Feedback Systems

An Introduction for Scientists and Engineers
SECOND EDITION

Karl Johan Åström
Richard M. Murray

Version v3.0h (25 Sep 2016)

This is the electronic edition of *Feedback Systems* and is available from http://www.cds.caltech.edu/~murray/FBS. Hardcover editions may be purchased from Princeton University Press, http://press.princeton.edu/titles/8701.html.

This manuscript is for personal use only and may not be reproduced, in whole or in part, without written consent from the publisher (see http://press.princeton.edu/permissions.html).

PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

Contents

Preface to the Second Edition vii

Preface to the First Edition ix

Chapter 1. Introduction 1-1

1.1 What Is Feedback? 1-1
1.2 What is Feedforward? 1-4
1.3 What Is Control? 1-4
1.4 Use of Feedback and Control 1-6
1.5 Feedback Properties 1-13
1.6 Simple Forms of Feedback 1-17
1.7 Combining Feedback with Logic 1-20
1.8 Control System Architectures 1-23
1.9 Further Reading 1-26
Exercises 1-27

Chapter 2. Feedback Principles 2-1

2.1 Mathematical Models 2-1
2.2 Using Feedback to Improve Disturbance Attenuation 2-7
2.3 Using Feedback to Follow Command Signals 2-12
2.4 Using Feedback to Provide Robustness 2-15
2.5 Positive Feedback 2-18
2.6 Feedback and Feedforward 2-23
2.7 Using Feedback to Shape Behavior 2-27
2.8 Further Reading 2-31
Exercises 2-31

Chapter 3. System Modeling 3-1

3.1 Modeling Concepts 3-1
3.2 State Space Models 3-11
3.3 Modeling Methodology 3-23
3.4 Modeling Examples 3-30
3.5 Further Reading 3-40
Exercises 3-41

Chapter 4. Examples **4-1**

4.1 Cruise Control 4-1
4.2 Bicycle Dynamics 4-5
4.3 Operational Amplifier Circuits 4-7
4.4 Computing Systems and Networks 4-11
4.5 Atomic Force Microscopy 4-17
4.6 Drug Administration 4-21
4.7 Population Dynamics 4-25
Exercises 4-27

Chapter 5. Dynamic Behavior **5-1**

5.1 Solving Differential Equations 5-1
5.2 Qualitative Analysis 5-4
5.3 Stability 5-8
5.4 Lyapunov Stability Analysis 5-17
5.5 Parametric and Nonlocal Behavior 5-27
5.6 Further Reading 5-32
Exercises 5-33

Chapter 6. Linear Systems **6-1**

6.1 Basic Definitions 6-1
6.2 The Matrix Exponential 6-6
6.3 Input/Output Response 6-15
6.4 Linearization 6-28
6.5 Further Reading 6-33
Exercises 6-34

Chapter 7. State Feedback **7-1**

7.1 Reachability 7-1
7.2 Stabilization by State Feedback 7-9
7.3 Design Considerations 7-17
7.4 Integral Action 7-30
7.5 Further Reading 7-32
Exercises 7-33

Chapter 8. Output Feedback **8-1**

8.1 Observability 8-1
8.2 State Estimation 8-6
8.3 Control Using Estimated State 8-11
8.4 Kalman Filtering 8-17
8.5 State Space Design 8-21
8.6 Further Reading 8-27
Exercises 8-27

Chapter 9. Transfer Functions **9-1**

9.1 Frequency Domain Modeling 9-1
9.2 Determining the Transfer Function 9-3
9.3 Gain Poles and Zeros 9-13
9.4 Block Diagrams and Transfer Functions 9-18
9.5 The Bode Plot 9-24
9.6 Further Reading 9-33
Exercises 9-34

Chapter 10. Frequency Domain Analysis **10-1**

10.1 The Loop Transfer Function 10-1
10.2 The Nyquist Criterion 10-4
10.3 Stability Margins 10-12
10.4 Bode's Relations and Minimum Phase Systems 10-16
10.5 Generalized Notions of Gain and Phase 10-19
10.6 Further Reading 10-23
Exercises 10-24

Chapter 11. PID Control **11-1**

11.1 Basic Control Functions 11-1
11.2 Simple Controllers for Complex Systems 11-6
11.3 PID Tuning 11-10
11.4 Integrator Windup 11-15
11.5 Implementation 11-19
11.6 Further Reading 11-24
Exercises 11-24

Chapter 12. Frequency Domain Design **12-1**

12.1 Sensitivity Functions 12-1
12.2 Feedforward Design 12-5
12.3 Performance Specifications 12-12
12.4 Feedback Design via Loop Shaping 12-16
12.5 The Root-Locus Method 12-21
12.6 Fundamental Limitations 12-25
12.7 Design Example 12-35
12.8 Further Reading 12-38
Exercises 12-38

Chapter 13. Robust Performance **13-1**

13.1 Modeling Uncertainty 13-1
13.2 Stability in the Presence of Uncertainty 13-6
13.3 Performance in the Presence of Uncertainty 13-12
13.4 Robust Pole Placement 13-15

13.5 Design for Robust Performance 13-23
13.6 Further Reading 13-28
Exercises 13-29

Chapter 14. Architecture and System Design 14-1
14.1 Introduction 14-1
14.2 System and Control Design 14-2
14.3 Top-Down Architectures 14-5
14.4 Bottom-Up Architectures 14-15
14.5 Interaction 14-21
14.6 Adaptation, Learning and Cognition 14-24
14.7 Control Design in Common Application Fields 14-33
14.8 Further Reading 14-37

Bibliography B-1

Index I-1

Preface to the Second Edition

The second edition of *Feedback Systems* contains a variety of changes that are based on feedback on the first edition, particularly in its use for introductory courses in control. One of the primary comments from users of the text was that the use of control tools for design purposes occurred only after several chapters of analytical tools, leaving the instructor having to try to convince students that the techniques would soon be useful. In our own teaching, we find that we often use design examples in the first few weeks of the class and use this to motivate the various techniques that follow. This approach has been particularly useful in engineering courses, where students are often eager to apply the tools to examples as part of gaining insight into the methods. We also found that universities that have a laboratory component attached to their controls class need to introduce some basic design techniques early, so that students can be implementing control laws in the laboratory in the early weeks of the course.

