential Equations 17
1.5 A Case Study in Motion Control 21
1.6 MATLAB Review 28
1.7 Chapter Review 34
Problems 35

CHAPTER 2
Dynamic Response Methods 40

2.1 Solving Differential Equations 41
2.2 Response Parameters and Stability 59
2.3 The Laplace Transform Method 72
2.4 Solving Equations with the Laplace Transform 80
2.5 Transfer Functions 88
2.6 Pulse and Impulse Inputs 92
2.7 Partial-Fraction Expansion 100
2.8 Laplace Transforms and MATLAB 109
2.9 Transfer-Function Analysis in MATLAB 111
2.10 Chapter Review 118
References 119
Problems 120

CHAPTER 3
Modeling of Rigid-Body Mechanical Systems 129

3.1 Translational Motion 130
3.2 Rotation About a Fixed Axis 135
3.3 Equivalent Mass and Inertia 143
3.4 General Planar Motion 150
3.5 Additional Examples 156
3.6 A Case Study in Motion Control 165
3.7 Chapter Review 170
Reference 171
Problems 171

CHAPTER 4
Spring and Damper Elements in Mechanical Systems 184

4.1 Spring Elements 185
4.2 Modeling Mass-Spring Systems 196
4.3 Energy Methods 207
4.4 Damping Elements 217
4.5 Additional Modeling Examples 228
4.6 Collisions and Impulse Response 239
4.7 MATLAB Applications 243
4.8 Case Study: Vehicle Suspension Design 247
4.9 Chapter Review 254
References 254
Problems 254

CHAPTER 5
Block Diagrams, State-Variable Models, and Simulation Methods 274

Part I. Model Forms 275
5.1 Transfer Functions and Block Diagram Models 275
5.2 State-Variable Models 284
Part II. MATLAB Methods 295
5.3 State-Variable Methods with MATLAB 295
5.4 The MATLAB `ode` Functions 303
Part III. Simulink Methods 315
5.5 Simulink and Linear Models 315
5.6 Simulink and Nonlinear Models 321

5.7 Case Study: Vehicle Suspension Simulation 328
5.8 Chapter Review 332
References 332
Problems 333

CHAPTER 6
Electrical and Electromechanical Systems 343

6.1 Electrical Elements 344
6.2 Circuit Examples 350
6.3 Transfer Functions and Impedance 359
6.4 Operational Amplifiers 368
6.5 Electric Motors 373
6.6 Analysis of Motor Performance 382
6.7 Case Study: Design of a Motion-Control System 386
6.8 Sensors and Electroacoustic Devices 394
6.9 MATLAB Applications 398
6.10 Simulink Applications 407
6.11 Chapter Review 410
Problems 411

CHAPTER 7
Fluid and Thermal Systems 421

Part I. Fluid Systems 422
7.1 Conservation of Mass 422
7.2 Fluid Capacitance 424
7.3 Fluid Resistance 428
7.4 Dynamic Models of Hydraulic Systems 432
7.5 Pneumatic Systems 445
Part II. Thermal Systems 448
7.6 Thermal Capacitance 449
7.7 Thermal Resistance 450
7.8 Dynamic Models of Thermal Systems 459
Part III. MATLAB and Simulink Applications 467
7.9 MATLAB Applications 467
7.10 Simulink Applications 471
7.11 Chapter Review 476
Reference 476
Problems 476

CHAPTER 8
System Analysis in the Time Domain 491

8.1 Response of First-Order Systems 492
8.2 Response of Second-Order Systems 501
8.3 Description and Specification of Step Response 512
8.4 Parameter Estimation in the Time Domain 520
8.5 MATLAB Applications 529
8.6 Simulink Applications 531
8.7 Chapter Review 533
Problems 533

CHAPTER 9
System Analysis in the Frequency Domain 542

9.1 Frequency Response of First-Order Systems 543
9.2 Frequency Response of Higher-Order Systems 556
9.3 Frequency Response Applications 568
9.4 Filtering Properties of Dynamic Systems 581
9.5 Response to General Periodic Inputs 589
9.6 System Identification from Frequency Response 592
9.7 Case Study: Vehicle Suspension Design 597
9.8 Frequency Response Analysis Using MATLAB 603
9.9 Chapter Review 606
Problems 607

CHAPTER 10
Introduction to Feedback Control Systems 618

10.1 Closed-Loop Control 619
10.2 Control System Terminology 625
10.3 Modeling Control Systems 626
10.4 The PID Control Algorithm 640

10.5 Control System Analysis 648
10.6 Controlling First-Order Plants 653
10.7 Controlling Second-Order Plants 662
10.8 Additional Examples 670
10.9 Case Study: Motion Control with Feedback 684
10.10 Simulink Applications 691
10.11 Chapter Review 694
Reference 695
Problems 695

CHAPTER **11**
Control System Design and the Root Locus Plot 711

11.1 Root Locus Plots 712
11.2 Design Using the Root Locus Plot 717
11.3 Tuning Controllers 747
11.4 Saturation and Reset Windup 753
11.5 State-Variable Feedback 760
11.6 MATLAB Applications 769
11.7 Simulink Applications 777
11.8 Chapter Review 778
References 779
Problems 779

CHAPTER **12**
Compensator Design 796

12.1 Series Compensation 797
12.2 Design Using the Bode Plot 816
12.3 MATLAB Applications 833
12.4 Simulink Applications 838
12.5 Chapter Review 839
Problems 840

CHAPTER **13**
Vibration Applications (on the text website)

APPENDICES
A. Guide to Selected MATLAB Commands and Functions 849
B. Fourier Series 856
C. Developing Models from Data 858
D. Introduction to MATLAB (on the text website)
E. Numerical Methods (on the text website)

Answers to Selected Problems 885

Glossary 893

Index 896

PREFACE

System dynamics deals with mathematical modeling and analysis of devices and processes for the purpose of understanding their time-dependent behavior. While other subjects, such as Newtonian dynamics and electrical circuit theory, also deal with time-dependent behavior, system dynamics emphasizes methods for handling applications containing multiple types of components and processes such as electromechanical devices, electrohydraulic devices, and fluid-thermal processes. Because the goal of system dynamics is to understand the time-dependent behavior of a system of interconnected devices and processes as a whole, the modeling and analysis methods used in system dynamics must be properly selected to reveal how the connections between the system elements affect its overall behavior. Because systems of interconnected elements often require a control system to work properly, control system design is a major application area in system dynamics.

TEXT PHILOSOPHY

This text is an introduction to system dynamics and is suitable for such courses commonly found in engineering curricula. It is assumed that the student has a background in elementary differential and integral calculus and college physics (dynamics, mechanics of materials, thermodynamics, and electrical circuits). Previous exposure to differential equations is desirable but not necessary, as the required material on differential equations, as well as Laplace transforms and matrices, is developed in the text.

The decision to write a textbook often comes from the author's desire to improve on available texts. The decisions as to what topics to include and what approach to take emerge from the author's teaching experiences that give insight as to what is needed for students to master the subject. This text is based on the author's forty-four years of experience in teaching system dynamics.

This experience shows that typical students in a system dynamics course are not yet comfortable with applying the relevant concepts from earlier courses in dynamics and differential equations. Therefore, this text reviews and reinforces these important topics early on. Students often lack sufficient physical insight to relate the mathematical results to applications. The text therefore uses everyday illustrations of system dynamics to help students to understand the material and its relevance.

If laboratory sessions accompany the system dynamics course, many of the text's examples can be used as the basis for experiments. The text is also a suitable reference on hardware and on parameter estimation methods.

MATLAB® AND SIMULINK®[1]

MATLAB and Simulink are used to illustrate how modern computer tools can be applied in system dynamics.[2] MATLAB was chosen because it is the most widely

[1]MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

[2]The programs in this text are based on the following software versions, or higher versions: Version 9.6 of MATLAB, Version 9.3 of Simulink, and Version 10.6 of the Control Systems Toolbox.

used program in system dynamics courses and by practitioners in the field. Simulink, which is based on MATLAB and uses a diagram-based interface, is increasing in popularity because of its power and ease of use. In fact, students convinced the author to use Simulink after they discovered it on their own and learned how easy it is to use! It provides a useful and motivational tool.