To help emphasize this more design-oriented flow, we have rearranged the material in the first third of the book. We have added a new chapter on "Feedback Principles" that illustrates some simple design principles and tools that can be used to show students what types of problems can be solved using feedback. This new chapter uses simple models, simulations and elementary analysis techniques, so that it should be accessible to students from a variety of engineering and scientific backgrounds. For courses in which students have already been exposed to the basic ideas of feedback, perhaps in an earlier discipline-specific course, this new chapter can easily be skipped without any loss of continuity.

We have also added a new chapter to the end of the book, focused on control architectures and design. Our intention in this chapter is to provide a systems view that describes how control design is integrated into a larger model-based development framework, motivated in part by our consulting activities with large companies. In this new chapter we also take the opportunity to present some overview material on "bottoms up" and "top down" approaches to control architectures, briefly introducing some of the many additional concepts from the field of control that are in widespread use in applications.

In addition to these relatively large changes, we have made many other smaller changes based on the feedback we have received from early adopters of the text. We have added some material on the Routh-Hurwitz criterion and root locus plots, to at least serve as "hooks" for instructors who wish to cover that material in more detail. We have also made some notational changes throughout, most notably changing the symbols for disturbance and noise signals to v and w, respectively.

The notation in the biological examples has also been updated to match the notation used in the textbook by Del Vecchio and Murray [DM14]. Finally, there are a variety of changes on the companion web site, including the addition of python code (using the python-control toolbox [Pyt]) for many of the examples.

We are indebted to numerous individuals who have taught out of the text and sent us feedback on changes that would better serve their needs. In addition to the many individuals listed in the preface to the first edition, we would like to also thank Constantine Caramanis, Clancy Rowley and André Tits for their feedback and insights.

Karl Johan Åström
Lund, Sweden

Richard M. Murray
Pasadena, California

Preface to the First Edition

This book provides an introduction to the basic principles and tools for the design and analysis of feedback systems. It is intended to serve a diverse audience of scientists and engineers who are interested in understanding and utilizing feedback in physical, biological, information and social systems. We have attempted to keep the mathematical prerequisites to a minimum while being careful not to sacrifice rigor in the process. We have also attempted to make use of examples from a variety of disciplines, illustrating the generality of many of the tools while at the same time showing how they can be applied in specific application domains.

A major goal of this book is to present a concise and insightful view of the current knowledge in feedback and control systems. The field of control started by teaching everything that was known at the time and, as new knowledge was acquired, additional courses were developed to cover new techniques. A consequence of this evolution is that introductory courses have remained the same for many years, and it is often necessary to take many individual courses in order to obtain a good perspective on the field. In developing this book, we have attempted to condense the current knowledge by emphasizing fundamental concepts. We believe that it is important to understand why feedback is useful, to know the language and basic mathematics of control and to grasp the key paradigms that have been developed over the past half century. It is also important to be able to solve simple feedback problems using back-of-the-envelope techniques, to recognize fundamental limitations and difficult control problems and to have a feel for available design methods.

This book was originally developed for use in an experimental course at Caltech involving students from a wide set of backgrounds. The course was offered to undergraduates at the junior and senior levels in traditional engineering disciplines, as well as first- and second-year graduate students in engineering and science. This latter group included graduate students in biology, computer science and physics. Over the course of several years, the text has been classroom tested at Caltech and at Lund University, and the feedback from many students and colleagues has been incorporated to help improve the readability and accessibility of the material.

Because of its intended audience, this book is organized in a slightly unusual fashion compared to many other books on feedback and control. In particular, we introduce a number of concepts in the text that are normally reserved for second-year courses on control and hence often not available to students who are not control systems majors. This has been done at the expense of certain traditional topics, which we felt that the astute student could learn independently and are often

explored through the exercises. Examples of topics that we have included are nonlinear dynamics, Lyapunov stability analysis, the matrix exponential, reachability and observability, and fundamental limits of performance and robustness. Topics that we have deemphasized include root locus techniques, lead/lag compensation and detailed rules for generating Bode and Nyquist plots by hand.

Several features of the book are designed to facilitate its dual function as a basic engineering text and as an introduction for researchers in natural, information and social sciences. The bulk of the material is intended to be used regardless of the audience and covers the core principles and tools in the analysis and design of feedback systems. Advanced sections, marked by the "dangerous bend" symbol shown here, contain material that requires a slightly more technical background, of the sort that would be expected of senior undergraduates in engineering. A few sections are marked by two dangerous bend symbols and are intended for readers with more specialized backgrounds, identified at the beginning of the section. To limit the length of the text, several standard results and extensions are given in the exercises, with appropriate hints toward their solutions.

To further augment the printed material contained here, a companion web site has been developed and is available from the publisher's web page:

http://www.cds.caltech.edu/~murray/amwiki

The web site contains a database of frequently asked questions, supplemental examples and exercises, and lecture material for courses based on this text. The material is organized by chapter and includes a summary of the major points in the text as well as links to external resources. The web site also contains the source code for many examples in the book, as well as utilities to implement the techniques described in the text. Most of the code was originally written using MATLAB M-files but was also tested with LabView MathScript to ensure compatibility with both packages. Many files can also be run using other scripting languages such as Octave, SciLab, SysQuake and Xmath.

The first half of the book focuses almost exclusively on state space control systems. We begin in Chapter 3[1] with a description of modeling of physical, biological and information systems using ordinary differential equations and difference equations. Chapter 4 presents a number of examples in some detail, primarily as a reference for problems that will be used throughout the text. Following this, Chapter 5 looks at the dynamic behavior of models, including definitions of stability and more complicated nonlinear behavior. We provide advanced sections in this chapter on Lyapunov stability analysis because we find that it is useful in a broad array of applications and is frequently a topic that is not introduced until later in one's studies.

The remaining three chapters of the first half of the book focus on linear systems, beginning with a description of input/output behavior in Chapter 6. In Chapter 7, we formally introduce feedback systems by demonstrating how state space control laws can be designed. This is followed in Chapter 8 by material on output

[1] Chapter numbers reflect those in the second edition.

feedback and estimators. Chapters 7 and 8 introduce the key concepts of reachability and observability, which give tremendous insight into the choice of actuators and sensors, whether for engineered or natural systems.