It is, however, not necessary to cover MATLAB or Simulink in order to use the text, and it is shown how to do this later in the Preface.

TEXT OVERVIEW

Chapter 1 introduces the basic terminology of system dynamics, covers commonly used functions, and reviews the two systems of units used in the text: British Engineering (FPS) units and SI units. These are the unit systems most commonly used in system dynamics applications. The examples and homework problems employ both sets of units so that the student will become comfortable with both. Chapter 1 also covers some basic methods for parameter estimation. These methods are particularly useful for obtaining numerical values of spring constants, damping coefficients, and other parameters commonly found in system dynamics models. The chapter also contains introductions to differential equations and to MATLAB, and it presents the first of the text's several case studies: design of motion-control systems. The material on function identification and least-squares fitting, formerly in Chapter 1 in the third edition, is now in Appendix C.

Chapter 2 covers differential equations in more depth, and develops the Laplace transform method for solving differential equations with applications to equations having step, ramp, sine, impulse, and other types of forcing functions. It also introduces transfer function models.

Chapter 3 covers rigid-body dynamics, including planar motion. This chapter continues the motion-control case study by showing how to select a suitable motor and gear system.

Chapter 4 covers modeling of mechanical systems having stiffness and damping, and it applies the analytical methods developed in Chapter 2 to solve the models. This chapter then introduces the second case study: design of vehicle suspensions.

Chapter 5 develops block diagrams and the state-variable model, which is useful for certain analytical techniques as well as for numerical solutions. The optional sections of this chapter introduce Simulink, which is based on block-diagram descriptions, and apply the chapter's concepts using MATLAB. This chapter concludes with further coverage of the vehicle suspension case study.

Chapter 6 treats modeling of electric circuits, operational amplifiers, electromechanical devices, sensors, and electroacoustic devices. It also discusses how motor parameters can be obtained, and it returns to the motion-control case study and shows how to analyze motor and amplifier performance.

Part I of Chapter 7 covers fluid systems. Part II covers thermal systems. These two parts are independent of each other. A background in fluid mechanics or heat transfer is not required to understand this chapter, but students should have had elementary thermodynamics before covering the material on pneumatic systems in Section 7.5.

Chapters 8 and 9 cover analysis methods in the time domain and the frequency domain, respectively.

Chapter 8 integrates the modeling and analysis techniques of earlier chapters with an emphasis on understanding system behavior in the time domain, using step, ramp, and impulse functions primarily. The chapter covers step-response specifications such as maximum overshoot, peak time, delay time, rise time, and settling time.

Chapter 9 demonstrates the usefulness of the transfer function for understanding and analyzing a system's frequency response. It introduces Bode plots and shows how they are sketched and interpreted to obtain information about time constants, resonant frequencies, and bandwidth. The chapter returns to the vehicle-suspension case study, and shows how to use frequency response methods to evaluate suspension performance.

Chapters 10, 11, and 12 deal with a major application of system dynamics, namely, control systems. Chapter 10 is an introduction to feedback control systems, including the PID control algorithm applied to first- and second-order plants. The chapter concludes with thorough coverage of feedback control applied to the motion-control case study.

Chapter 11 deals with control systems in more depth and includes design methods based on the root locus plot and practical topics such as compensation, controller tuning, actuator saturation, reset windup, and state-variable feedback, with emphasis on motion-control systems. Chapter 12 covers series compensation methods and design with the root locus plot and the Bode plot.

Chapter 13 covers another major application area, vibrations. Important practical applications covered are vibration isolators, vibration absorbers, modes, and suspension system design. This chapter is now on the text website to allow room for the new case studies in earlier chapters.

ALTERNATIVE COURSES IN SYSTEM DYNAMICS

The choice of topics depends partly on the desired course emphasis, partly on the students' background in differential equations and dynamics, and partly on whether the course is a quarter or semester course.

Fluid and thermal systems are covered in Chapter 7, which has been shortened in this edition. Some students may have had courses in fluid mechanics and heat transfer, but probably have not been exposed to the system dynamics viewpoint, which focuses on the analogies between fluid and thermal resistance and capacitance and the corresponding electrical concepts. The theory and methods of the remaining chapters do not depend on Chapter 7, but some examples do.

In the author's opinion, a basic semester course in system dynamics should include most of the material in Chapters 1 through 7, and Chapters 9 and 10. At the author's institution, the system dynamics course is a junior course required for mechanical engineering majors, who have already had courses in dynamics and differential equations. It covers Chapters 1 through 10, with brief coverage of Chapter 7 and Chapter 8, and with some MATLAB and Simulink sections omitted. This optional material is then covered in a senior elective course in control systems, which also covers Simulink, and Chapters 11 and 12.

The text is flexible enough to support a variety of courses. The sections dealing with MATLAB and Simulink are at the end of the chapters and may be omitted. If students are familiar with Laplace transform methods and linear differential equations, Chapter 2 may be covered quickly. If students are comfortable with rigid-body planar motion, Chapter 3 may be used for a quick review.

GLOSSARY AND APPENDICES

There is a glossary containing the definitions of important terms, five appendices, and an index. Appendices D and E are on the text website.

Appendix A is a collection of tables of MATLAB commands and functions, organized by category. The purpose of each command and function is briefly described in the tables.

Appendix B is a brief summary of the Fourier series, which is used to represent a periodic function as a series consisting of a constant plus a sum of sine terms and cosine terms. It provides the background for some applications of the material in Chapter 9.

Appendix C covers function identification, and shows how to use MATLAB to fit models to scattered data using the least-squares method.

Appendix D is a self-contained introduction to MATLAB, and it should be read first by anyone unfamiliar with MATLAB if they intend to cover the MATLAB and Simulink sections. It also provides a useful review for those students having prior experience with MATLAB.

Appendix E covers numerical methods, such as the Runge-Kutta algorithms, that form the basis for the differential equation solvers of MATLAB. It is not necessary to master this material to use the MATLAB solvers, but the appendix provides a background for the interested reader.

Answers to selected homework problems are given following Appendix C.

CHAPTER FORMAT

The format of each chapter follows the same pattern, which is

1. Chapter outline
2. Chapter objectives
3. Chapter sections
4. MATLAB sections (in most chapters)
5. Simulink section (in most chapters)
6. Chapter review
7. References
8. Problems

This structure has been designed partly to accommodate those courses that do not cover MATLAB and/or Simulink, by placing the optional MATLAB and Simulink material at the end of the chapter. Chapter problems are arranged according to the chapter section whose concepts they illustrate. All problems requiring MATLAB and/or Simulink have thus been placed in separate, identifiable groups.

OPTIONAL TOPICS

In addition to the optional chapters (11, 12, and 13), some chapters have sections dealing with material other than MATLAB and Simulink that can be omitted without affecting understanding of the core material in subsequent chapters. All such optional material has been placed in sections near the end of the chapter. This optional material includes:

1. Function discovery, parameter estimation, and system identification techniques (Sections 8.4 and 9.6)
2. General theory of partial-fraction expansion (Section 2.7)
3. Impulse response (Sections 2.6 and 4.6)
4. Motor performance (Section 6.6)
5. Sensors and electroacoustic devices (Section 6.8)

DISTINGUISHING FEATURES

The following are considered to be the major distinguishing features of the text.