The second half of the book presents material that is often considered to be from the field of "classical control." This includes the transfer function, introduced in Chapter 9, which is a fundamental tool for understanding feedback systems. Using transfer functions, one can begin to analyze the stability of feedback systems using frequency domain analysis, including the ability to reason about the closed loop behavior of a system from its open loop characteristics. This is the subject of Chapter 10, which revolves around the Nyquist stability criterion.

In Chapters 11 and 12, we again look at the design problem, focusing first on proportional-integral-derivative (PID) controllers and then on the more general process of loop shaping. PID control is by far the most common design technique in control systems and a useful tool for any student. The chapter on frequency domain design introduces many of the ideas of modern control theory, including the sensitivity function. In Chapter 13, we combine the results from the second half of the book to analyze some of the fundamental trade-offs between robustness and performance. This is also a key chapter illustrating the power of the techniques that have been developed and serving as an introduction for more advanced studies.

The book is designed for use in a 10- to 15-week course in feedback systems that provides many of the key concepts needed in a variety of disciplines. For a 10-week course, Chapters 1–3, 5–7 and 9–12 can each be covered in a week's time, with the omission of some topics from the final chapters. A more leisurely course, spread out over 14–15 weeks, could cover the entire book, with 2 weeks on modeling (Chapters 3 and 2)—particularly for students without much background in ordinary differential equations—and 2 weeks on robust performance (Chapter 13).

The mathematical prerequisites for the book are modest and in keeping with our goal of providing an introduction that serves a broad audience. We assume familiarity with the basic tools of linear algebra, including matrices, vectors and eigenvalues. These are typically covered in a sophomore-level course on the subject, and the textbooks by Apostol [Apo69], Arnold [Arn87] and Strang [Str88] can serve as good references. Similarly, we assume basic knowledge of differential equations, including the concepts of homogeneous and particular solutions for linear ordinary differential equations in one variable. Apostol [Apo69] and Boyce and DiPrima [BD04] cover this material well. Finally, we also make use of complex numbers and functions and, in some of the advanced sections, more detailed concepts in complex variables that are typically covered in a junior-level engineering or physics course in mathematical methods. Apostol [Apo67] or Stewart [Ste02] can be used for the basic material, with Ahlfors [Ahl66], Marsden and Hoffman [MH98] or Saff and Snider [SS02] being good references for the more advanced material. We have chosen not to include appendices summarizing these various topics since there are a number of good books available.

One additional choice that we felt was important was the decision not to rely on a knowledge of Laplace transforms in the book. While their use is by far the

most common approach to teaching feedback systems in engineering, many students in the natural and information sciences may lack the necessary mathematical background. Since Laplace transforms are not required in any essential way, we have included them only in an advanced section intended to tie things together for students with that background. Of course, we make tremendous use of *transfer functions*, which we introduce through the notion of response to exponential inputs, an approach we feel is more accessible to a broad array of scientists and engineers. For classes in which students have already had Laplace transforms, it should be quite natural to build on this background in the appropriate sections of the text.

Acknowledgments

The authors would like to thank the many people who helped during the preparation of this book. The idea for writing this book came in part from a report on future directions in control [Mur03] to which Stephen Boyd, Roger Brockett, John Doyle and Gunter Stein were major contributors. Kristi Morgansen and Hideo Mabuchi helped teach early versions of the course at Caltech on which much of the text is based, and Steve Waydo served as the head TA for the course taught at Caltech in 2003–2004 and provided numerous comments and corrections. Charlotta Johnsson and Anton Cervin taught from early versions of the manuscript in Lund in 2003–2007 and gave very useful feedback. Other colleagues and students who provided feedback and advice include Leif Andersson, John Carson, K. Mani Chandy, Michel Charpentier, Domitilla Del Vecchio, Kate Galloway, Per Hagander, Toivo Henningsson Perby, Joseph Hellerstein, George Hines, Tore Hägglund, Cole Lepine, Anders Rantzer, Anders Robertsson, Dawn Tilbury and Francisco Zabala. The reviewers for Princeton University Press and Tom Robbins at NI Press also provided valuable comments that significantly improved the organization, layout and focus of the book. Our editor, Vickie Kearn, was a great source of encouragement and help throughout the publishing process. Finally, we would like to thank Caltech, Lund University and the University of California at Santa Barbara for providing many resources, stimulating colleagues and students, and pleasant working environments that greatly aided in the writing of this book.

Karl Johan Åström
Lund, Sweden
Santa Barbara, California

Richard M. Murray
Pasadena, California

Chapter One
Introduction

Feedback is a central feature of life. The process of feedback governs how we grow, respond to stress and challenge, and regulate factors such as body temperature, blood pressure and cholesterol level. The mechanisms operate at every level, from the interaction of proteins in cells to the interaction of organisms in complex ecologies.

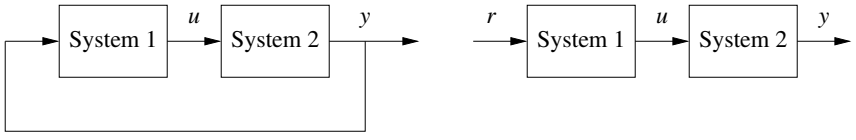
M. B. Hoagland and B. Dodson, *The Way Life Works*, 1995 [HD95].

In this chapter we provide an introduction to the basic concept of *feedback* and the related engineering discipline of *control*. We focus on both historical and current examples, with the intention of providing the context for current tools in feedback and control.

1.1 What Is Feedback?

A *dynamical system* is a system whose behavior changes over time, often in response to external stimulation or forcing. The term *feedback* refers to a situation in which two (or more) dynamical systems are connected together such that each system influences the other and their dynamics are thus strongly coupled. Simple causal reasoning about a feedback system is difficult because the first system influences the second and the second system influences the first, leading to a circular argument. A consequence of this is that the behavior of feedback systems is often counter-intuitive, and it is therefore necessary to resort to formal methods to understand them.

Figure 1.1 illustrates in block diagram form the idea of feedback. We often use the terms *open loop* and *closed loop* when referring to such systems. A system is said to be a closed loop system if the systems are interconnected in a cycle, as shown in Figure 1.1a. If we break the interconnection, we refer to the configuration



(a) Closed loop (b) Open loop

Figure 1.1: Open and closed loop systems. (a) The output of system 1 is used as the input of system 2, and the output of system 2 becomes the input of system 1, creating a closed loop system. (b) The interconnection between system 2 and system 1 is removed, and the system is said to be open loop.

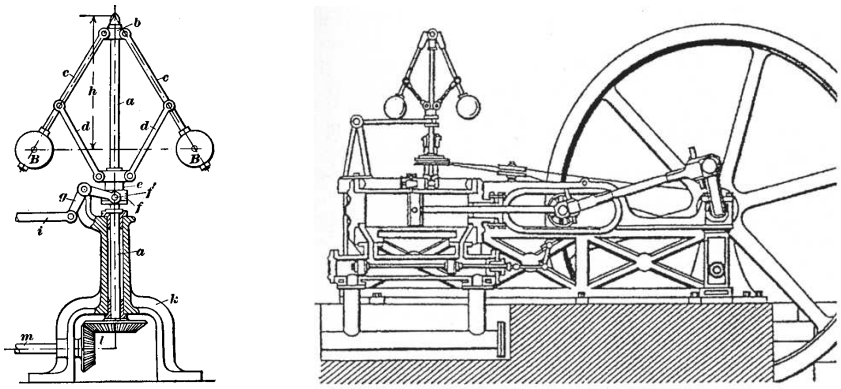


Figure 1.2: The centrifugal governor and the steam engine. The centrifugal governor on the left consists of a set of flyballs that spread apart as the speed of the engine increases. The steam engine on the right uses a centrifugal governor (above and to the left of the flywheel) to regulate its speed. (Credit: Machine a Vapeur Horizontale de Philip Taylor [1828].)

as an open loop system, as shown in Figure 1.1b. Note that since the system is in a feedback loop, the choice of system 1 versus system 2 is somewhat arbitrary. It just depends where you want to start describing how the system works.

As the quote at the beginning of this chapter illustrates, a major source of examples of feedback systems is biology. Biological systems make use of feedback in an extraordinary number of ways, on scales ranging from molecules to cells to organisms to ecosystems. One example is the regulation of glucose in the bloodstream through the production of insulin and glucagon by the pancreas. The body attempts to maintain a constant concentration of glucose, which is used by the body's cells to produce energy. When glucose levels rise (after eating a meal, for example), the hormone insulin is released and causes the body to store excess glucose in the liver. When glucose levels are low, the pancreas secretes the hormone glucagon, which has the opposite effect. Referring to Figure 1.1, we can view the liver as system 1 and the pancreas as system 2. The output from the liver is the glucose concentration in the blood, and the output from the pancreas is the amount of insulin or glucagon produced. The interplay between insulin and glucagon secretions throughout the day helps to keep the blood-glucose concentration constant, at about 90 mg per 100 mL of blood.

An early engineering example of a feedback system is a centrifugal governor, in which the shaft of a steam engine is connected to a flyball mechanism that is itself connected to the throttle of the steam engine, as illustrated in Figure 1.2. The system is designed so that as the speed of the engine increases (perhaps because of a lessening of the load on the engine), the flyballs spread apart and a linkage causes the throttle on the steam engine to be closed. This in turn slows down the engine, which causes the flyballs to come back together. We can model this system as a closed loop system by taking system 1 as the steam engine and system 2 as the governor. When properly designed, the flyball governor maintains a constant

speed of the engine, roughly independent of the loading conditions. The centrifugal governor was an enabler of the successful Watt steam engine, which fueled the industrial revolution.

The examples given so far all deal with *negative feedback*, in which we attempt to react to disturbances in a such a way that their effects decrease. *Positive feedback* is the opposite, where the increase in some variable or signal leads to a situation in which that quantity is further increased through feedback. This has a destabilizing effect and is usually accompanied by a saturation that limits the growth of the quantity. Although often considered undesirable, this behavior is used in biological (and engineering) systems to obtain a very fast response to a condition or signal. Encouragement is a positive feedback that is often used in education. Another common use of positive feedback is the design of systems with oscillatory dynamics.

One example of the use of positive feedback is to create switching behavior, in which a system maintains a given state until some input crosses a threshold. Hysteresis is often present so that noisy inputs near the threshold do not cause the system to jitter. This type of behavior is called *bistability* and is often associated with memory devices.

Feedback has many interesting properties that can be exploited in designing systems. As in the case of glucose regulation or the flyball governor, feedback can make a system resilient toward external influences. It can also be used to create linear behavior out of nonlinear components, a common approach in electronics. More generally, feedback allows a system to be insensitive both to external disturbances and to variations in its individual elements.

Feedback has potential disadvantages as well. It can create dynamic instabilities in a system, causing oscillations or even runaway behavior. Another drawback, especially in engineering systems, is that feedback can introduce unwanted sensor noise into the system, requiring careful filtering of signals. It is for these reasons that a substantial portion of the study of feedback systems is devoted to developing an understanding of dynamics and a mastery of techniques in dynamical systems.

Feedback systems are ubiquitous in both natural and engineered systems. Control systems maintain the environment, lighting and power in our buildings and factories; they regulate the operation of our cars, consumer electronics and manufacturing processes; they enable our transportation and communications systems; and they are critical elements in our military and space systems. For the most part they are hidden from view, buried within the code of embedded microprocessors, executing their functions accurately and reliably. Feedback has also made it possible to increase dramatically the precision of instruments such as atomic force microscopes (AFMs) and telescopes.

In nature, homeostasis in biological systems maintains thermal, chemical and biological conditions through feedback. At the other end of the size scale, global climate dynamics depend on the feedback interactions between the atmosphere, the oceans, the land and the sun. Ecosystems are filled with examples of feedback due to the complex interactions between animal and plant life. Even the dynam-

Table 1.1: Properties of feedback and feedforward

Feedback	Feedforward
Closed loop	Open loop
Acts on deviations	Acts on plans
Robust to model uncertainty	Sensitive to model uncertainty
Risk for instability	No risk for instability
Sensitive to measurement noise	Insensitive to measurement noise

ics of economies are based on the feedback between individuals and corporations through markets and the exchange of goods and services.