1. **MATLAB.** Standalone sections in most chapters provide concise summaries and illustrations of MATLAB features relevant to the chapter's topics.
2. **Simulink.** Standalone sections in Chapters 5 through 12 provide extensive Simulink coverage not found in most system dynamics texts.
3. **Parameter estimation.** Coverage of function discovery, parameter estimation, and system identification techniques is given in Sections 1.3, 8.4, 9.6, and Appendix C. Students are uneasy when they are given parameter values such as spring stiffness and damping coefficients in examples and homework problems, because they want to know how they will obtain such values in practice. These sections show how this is done.
4. **Motor performance evaluation.** Section 6.6 discusses the effect of motor dynamics on practical considerations for motor and amplifier applications, such as motion profiles and the required peak and rated continuous current and torque, and maximum required voltage and motor speed. These considerations offer excellent examples of practical applications of system dynamics but are not discussed in most system dynamics texts.
5. **System dynamics in everyday life.** Commonly found illustrations of system dynamics are important for helping students to understand the material and its relevance. This text provides examples drawn from objects encountered in everyday life. These examples include a storm door closer, fluid flow from a bottle, shock absorbers and suspension springs, motors, systems with gearing, chain drives, belt drives, a backhoe, a water tower, and cooling of liquid in a cup.
6. **Case studies and theme applications.** Two common applications provide themes for case studies, examples, and problems throughout the text. These are motion-control systems, such as a conveyor system and a robot arm, and vehicle suspension systems.

ACKNOWLEDGMENTS

I want to acknowledge and thank the many individuals who contributed to this effort. At McGraw-Hill, my thanks go to the editors who helped me through four editions: Tom Casson, who initiated the project, Jonathan Plant, Lora Neyens, Bill Stenquist, and Thomas Scaiffe. I am grateful to Tina Bower and Laura Bies for their patience and help with the fourth edition.

The University of Rhode Island provided an atmosphere that encourages teaching excellence, course development, and writing, and for that I am appreciative.

I am grateful to my wife, Mary Louise; and my children, Aileene, Bill, and Andrew, for their support, patience, and understanding through forty years of textbook creation. Finally, thanks to my grandchildren, Elizabeth, Emma, James, and Henry, for many enjoyable diversions!

William J. Palm III
Kingston, Rhode Island
March 2019

ABOUT THE AUTHOR

William J. Palm III is Professor Emeritus of Mechanical, Industrial, and Systems Engineering at the University of Rhode Island. In 1966 he received a B.S. from Loyola College in Baltimore, and in 1971 a Ph.D. in Mechanical Engineering and Astronautical Sciences from Northwestern University in Evanston, Illinois.

During his forty-four years as a faculty member, he has taught nineteen courses. One of these is a junior system dynamics course, which he developed. He has authored nine textbooks dealing with modeling and simulation, system dynamics, control systems, vibrations, and MATLAB. These include *MATLAB for Engineering Applications*, fourth edition (McGraw-Hill, 2019), *A Concise Introduction to MATLAB* (McGraw-Hill, 2008), and *Differential Equations for Engineers and Scientists* (McGraw-Hill, 2013) with Yunus Çengel. He wrote a chapter on control systems in the *Mechanical Engineers' Handbook*, fourth edition (M. Kutz, ed., Wiley, 2014), and was a special contributor to the fifth editions of *Statics* and *Dynamics,* both by J. L. Meriam and L. G. Kraige (Wiley, 2002).

Professor Palm's research and industrial experience are in control systems, robotics, vibrations, and system modeling. He was the Director of the Robotics Research Center at the University of Rhode Island from 1985 to 1993, and is the co-holder of a patent for a robot hand. He served as Acting Department Chair from 2002 to 2003. His industrial experience is in automated manufacturing; modeling and simulation of naval systems, including underwater vehicles and tracking systems; and design of control systems for underwater vehicle engine test facilities.

CHAPTER 1

Introduction

CHAPTER OUTLINE

1.1 Introduction to System Dynamics 2
1.2 Units 9
1.3 Developing Linear Models 11
1.4 Introduction to Differential Equations 17
1.5 A Case Study in Motion Control 21
1.6 MATLAB Review 28
1.7 Chapter Review 34
Problems 35

CHAPTER OBJECTIVES

When you have finished this chapter, you should be able to

1. Define the basic terminology of system dynamics.
2. Apply the basic steps used for engineering problem solving.
3. Apply the necessary steps for developing a computer solution.
4. Use units in both the FPS and the SI systems.
5. Develop linear models from given algebraic expressions.
6. Use direct integration to solve dynamics problems involving a differential equation in which the derivative can be isolated.
7. Model and design a simple motion-control system for a single rotational load.
8. Use MATLAB to perform simple calculations and plotting, and use the MATLAB help system.

This chapter introduces the basic terminology of system dynamics, which includes the notions of *system*, *static* and *dynamic elements*, *input*, and *output*. Because we will use both the foot-pound-second (FPS) and the metric (SI) systems of units, the chapter introduces these two systems. Developing mathematical models of input-output relations is essential to the applications of system dynamics. Therefore, we begin our study by introducing some basic methods for developing algebraic models of static elements. We show how to use the methods of function identification and parameter estimation to develop models from data, and how to fit models to data that have little scatter. ■

1.1 INTRODUCTION TO SYSTEM DYNAMICS

This text is an introduction to system dynamics. We presume that the reader has some background in calculus (specifically, differentiation and integration of functions of a single variable) and in physics (specifically, free body diagrams, Newton's laws of motion for a particle, and elementary dc electricity). In this section we establish some basic terminology and discuss the meaning of the topic "system dynamics," its methodology, and its applications.

1.1.1 SYSTEMS

The meaning of the term *system* has become somewhat vague because of overuse. The original meaning of the term is a *combination of elements intended to act together to accomplish an objective*. For example, a link in a bicycle chain is usually not considered to be a system. However, when it is used with other links to form a chain, it becomes part of a system. The objective for the chain is to transmit force. When the chain is combined with gears, wheels, crank, handlebars, and other elements, it becomes part of a larger system whose purpose is to transport a person.

The system designer must focus on how all the elements act together to achieve the system's intended purpose, keeping in mind other important factors such as safety, cost, and so forth. Thus, the system designer often cannot afford to spend time on the details of designing the system elements. For example, our bicycle designer might not have time to study the metallurgy involved with link design; that is the role of the chain designer. All the systems designer needs to know about the chain is its strength, its weight, and its cost, because these are the factors that influence its role in the system.

With this "systems point of view," we focus on how *connections* between the elements influence the *overall* behavior of the system. This means that sometimes we must accept a less-detailed description of the operation of the individual elements to achieve an overall understanding of the system's performance.

Figure 1.1.1 illustrates a liquid-filled tank with a volume inflow f (say in cubic feet per second). The liquid height is h (say in feet). We see in Example 1.4.2 that the functional relationship between f and h has the form $f = bh^m$, where b and m are constants. We would not call this a "system." However, if two tanks are connected as shown in Figure 1.1.2, this connection forms a "system." Each tank is a "subsystem"

Figure 1.1.1 The effect of liquid height h on the out flow rate f.

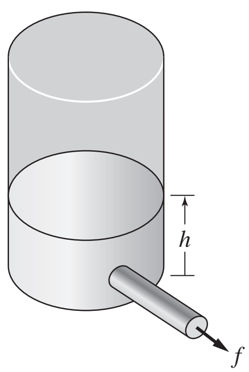
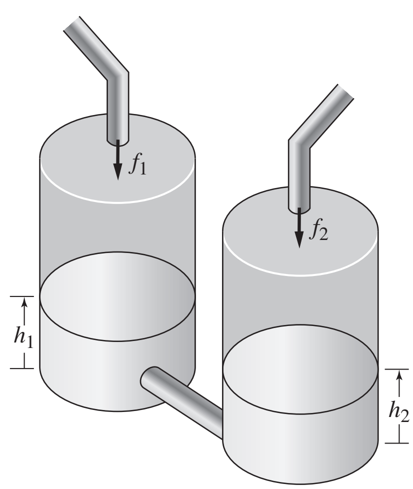


Figure 1.1.2 Two connected tanks.



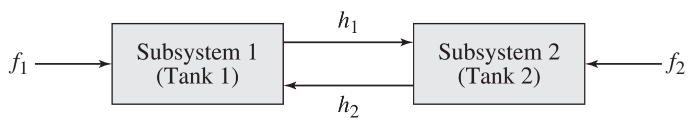


Figure 1.1.3 A system diagram illustrating how the two liquid heights affect each other.

whose liquid height is influenced by the other tank. We can obtain a differential equation model for each height by using the single-tank relationship $f = bh^m$ and applying the basic physical principle called conservation of mass to express the connection between the two tanks. This results in a model of the entire system.

We often use diagrams to illustrate the connections between the subsystems. Figure 1.1.2 illustrates the physical connection, but Figure 1.1.3 is an example of a diagram showing that the height h_1 affects the height h_2, and vice versa. (The flow goes from the higher height to the lower one.) Such a diagram may be useful for a nontechnical audience, but it does not show *how* the heights affect each other. To do that, we will use two other types of diagrams—called *simulation* diagrams and *block diagrams*—to represent the connections between the subsystems and the *variables* that describe the system behavior. These diagrams represent the differential equation model.

1.1.2 INPUT AND OUTPUT

Like the term "system," the meanings of *input* and *output* have become less precise. For example, a factory manager will call a meeting to seek "input," meaning opinions or data, from the employees, and the manager may refer to the products manufactured in the factory as its "output." However, in the system dynamics meaning of the terms, an *input* is a *cause*; an *output* is an *effect* due to the input. Thus, one input to the bicycle is the force applied to the pedal. One resulting output is the acceleration of the bike. Another input is the angle of the front wheel; the output is the direction of the bike's path of travel.

The behavior of a system element is specified by its *input-output relation*, which is a description of how the output is affected by the input. The input-output relation expresses the cause-and-effect behavior of the element. Such a description, which is represented graphically by the diagram in Figure 1.1.4, can be in the form of a table of numbers, a graph, or a mathematical relation. For example, a force f applied to a particle of mass m causes an acceleration a of the particle. The input-output or causal relation is, from Newton's second law, $a = f/m$. The input is f and the output is a.

The input-output relations for the elements in the system provide a means of specifying the connections between the elements. When connected together to form a system, the inputs to some elements will be the outputs from other elements.

The inputs and outputs of a system are determined by the selection of the system's boundary (see Figure 1.1.4). Any causes acting on the system from the world external to this boundary are considered to be system inputs. Similarly, a system's outputs are the outputs from any one or more of the system elements that act on the world outside the system boundary. If we take the bike to be the system, one system input would be the pedal force; another input is the force of gravity acting on the bike. The outputs

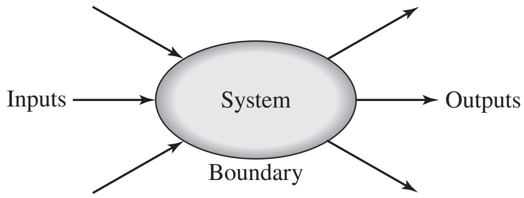


Figure 1.1.4 A system input-output diagram, showing the system boundary.

may be taken to be the bike's position, velocity, and acceleration. Usually, our choices for system outputs are a subset of the possible outputs and are the variables in which we are interested. For example, a performance analysis of the bike would normally focus on the acceleration or velocity, but not on the bike's position.

Sometimes input-output relations are reversible, sometimes not. For example, we can apply a current as input to a resistor and consider the resulting voltage drop to be the output ($v = iR$). Or we can apply a voltage to produce a current through the resistor ($i = v/R$). However, acceleration is the cause of a change in velocity, but not vice versa. If we integrate acceleration a over time, we obtain velocity v; that is, $v = \int a\,dt$. Whenever an output of an element is the time integral of the input and the direction of the cause-effect relation is not reversible, we say that the element exhibits *integral causality*. We will see that integral causality constitutes a basic form of causality for all physical systems.

Similar statements can be made about the relation between velocity and displacement. Integration of velocity produces displacement x: $x = \int v\,dt$. Velocity is the cause of displacement, but not vice versa.

Note that the mathematical relations describing integral causality can be reversed; for example, we may write $a = dv/dt$, but this does not mean that the cause-and-effect relation can be reversed.

1.1.3 STATIC AND DYNAMIC ELEMENTS

When the present value of an element's output depends only on the present value of its input, we say the element is a *static* element. For example, the current flowing through a resistor depends only on the present value of the applied voltage. The resistor is thus a static element. However, because no physical element can respond instantaneously, the concept of a static element is an approximation. It is widely used, however, because it results in a simpler mathematical representation; that is, an algebraic representation rather than one involving differential equations.

If an element's present output depends on past inputs, we say it is a *dynamic element*. For example, the present position of a bike depends on what its velocity has been from the start.

In popular usage, the terms static and dynamic distinguish situations in which no change occurs from those that are subject to changes over time. This usage conforms to the preceding definitions of these terms if the proper interpretation is made. A static element's output can change with time only if the input changes and will not change if the input is constant or absent. However, if the input is constant or removed from a dynamic element, its output can still change. For example, if we stop pedaling, the bike's displacement will continue to change because of its momentum, which is due to past inputs.

A *dynamic system* is one whose present output depends on past inputs. A *static system* is one whose output at any given time depends only on the input at that time. A static system contains all static elements. Any system that contains at least one dynamic element must be a dynamic system. *System dynamics*, then, is the study of systems that contain dynamic elements.

1.1.4 MODELING OF SYSTEMS

Table 1.1.1 contains a summary of the methodology that has been tried and tested by the engineering profession for many years. These steps describe a general problem-solving procedure. Simplifying the problem sufficiently and applying the appropriate

Table 1.1.1 Steps in engineering problem solving.

1. Understand the purpose of the problem.
2. Collect the known information. Realize that some of it might turn out to be not needed.
3. Determine what information you must find.
4. Simplify the problem only enough to obtain the required information. State any assumptions you make.
5. Draw a sketch and label any necessary variables.
6. Determine what fundamental principles are applicable.
7. Think generally about your proposed solution approach and consider other approaches before proceeding with the details.
8. Label each step in the solution process.
9. If you use a program to solve the problem, hand check the results using a simple version of the problem. Checking the dimensions and units, and printing the results of intermediate steps in the calculation sequence can uncover mistakes.
10. Perform a "reality check" on your answer. Does it make sense? Estimate the range of the expected result and compare it with your answer. Do not state the answer with greater precision than is justified by any of the following:
    a. The precision of the given information.
    b. The simplifying assumptions.
    c. The requirements of the problem.

    Interpret the mathematics. If the mathematics produces multiple answers, do not discard some of them without considering what they mean. The mathematics might be trying to tell you something, and you might miss an opportunity to discover more about the problem.

fundamental principles is called *modeling*, and the resulting mathematical description is called a *mathematical model*, or just a *model*. When the modeling has been finished, we need to solve the mathematical model to obtain the required answer. If the model is highly detailed, we may need to solve it with a computer program.

Modeling is the art of obtaining a quantitative description of a system or one of its elements that is simple enough to be useful for making predictions and realistic enough to trust those predictions. For example, consider a potato being heated in an oven. The oven designer wants to design an oven that is powerful enough to bake a potato within a prescribed time (Figure 1.1.5). Note that because the oven has yet to be designed, we cannot do an experiment to obtain the answer. Potatoes vary in size and shape, but a good estimate of the required oven power can be obtained by modeling the potato as a sphere having the thermal properties of water. Then, using the thermal systems methods given in Chapter 7, we can predict how long it will take to bake the potato.

It often is necessary to choose between a very accurate but complicated model and a simple but not so accurate model. Complicated models may be difficult to solve, or they may require experimental data that are unavailable or hard to find. There usually is no "right" model choice because it depends on the particular situation. We aim to

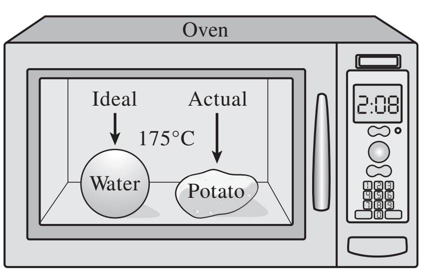


Figure 1.1.5 A potato modeled as a sphere of water.

choose the simplest model that yields adequate results. Just remember that the predictions obtained from a model are no more accurate than the simplifying assumptions made to develop the model. That is why we call modeling an art; it depends partly on judgment obtained by experience.