1.2 What is Feedforward?

Feedback is reactive: there must be an error before corrective actions are taken. However, in some circumstances it is possible to measure a disturbance before it enters the system, and this information can then be used to take corrective action before the disturbance has influenced the system. The effect of the disturbance is thus reduced by measuring it and generating a control signal that counteracts it. This way of controlling a system is called *feedforward*. Feedforward is particularly useful in shaping the response to command signals because command signals are always available. Since feedforward attempts to match two signals, it requires good process models; otherwise the corrections may have the wrong size or may be badly timed.

The ideas of feedback and feedforward are very general and appear in many different fields. In economics, feedback and feedforward are analogous to a market-based economy versus a planned economy. In business, a feedforward strategy corresponds to running a company based on extensive strategic planning, while a feedback strategy corresponds to a reactive approach. In biology, feedforward has been suggested as an essential element for motion control in humans that is tuned during training. Experience indicates that it is often advantageous to combine feedback and feedforward, and the correct balance requires insight and understanding of their respective properties which are summarized in Table 1.1.

1.3 What Is Control?

The term *control* has many meanings and often varies between communities. In this book, we define control to be the use of algorithms and feedback in engineered systems. Thus, control includes such examples as feedback loops in electronic amplifiers, setpoint controllers in chemical and materials processing, "fly-by-wire" systems on aircraft and even router protocols that control traffic flow on the Internet. Emerging applications include high-confidence software systems, autonomous vehicles and robots, real-time resource management systems and biologically en-

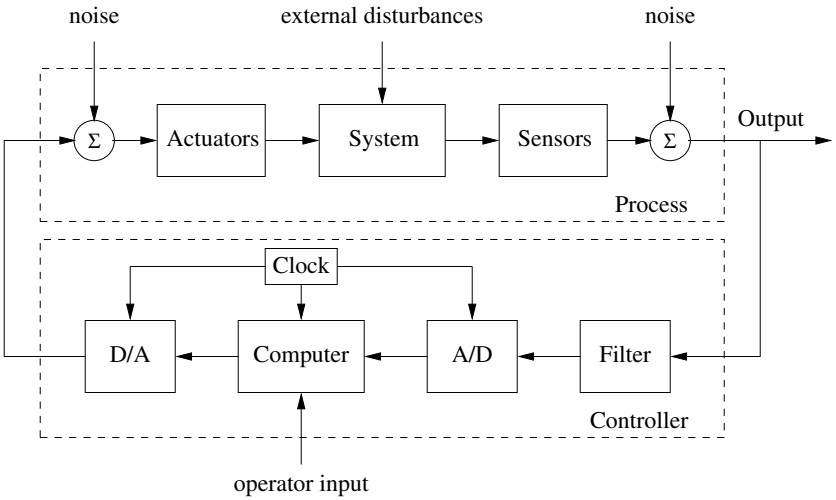


Figure 1.3: Components of a computer-controlled system. The upper dashed box represents the process dynamics, which include the sensors and actuators in addition to the dynamical system being controlled. Noise and external disturbances can perturb the dynamics of the process. The controller is shown in the lower dashed box. It consists of a filter and analog-to-digital (A/D) and digital-to-analog (D/A) converters, as well as a computer that implements the control algorithm. A system clock controls the operation of the controller, synchronizing the A/D, D/A and computing processes. The operator input is also fed to the computer as an external input.

gineered systems. At its core, control is an *information* science and includes the use of information in both analog and digital representations.

A modern controller senses the operation of a system, compares it against the desired behavior, computes corrective actions based on a model of the system's response to external inputs and actuates the system to effect the desired change. This basic *feedback loop* of sensing, computation and actuation is the central concept in control. The key issues in designing control logic are ensuring that the dynamics of the closed loop system are stable (bounded disturbances give bounded errors) and that they have additional desired behavior (good disturbance attenuation, fast responsiveness to changes in operating point, etc). These properties are established using a variety of modeling and analysis techniques that capture the essential dynamics of the system and permit the exploration of possible behaviors in the presence of uncertainty, noise and component failure.

A typical example of a control system is shown in Figure 1.3. The basic elements of sensing, computation and actuation are clearly seen. In modern control systems, computation is typically implemented on a digital computer, requiring the use of analog-to-digital (A/D) and digital-to-analog (D/A) converters. Uncertainty enters the system through noise in sensing and actuation subsystems, external disturbances that affect the underlying system operation and uncertain dynamics in the system (parameter errors, unmodeled effects, etc). The algorithm that computes the control action as a function of the sensor values is often called a *control*

law. The system can be influenced externally by an operator who introduces *command signals* to the system.

Control engineering relies on and shares tools from physics (dynamics and modeling), computer science (information and software) and operations research (optimization, probability theory and game theory), but it is also different from these subjects in both insights and approach.

Perhaps the strongest area of overlap between control and other disciplines is in the modeling of physical systems, which is common across all areas of engineering and science. One of the fundamental differences between control-oriented modeling and modeling in other disciplines is the way in which interactions between subsystems are represented. Control relies on a type of input/output modeling that allows many new insights into the behavior of systems, such as disturbance attenuation and stable interconnection. Model reduction, where a simpler (lower-fidelity) description of the dynamics is derived from a high-fidelity model, is also naturally described in an input/output framework. Perhaps most importantly, modeling in a control context allows the design of *robust* interconnections between subsystems, a feature that is crucial in the operation of all large engineered systems.

Control is also closely associated with computer science since virtually all modern control algorithms for engineering systems are implemented in software. However, control algorithms and software can be very different from traditional computer software because of the central role of the dynamics of the system and the real-time nature of the implementation.

1.4 Use of Feedback and Control

Feedback has many interesting and useful properties. It makes it possible to design precise systems from imprecise components and to make relevant quantities in a system change in a prescribed fashion. An unstable system can be stabilized using feedback, and the effects of external disturbances can be reduced. Feedback also offers new degrees of freedom to a designer by exploiting sensing, actuation and computation. In this section we briefly survey some of the important applications and trends for feedback in the world around us. Considerably more detail is available in several recent reports [Mur03, MÅB+03, SA14].