The form of a mathematical model depends on its purpose. For example, design of electrical equipment requires more than a knowledge of electrical principles. An electric circuit can be damaged if its mounting board experiences vibration. In this case, its force-deflection properties must be modeled. In addition, resistors generate heat, and a thermal model is required to describe this process. Thus, we see that devices can have many facets: thermal, mechanical, electrical, and so forth. No mathematical model can deal with all these facets. Even if it could, it would be too complex, and thus too cumbersome, to be useful.

For example, a map is a model of a geographic region. But if a single map contains all information pertaining to the roads, terrain elevation, geology, population density, and so on, it would be too cluttered to be useful. Instead, we select the particular type of map required for the purpose at hand. In the same way, we select or construct a mathematical model to suit the requirements of a particular study.

The examples in this text follow the steps in Table 1.1.1, although for compactness the steps are usually not numbered. In each example, following the example's title, there is a *problem statement* that summarizes the results of steps 1 through 5. Steps 6 through 10 are described in the *solution* part of the example. To save space, some steps, such as checking dimensions and units, are not always explicitly displayed. However, you are encouraged to perform these steps on your own.

1.1.5 MATHEMATICAL METHODS

Because system dynamics deals with changes in time, mathematical models of dynamic systems naturally involve differential equations. Therefore, we introduce differential equation solution methods starting in this chapter. Additional methods, such as those that make use of computers, are introduced in subsequent chapters.

1.1.6 CONTROL SYSTEMS

Often dynamic systems require a *control system* to perform properly. Thus, proper control system design is one of the most important objectives of system dynamics. Microprocessors have greatly expanded the applications for control systems. These new applications include robotics, mechatronics, micromachines, precision engineering, active vibration control, active noise cancellation, and adaptive optics. Recent technological advancements mean that many machines now operate at high speeds and high accelerations. It is therefore now more often necessary for engineers to pay more attention to the principles of system dynamics. Starting in Chapter 10, we apply these principles to control system design.

1.1.7 APPLICATIONS IN MECHANICAL SYSTEMS

Mechanical systems are loosely defined as those whose operating principles are primarily Newton's laws of motion. The bicycle is an example of a mechanical system. Chapters 3 and 4 deal with mechanical systems. The topic of mechanical vibrations covers the oscillations of machines and structures due either to their own inherent flexibility or to the action of an external force or motion. This is treated in Chapter 13.

Figure 1.1.6 A vehicle suspension system.

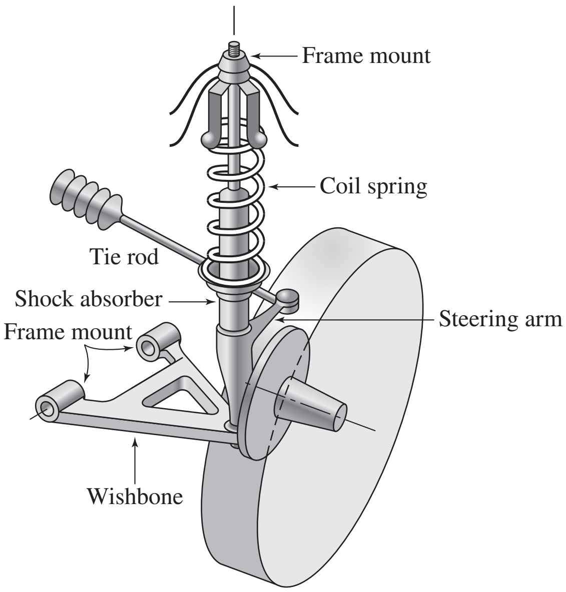
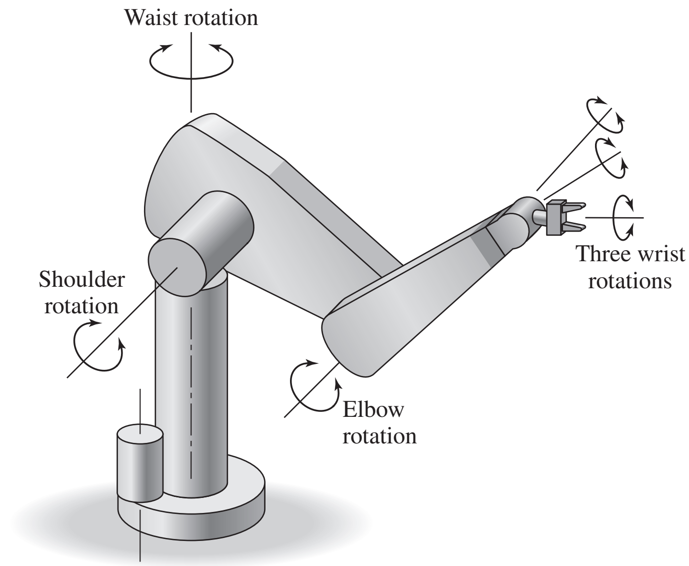


Figure 1.1.7 A robot arm.



One of our major theme applications in mechanical systems is vehicle dynamics. This topic has received renewed importance for reasons related to safety, energy efficiency, and passenger comfort. Of major interest under this topic is the design of vehicle suspension systems, whose elements include various types of springs and shock absorbers (Figure 1.1.6). *Active* suspension systems, whose characteristics can be changed under computer control, and vehicle-dynamics control systems are undergoing rapid development, and their design requires an understanding of system dynamics.

1.1.8 APPLICATIONS IN ELECTRICAL AND ELECTROMECHANICAL SYSTEMS

Electromechanical systems contain both mechanical elements and electrical elements such as electric motors. Two common applications of system dynamics in electromechanical systems are in (1) motion-control systems and (2) vehicle dynamics. Therefore, we will use these applications as major themes in many of our examples and problems. Chapter 6 introduces electrical and electromechanical systems.

Figure 1.1.7 shows a robot arm, whose motion must be properly controlled to move an object to a desired position and orientation. To do this, each of the several motors and drive trains in the arm must be adequately designed to handle the load, and the motor speeds and angular positions must be properly controlled. Figure 1.1.8 shows a typical motor and drive train for one arm joint. Knowledge of system dynamics is essential to design these subsystems and to control them properly.

Mobile robots are another motion-control application, but motion-control applications are not limited to robots. Figure 1.1.9 shows the mechanical drive for a conveyor system. The motor, the gears in the speed reducer, the chain, the sprockets, and the drive wheels all must be properly selected, and the motor must be properly controlled for the system to work well. In subsequent chapters we will develop models of these components and use them to design the system and analyze its performance.

Figure 1.1.8 Mechanical drive for a robot arm joint.

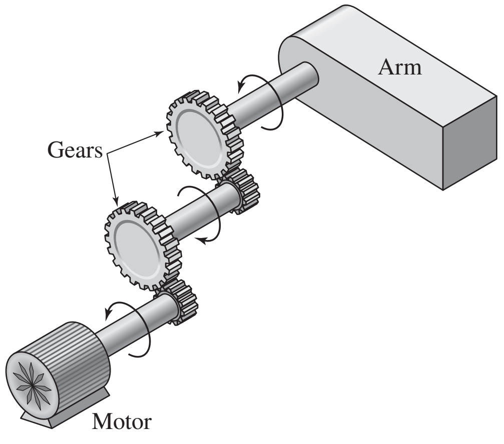
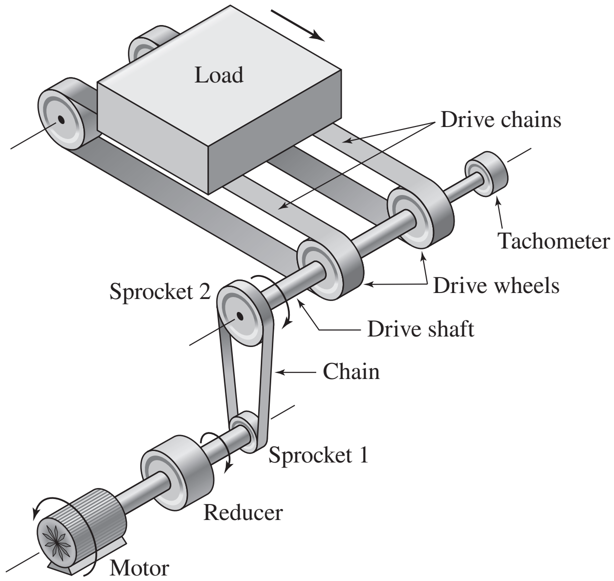


Figure 1.1.9 Mechanical drive for a conveyor system.



1.1.9 APPLICATIONS IN FLUID SYSTEMS

A fluid system is one whose operation depends on the flow of a fluid. If the fluid is *incompressible*, that is, if its density does not change appreciably with pressure changes, we call it a liquid, or a *hydraulic* fluid. On the other hand, if the fluid is *compressible*, that is, if its density does change appreciably with pressure changes, we call it a gas, or a *pneumatic* fluid.

Figure 1.1.10 shows a commonly seen backhoe. The bucket, forearm, and upper arm are each driven by a *hydraulic servomotor*. A cutaway view of such a motor is shown in Figure 1.1.11. We will analyze its behavior in Chapter 7. Compressed air

Figure 1.1.10 A backhoe.

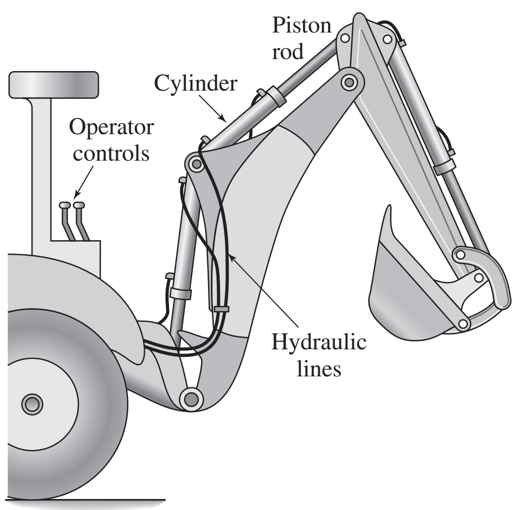


Figure 1.1.11 A hydraulic servomotor.

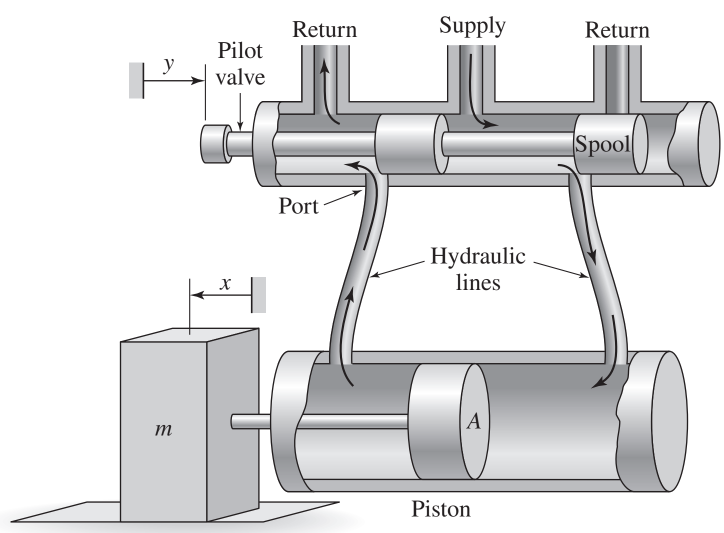


Table 1.1.2 Steps for developing a computer solution.

1. State the problem concisely.
2. Specify the data to be used by the program. This is the "input."
3. Specify the information to be generated by the program. This is the "output."
4. Work through the solution steps by hand or with a calculator; use a simpler set of data if necessary.
5. Write and run the program.
6. Check the output of the program with your hand solution.
7. Run the program with your input data and perform a reality check on the output.
8. If you will use the program as a general tool in the future, test it by running it for a range of reasonable data values, and perform a reality check on the results. Document the program with comment statements, flow charts, pseudo-code, or whatever else is appropriate.

cylinders and the common storm door closer are examples of pneumatic systems, and we encounter them in Chapter 7.

1.1.10 APPLICATIONS IN THERMAL SYSTEMS

A thermal system is one whose behavior depends primarily on the exchange of heat. The oven-potato application we saw earlier is an example of a thermal system. Many thermal systems involve fluid flow, such as with a steam engine or an air conditioner, and so we often speak of *thermo-fluid systems*. These examples also have mechanical components such as pistons, and so we could refer to them as thermo-fluid-mechanical systems, although we rarely use such cumbersome terminology. The designation as thermal, fluid, or mechanical depends on what aspect of the system we are analyzing. Thermal systems are first treated in Chapter 7, with more applications covered in later chapters.

1.1.11 COMPUTER METHODS

The computer methods used in this text are based on MATLAB and Simulink.®[1] If you are unfamiliar with MATLAB, Appendix D on the textbook website contains a thorough introduction to the program. No prior experience with Simulink is required; we will introduce the necessary methods as we need them. For the convenience of those who prefer to use a software package other than MATLAB or Simulink, we have placed all the MATLAB and Simulink material in optional sections at the end of each chapter. They can be skipped without affecting your understanding of the following chapters. If you use a program, such as MATLAB, to solve a problem, follow the steps shown in Table 1.1.2.

1.2 UNITS

In this book we use two systems of units, the *FPS* system and the metric SI. The common system of units in business and industry in English-speaking countries has been the foot-pound-second (FPS) system. This system is also known as the U.S. customary system or the British Engineering system. Much engineering work in the United States has been based on the FPS system, and some industries continue to use it. The metric Système International d'Unités (SI) nevertheless is becoming the worldwide standard.

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

Until the changeover is complete, engineers in the United States will have to be familiar with both systems.

In our examples, we will use SI and FPS units in the hope that the student will become comfortable with both. Other systems are in use, such as the meter-kilogram-second (mks) and centimeter-gram-second (cgs) metric systems and the British system, in which the mass unit is a pound. We will not use these, in order to simplify our coverage and because FPS and SI units are the most common in engineering applications. We now briefly summarize these two systems.

1.2.1 FPS UNITS

The FPS system is a *gravitational* system. This means that the primary variable is force, and the unit of mass is derived from Newton's second law. The *pound* is selected as the unit of force and the *foot* and *second* as units of length and time, respectively. From Newton's second law of motion, force equals mass times acceleration, or

$$f = ma \tag{1.2.1}$$

where f is the net force acting on the mass m and producing an acceleration a. Thus, the unit of mass must be

$$\text{mass} = \frac{\text{force}}{\text{acceleration}} = \frac{\text{pound}}{\text{foot}/(\text{second})^2}$$

This mass unit is named the *slug*.

Through Newton's second law, the weight W of an object is related to the object mass m and the acceleration due to gravity, denoted by g, as follows: $W = mg$. At the surface of the earth, the standard value of g in FPS units is $g = 32.2$ ft/sec^2.

Energy has the dimensions of mechanical work; namely, force times displacement. Therefore, the unit of energy in this system is the *foot-pound* (ft-lb). Another energy unit in common use for historical reasons is the *British thermal unit* (Btu). The relationship between the two is given in Table 1.2.1. Power is the rate of change of energy with time, and a common unit is *horsepower*. Finally, temperature in the FPS system can be expressed in degrees *Fahrenheit* or in absolute units, degrees *Rankine*.

1.2.2 SI UNITS

The SI metric system is an *absolute* system, which means that the mass is chosen as the primary variable, and the force unit is derived from Newton's law. The *meter* and the

Table 1.2.1 SI and FPS units.

	Unit name and abbreviation	
Quantity	**SI Unit**	**FPS Unit**
Time	second (s)	second (sec)
Length	meter (m)	foot (ft)
Force	newton (N)	pound (lb)
Mass	kilogram (kg)	slug
Energy	joule (J)	foot-pound (ft-lb), Btu (= 778 ft-lb)
Power	watt (W)	ft-lb/sec, horsepower (hp)
Temperature	degrees Celsius (°C), degrees Kelvin (K)	degrees Fahrenheit (°F), degrees Rankine (°R)

Table 1.2.2 Unit conversion factors.