Power Generation and Transmission. Access to electrical power has been one of the major drivers of technological progress in modern society. Much of the early development of control was driven by the generation and distribution of electrical power. Control is mission critical for power systems, and there are many control loops in individual power stations. Control is also important for the operation of the whole power network since it is difficult to store energy and it is thus necessary to match production to consumption. Power management is a straightforward regulation problem for a system with one generator and one power consumer, but it is more difficult in a highly distributed system with many generators and long distances between consumption and generation. Power demand can change rapidly in

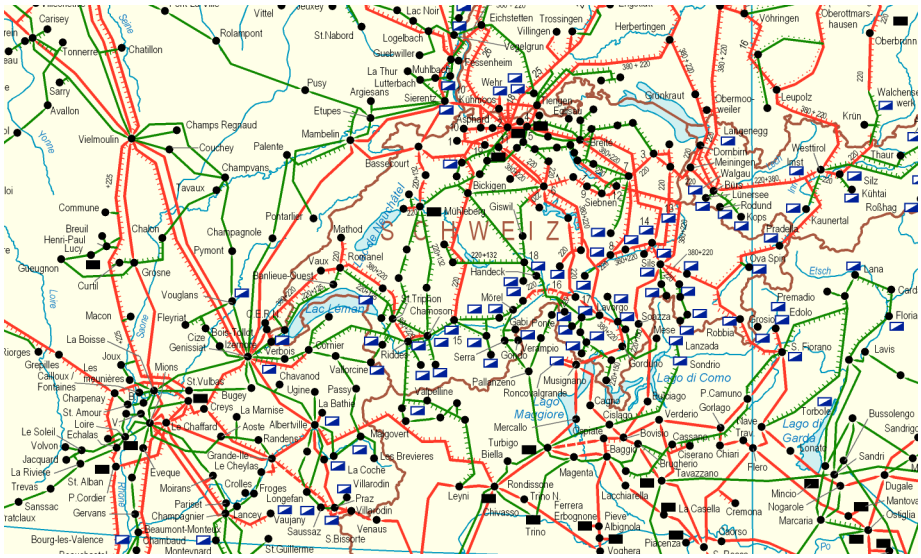


Figure 1.4: A small portion of the European power network. In 2016 European power suppliers operated a single interconnected network covering a region from the Arctic to the Mediterranean and from the Atlantic to the Urals. The installed power was more than 800 GW (8×10^{11} W) serving more than 500 million citizens. (Source: ENTSO-E http://www.entsoe.eu)

an unpredictable manner and combining generators and consumers into large networks makes it possible to share loads among many suppliers and to average consumption among many customers. Large transcontinental and transnational power systems have therefore been built, such as the one show in Figure 1.4.

Telecommunications. When telecommunications emerged in the early 20th century there was a strong need to amplify signals to enable telephone communication over long distances. The only amplifier available at the time was based on vacuum tubes. Since the properties of vacuum tubes are nonlinear and time varying the amplifiers created a lot of distortion. A major advance was made when Black invented the negative feedback amplifier[Bla34, Bla77], which made it possible to obtain stable amplifiers with linear characteristics. Research on feedback amplifiers also generated fundamental understanding of feedback in the form of Nyquist's stability criterion and Bode's methods for design of feedback amplifiers and his theorems on fundamental limitations [Bod45, Nyq56]. Feedback is used extensively in cellular phones and networks and the future 5G communication networks will permit execution of control over the networks.

Aerospace and Transportation. In aerospace, control has been a key technological capability tracing back to the beginning of the 20th century. Indeed, the Wright brothers are correctly famous not for demonstrating simply powered flight but *controlled* powered flight. Their early Wright Flyer incorporated moving control surfaces (vertical fins and canards) and warpable wings that allowed the pilot to

regulate the aircraft's flight. In fact, the aircraft itself was not stable, so continuous pilot corrections were mandatory. This early example of controlled flight was followed by a fascinating success story of continuous improvements in flight control technology, culminating in the high-performance, highly reliable automatic flight control systems we see in modern commercial and military aircraft today.

Materials and Processing. The chemical industry is responsible for the remarkable progress in developing new materials that are key to our modern society. In addition to the continuing need to improve product quality, several other factors in the process control industry are drivers for the use of control. Environmental statutes continue to place stricter limitations on the production of pollutants, forcing the use of sophisticated pollution control devices. Environmental safety considerations have led to the design of smaller storage capacities to diminish the risk of major chemical leakage, requiring tighter control on upstream processes and, in some cases, supply chains. And large increases in energy costs have encouraged engineers to design plants that are highly integrated, coupling many processes that used to operate independently. All of these trends increase the complexity of these processes and the performance requirements for the control systems, making control system design increasingly challenging.

Instrumentation. The measurement of physical variables is of prime interest in science and engineering. Consider, for example, an accelerometer, where early instruments consisted of a mass suspended on a spring with a deflection sensor. The precision of such an instrument depends critically on accurate calibration of the spring and the sensor. There is also a design compromise because a weak spring gives high sensitivity but low bandwidth. A different way of measuring acceleration is to use *force feedback*. The spring is replaced by a voice coil that is controlled so that the mass remains at a constant position. The acceleration is proportional to the current through the voice coil. In such an instrument, the precision depends entirely on the calibration of the voice coil and does not depend on the sensor, which is used only as the feedback signal. The sensitivity/bandwidth compromise is also avoided.

Another important application of feedback is in instrumentation for biological systems. Feedback is widely used to measure ion currents in cells using a device called a *voltage clamp*, which is illustrated in Figure 1.5. Hodgkin and Huxley used the voltage clamp to investigate propagation of action potentials in the giant axon of the squid. In 1963 they shared the Nobel Prize in Medicine with Eccles for "their discoveries concerning the ionic mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane." A refinement of the voltage clamp called a *patch clamp* made it possible to measure exactly when a single ion channel is opened or closed. This was developed by Neher and Sakmann, who received the 1991 Nobel Prize in Medicine "for their discoveries concerning the function of single ion channels in cells."