Length	1 m = 3.281 ft	1 ft = 0.3048 m
	1 mile = 5280 ft	1 km = 1000 m
Speed	1 ft/sec = 0.6818 mi/hr	1 mi/hr = 1.467 ft/sec
	1 m/s = 3.6 km/h	1 km/h = 0.2778 m/s
	1 km/hr = 0.6214 mi/hr	1 mi/hr = 1.609 km/h
Force	1 N = 0.2248 lb	1 lb = 4.4484 N
Mass	1 kg = 0.06852 slug	1 slug = 14.594 kg
Energy	1 J = 0.7376 ft-lb	1 ft-lb = 1.3557 J
Power	1 hp = 550 ft-lb/sec	1 hp = 745.7 W
	1 W = 1.341×10^{-3} hp	
Temperature	T°C = 5 (T°F − 32)/9	T°F = 9T°C/5 + 32

second are selected as the length and time units, and the *kilogram* is chosen as the mass unit. The derived force unit is called the *newton*. In SI units the common energy unit is the newton-meter, also called the *joule*, while the power unit is the joule/second, or *watt*. Temperatures are measured in degrees Celsius, °C, and in absolute units, which are degrees *Kelvin*, K. The difference between the boiling and freezing temperatures of water is 100°C, with 0°C being the freezing point.

At the surface of the earth, the standard value of g in SI units is $g = 9.81$ m/s^2.

Table 1.2.2 gives the most commonly needed factors for converting between the FPS and the SI systems.

1.2.3 OSCILLATION UNITS

There are three commonly used units for frequency of oscillation. If time is measured in seconds, frequency can be specified as *radians/second* or as *hertz*, abbreviated Hz. One hertz is one cycle per second (cps). The relation between cycles per second f and radians per second ω is $2\pi f = \omega$. For sinusoidal oscillation, the *period* P, which is the time between peaks, is related to frequency by $P = 1/f = 2\pi/\omega$. The third way of specifying frequency is revolutions per minute (rpm). Because there are 2π radians per revolution, one rpm $=$ $(2\pi/60)$ radians per second.

1.3 DEVELOPING LINEAR MODELS

A *linear* model of a static element has the form $y = mx + b$, where x is the input and y is the output of the element. As we will see in Chapter 2, solution of dynamic models to predict system performance requires solution of differential equations. Differential equations based on linear models of the system elements are easier to solve than ones based on nonlinear models. Therefore, when developing models we try to obtain a linear model whenever possible. Sometimes the use of a linear model results in a loss of accuracy, and the engineer must weigh this disadvantage with advantages gained by using a linear model. In this section, we illustrate some ways to obtain linear models.

1.3.1 DEVELOPING LINEAR MODELS FROM DATA

If we are given data on the input-output characteristics of a system element, we can first plot the data to see whether a linear model is appropriate, and if so, we can extract a suitable model. Example 1.3.1 illustrates a common engineering problem—the estimation of the force-deflection characteristics of a cantilever support beam.

EXAMPLE 1.3.1

A Cantilever Beam Deflection Model

■ Problem

The deflection of a cantilever beam is the distance its end moves in response to a force applied at the end (Figure 1.3.1). This distance is called the *deflection* and it is the output variable. The applied force is the input. The following table gives the measured deflection x that was produced in a particular beam by the given applied force f. Plot the data to see whether a linear relation exists between f and x.

Force f (lb)	0	100	200	300	400	500	600	700	800
Deflection x (in.)	0	0.15	0.23	0.35	0.37	0.5	0.57	0.68	0.77

■ Solution

The plot is shown in Figure 1.3.2. Common sense tells us that there must be zero beam deflection if there is no applied force, so the curve describing the data must pass through the origin. The straight line shown was drawn by aligning a straightedge so that it passes through the origin

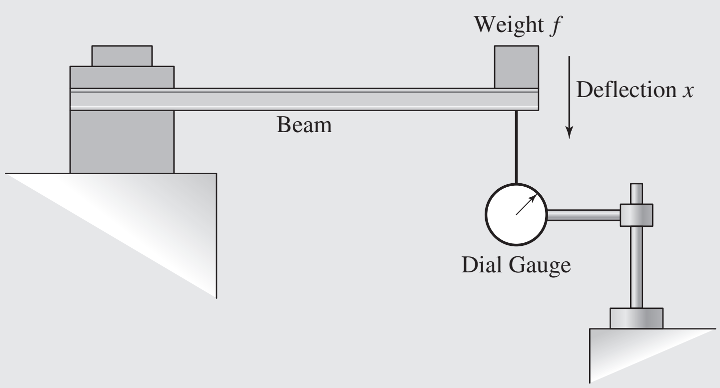


Figure 1.3.1 Measurement of beam deflection.

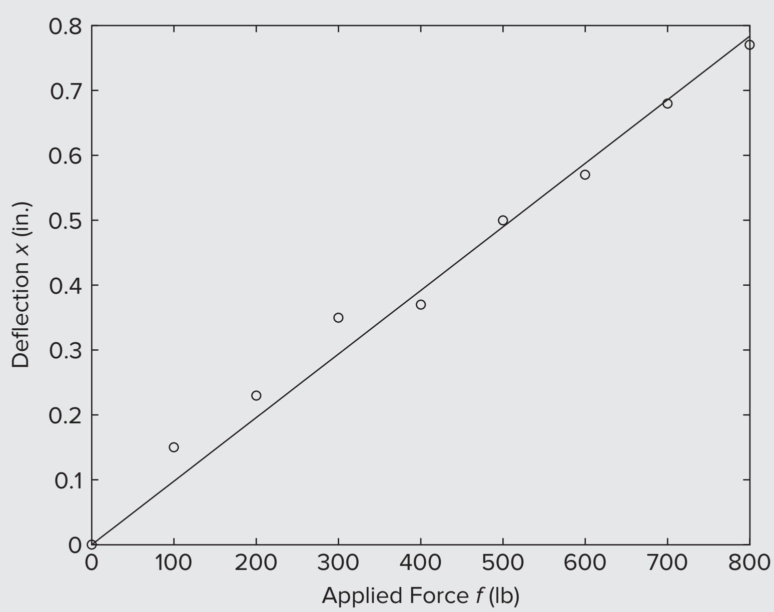


Figure 1.3.2 Plot of beam deflection versus applied force.

and near most of the data points (note that this line is subjective; another person might draw a different line). The data lie close to a straight line, so we can use the linear function $x = af$ to describe the relation. The value of the constant a can be determined from the slope of the line. Choosing the origin and the last data point to find the slope, we obtain

$$a = \frac{0.78 - 0}{800 - 0} = 9.75 \times 10^{-4} \text{ in./lb}$$

As we will see in Chapter 4, this relation is usually written as $f = kx$, where k is called the beam *stiffness*. Thus, $k = 1/a = 1025$ lb/in.

Once we have discovered a functional relation that describes the data, we can use it to make predictions for conditions that lie *within* the range of the original data. This process is called *interpolation*. For example, we can use the beam model to estimate the deflection when the applied force is 550 lb. We can be fairly confident of this prediction because we have data below and above 550 lb and we have seen that our model describes these data very well.

Extrapolation is the process of using the model to make predictions for conditions that lie *outside* the original data range. Extrapolation might be used in the beam application to predict how much force would be required to bend the beam 1.2 in. We must be careful when using extrapolation, because we usually have no reason to believe that the mathematical model is valid beyond the range of the original data. For example, if we continue to bend the beam, eventually the force is no longer proportional to the deflection, and it becomes much greater than that predicted by the linear model. Extrapolation has a use in making tentative predictions, which must be backed up later on by testing.

In some applications, the data contain so much scatter that it is difficult to identify an appropriate straight line. In such cases, we must resort to a more systematic and objective way of obtaining a model. This topic is treated in Appendix C.

1.3.2 LINEARIZATION

Not all element descriptions are in the form of data. Often we know the analytical form of the model, and if the model is nonlinear, we can obtain a linear model that is an accurate approximation over a limited range of the independent variable. Examples 1.3.2 and 1.3.3 illustrate this technique, which is called *linearization*.

Linearization of the Sine Function

EXAMPLE 1.3.2

■ **Problem**

We will see in Chapter 3 that the models of many mechanical systems involve the sine function $\sin\theta$, which is nonlinear. Obtain three linear approximations of $f(\theta) = \sin\theta$, one valid near $\theta = 0$, one near $\theta = \pi/3$ rad (60°), and one near $\theta = 2\pi/3$ rad (120°).

■ **Solution**

The essence of the linearization technique is to replace the plot of the nonlinear function with a straight line that passes through the reference point and has the same slope as the nonlinear function at that point. Figure 1.3.3 shows the sine function and the three straight lines obtained with this technique. Note that the slope of the sine function is its derivative, $d\sin\theta/d\theta = \cos\theta$, and thus the slope is not constant but varies with θ.

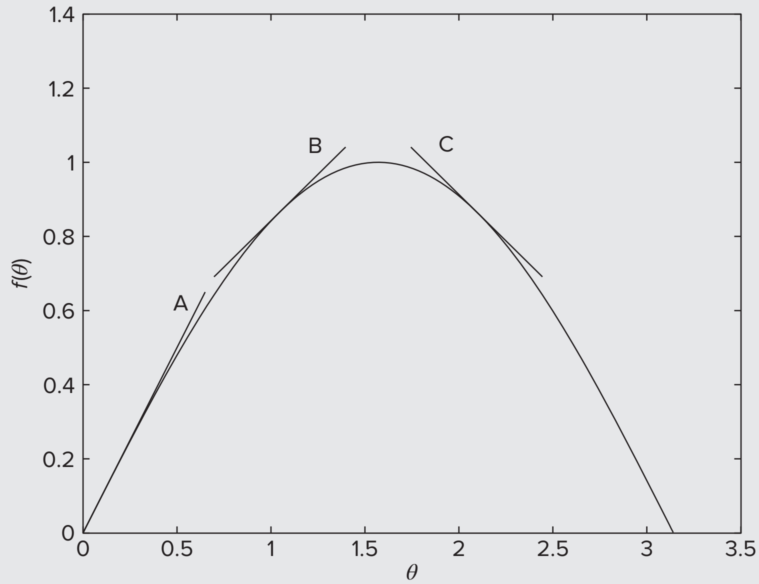


Figure 1.3.3 Three linearized models of the sine function.

Consider the first reference point, $\theta = 0$. At this point the sine function has the value $\sin 0 = 0$, the slope is $\cos 0 = 1$, and thus the straight line passing through this point with a slope of 1 is $f(\theta) = \theta$. This is the linear approximation of $f(\theta) = \sin\theta$ valid near $\theta = 0$, line A in Figure 1.3.3. Thus, we have derived the commonly seen small-angle approximation $\sin\theta \approx \theta$.

Next consider the second reference point, $\theta = \pi/3$ rad. At this point the sine function has the value $\sin\pi/3 = 0.866$, the slope is $\cos\pi/3 = 0.5$, and thus the straight line passing through this point with a slope of 0.5 is $f(\theta) = 0.5(\theta - \pi/3) + 0.866$, line B in Figure 1.3.3. This is the linear approximation of $f(\theta) = \sin\theta$ valid near $\theta = \pi/3$.

Now consider the third reference point, $\theta = 2\pi/3$ rad. At this point the sine function has the value $\sin 2\pi/3 = 0.866$, the slope is $\cos 2\pi/3 = -0.5$, and thus the straight line passing through this point with a slope of -0.5 is $f(\theta) = -0.5(\theta - 2\pi/3) + 0.866$, line C in Figure 1.3.3. This is the linear approximation of $f(\theta) = \sin\theta$ valid near $\theta = 2\pi/3$.

In Example 1.3.2 we used a graphical approach to develop the linear approximation. The linear approximation can also be developed with an analytical approach based on the Taylor series. The Taylor series represents a function $f(\theta)$ in the vicinity of $\theta = \theta_r$ by the expansion

$$f(\theta) = f(\theta_r) + \left(\frac{df}{d\theta}\right)_{\theta=\theta_r}(\theta - \theta_r) + \frac{1}{2}\left(\frac{d^2f}{d\theta^2}\right)_{\theta=\theta_r}(\theta - \theta_r)^2 + \cdots$$
$$+ \frac{1}{k!}\left(\frac{d^kf}{d\theta^k}\right)_{\theta=\theta_r}(\theta - \theta_r)^k + \cdots \qquad (1.3.1)$$

Consider the nonlinear function $f(\theta)$, which is sketched in Figure 1.3.4. Let $[\theta_r, f(\theta_r)]$ denote the reference operating condition of the system. A model that is linear can be obtained by expanding $f(\theta)$ in a Taylor series near this point and truncating

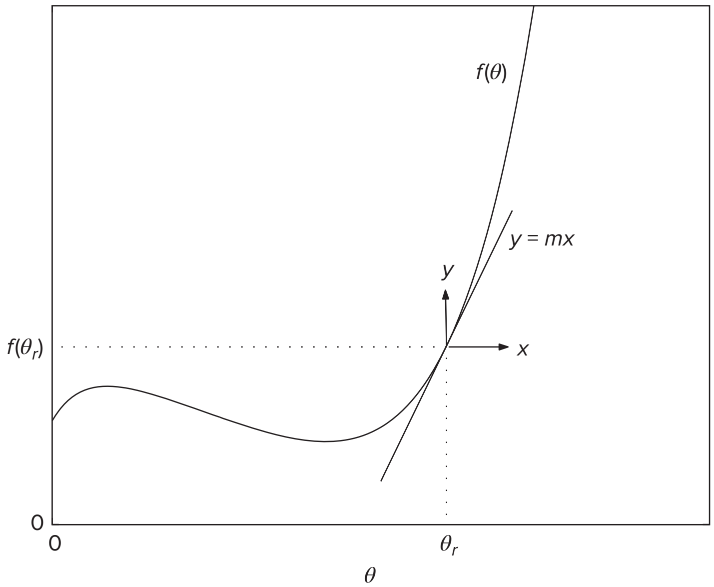


Figure 1.3.4 Graphical interpretation of function linearization.

the series beyond the first-order term. If θ is "close enough" to θ_r, the terms involving $(\theta - \theta_r)^i$ for $i \geq 2$ are small compared to the first two terms in the series. The result is

$$f(\theta) = f(\theta_r) + \left(\frac{df}{d\theta}\right)_r (\theta - \theta_r) \tag{1.3.2}$$

where the subscript r on the derivative means that it is evaluated at the reference point. This is a linear relation. To put it into a simpler form, let m denote the slope at the reference point.

$$m = \left(\frac{df}{d\theta}\right)_r \tag{1.3.3}$$

Let y denote the difference between $f(\theta)$ and the reference value $f(\theta_r)$.

$$y = f(\theta) - f(\theta_r) \tag{1.3.4}$$

Let x denote the difference between θ and the reference value θ_r.

$$x = \theta - \theta_r \tag{1.3.5}$$

Then (1.3.2) becomes

$$y = mx \tag{1.3.6}$$

The geometric interpretation of this result is shown in Figure 1.3.4. We have replaced the original function $f(\theta)$ with a straight line passing through the point $[\theta_r, f(\theta_r)]$ and having a slope equal to the slope of $f(\theta)$ at the reference point. Using the (y, x) coordinates gives a zero intercept, and simplifies the relation.

Linearization of a Square-Root Model

EXAMPLE 1.3.3

■ **Problem**

We will see in Chapter 7 that the models of many fluid systems involve the square-root function $\sqrt{h}$, which is nonlinear. Obtain a linear approximation of $f(h) = \sqrt{h}$ valid near $h = 9$.