Robotics and Intelligent Machines. The goal of cybernetic engineering, already articulated in the 1940s and even before, has been to implement systems capable of

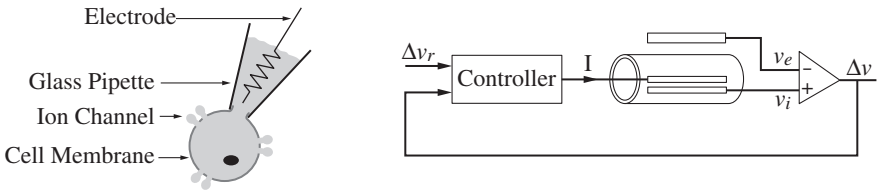


Figure 1.5: The voltage clamp method for measuring ion currents in cells using feedback. A pipette is used to place an electrode in a cell (left and middle) and maintain the potential of the cell at a fixed level. The internal voltage in the cell is v_i, and the voltage of the external fluid is v_e. The feedback system (right) controls the current I into the cell so that the voltage drop across the cell membrane $\Delta v = v_i - v_e$ is equal to its reference value Δv_r. The current I is then equal to the ion current.

exhibiting highly flexible or "intelligent" responses to changing circumstances. In 1948 the MIT mathematician Norbert Wiener gave a widely read account of cybernetics [Wie48]. A more mathematical treatment of the elements of engineering cybernetics was presented by H. S. Tsien in 1954, driven by problems related to the control of missiles [Tsi54]. Together, these works and others of that time form much of the intellectual basis for modern work in robotics and control.

Two recent areas of advancement in robotics and autonomous systems are (consumer) drones and autonomous cars, some examples of which are shown in Figure 1.6. Quadrocopters such as the DJI Phantom make use of GPS receivers, accelerometers, magnetometers and gyros to provide stable flight, and also use stabilized camera platforms to provide high quality images and movies. Autonomous vehicles, such as the Google autonomous car project, make use of a variety of laser rangefinders, cameras and radars to perceive their environment and then use sophisticated decision-making and control algorithms to enable them to safely drive in a variety of traffic conditions, from high-speed freeways to crowded city streets.

Networks and Computing Systems. Control of networks is a large research area spanning many topics, including congestion control, routing, data caching and

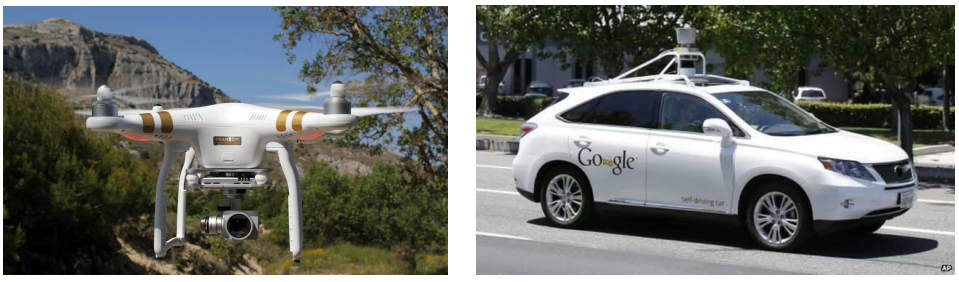
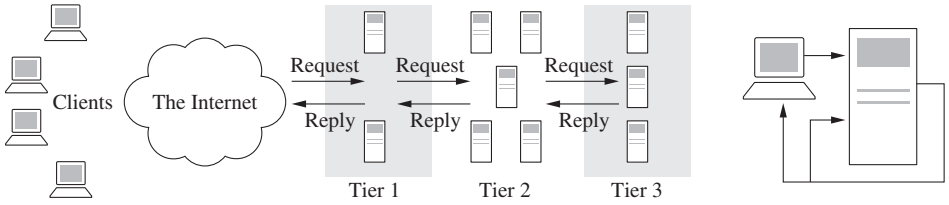


Figure 1.6: Autonomous vehicles. The figure on the left is a DJI Phantom 3 drone, which is able to maintain its position using GPS and inertial sensors. The figure on the right is an autonomous car that was developed by Google and is capable of driving on city streets by using sophisticated sensing and decision-making (control) software.



(a) Multitiered Internet services (b) Individual server

Figure 1.7: A multitier system for services on the Internet. In the complete system shown schematically in (a), users request information from a set of computers (tier 1), which in turn collect information from other computers (tiers 2 and 3). The individual server shown in (b) has a set of reference parameters set by a (human) system operator, with feedback used to maintain the operation of the system in the presence of uncertainty. (Based on Hellerstein et al. [HDPT04].)

power management. Several features of these control problems make them very challenging. The dominant feature is the extremely large scale of the system; the Internet is probably the largest feedback control system humans have ever built. Another is the decentralized nature of the control problem: decisions must be made quickly and based only on local information. Stability is complicated by the presence of varying time lags, as information about the network state can be observed or relayed to controllers only after a delay, and the effect of a local control action can be felt throughout the network only after substantial delay.

Related to the control of networks is control of the servers that sit on these networks. Computers are key components of the systems of routers, web servers and database servers used for communication, electronic commerce, advertising and information storage. A typical example of a multilayer system for e-commerce is shown in Figure 1.7a. The system has several tiers of servers. The edge server accepts incoming requests and routes them to the HTTP server tier where they are parsed and distributed to the application servers. The processing for different requests can vary widely, and the application servers may also access external servers managed by other organizations. Control of an individual server in a layer is illustrated in Figure 1.7b. A quantity representing the quality of service or cost of operation—such as response time, throughput, service rate or memory usage—is measured in the computer. The control variables might represent incoming messages accepted, priorities in the operating system or memory allocation. The feedback loop then attempts to maintain quality-of-service variables within a target range of values.

Economics. The economy is a large, dynamical system with many actors: governments, organizations, companies and individuals. Governments control the economy through laws and taxes, the central banks by setting interest rates and companies by setting prices and making investments. Individuals control the economy through purchases, savings and investments. Many efforts have been made to model and control the system both at the macro level and at the micro level, but this modeling is difficult because the system is strongly influenced by the behaviors of

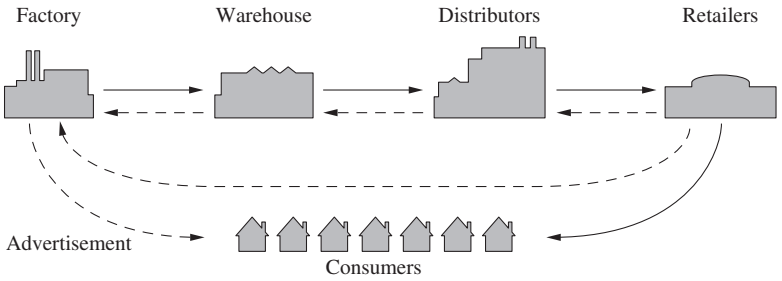


Figure 1.8: Supply chain dynamics (after Forrester [For61]). Products flow from the producer to the customer through distributors and retailers as indicated by the solid lines. There are typically many factories and warehouses and even more distributors and retailers. Multiple feedback loops are present as each agent tries to maintain the proper inventory level.

the different actors in the system.

The financial system can be viewed as a global controller for the economy. Unfortunately this important controller does not always function as desired, as expressed by the following quote by Paul Krugman [Kru09]:

> We have magneto trouble, said John Maynard Keynes at the start of the Great Depression: most of the economic engine was in good shape, but a crucial component, the financial system, was not working. He also said this: "We have involved ourselves in a colossal muddle, having blundered in the control of a delicate machine, the working of which we do not understand." Both statements are as true now as they were then.

One of the reasons why it is difficult to model economic systems is that conservation laws for important variables are missing. A typical example is that the value of a company as expressed by its stock can change rapidly and erratically. There are, however, some areas with conservation laws that permit accurate modeling. One example is the flow of products from a manufacturer to a retailer as illustrated in Figure 1.8. The products are physical quantities that obey a conservation law, and the system can be modeled by accounting for the number of products in the different inventories. There are considerable economic benefits in controlling supply chains so that products are available to customers while minimizing products that are in storage. The real problems are more complicated than indicated in the figure because there may be many different products, there may be different factories that are geographically distributed and the factories may require raw material or subassemblies.

Feedback in Nature. Many problems in the natural sciences involve understanding aggregate behavior in complex large-scale systems. This behavior emerges from the interaction of a multitude of simpler systems with intricate patterns of information flow. Representative examples can be found in fields ranging from embryology to seismology. Researchers who specialize in the study of specific complex

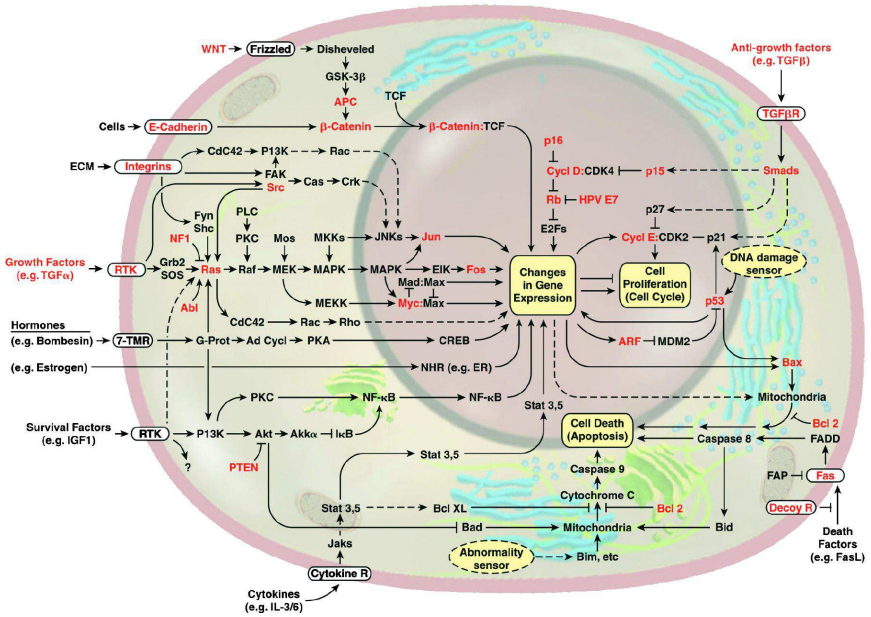


Figure 1.9: The wiring diagram of the growth-signaling circuitry of the mammalian cell [HW00]. The major pathways that are thought to play a role in cancer are indicated in the diagram. Lines represent interactions between genes and proteins in the cell. Lines ending in arrowheads indicate activation of the given gene or pathway; lines ending in a T-shaped head indicate repression. (Used with permission of Elsevier Ltd. and the authors.)

systems often develop an intuitive emphasis on analyzing the role of feedback (or interconnection) in facilitating and stabilizing aggregate behavior. We briefly highlight three application areas here.

A major theme currently of interest to the biology community is the science of reverse (and eventually forward) engineering of biological control networks such as the one shown in Figure 1.9. There are a wide variety of biological phenomena that provide a rich source of examples of control, including gene regulation and signal transduction; hormonal, immunological and cardiovascular feedback mechanisms; muscular control and locomotion; active sensing, vision and proprioception; attention and consciousness; and population dynamics and epidemics. Each of these (and many more) provide opportunities to figure out what works, how it works, and what we can do to affect it.

In contrast to individual cells and organisms, emergent properties of aggregations and ecosystems inherently reflect selection mechanisms that act on multiple levels, and primarily on scales well below that of the system as a whole. Because ecosystems are complex, multiscale dynamical systems, they provide a broad range of new challenges for the modeling and analysis of feedback systems. Recent experience in applying tools from control and dynamical systems to bacterial networks suggests that much of the complexity of these networks is due to the presence of multiple layers of feedback loops that provide robust functionality to

the individual cell [Kit04, SSS+04, YHSD00a]. Yet in other instances, events at the cell level benefit the colony at the expense of the individual. Systems level analysis can be applied to ecosystems with the goal of understanding the robustness of such systems and the extent to which decisions and events affecting individual species contribute to the robustness and/or fragility of the ecosystem as a whole.

In nature, development of organisms and their control systems have often developed in synergy. The development of birds is an interesting example, as noted by John Maynard Smith in 1952 [Smi52b]:

> The earliest birds pterosaurs, and flying insects were stable. This is believed to be because in the absence of a highly evolved sensory and nervous system they would have been unable to fly if they were not. To a flying animal there are great advantages to be gained by instability. Among the most obvious is manoeuvrability. It is of equal importance to an animal which catches its food in the air and to the animals upon which it preys. It appears that in the birds and at least in some insects the evolution of the sensory and nervous systems rendered the stability found in earlier forms no longer necessary.

1.5 Feedback Properties

Feedback is a powerful idea which, as we have seen, is used extensively in natural and technological systems. The principle of feedback is simple: base correcting actions on the difference between desired and actual performance. In engineering, feedback has been rediscovered and patented many times in many different contexts. The use of feedback has often resulted in vast improvements in system capability, and these improvements have sometimes been revolutionary, as discussed above. The reason for this is that feedback has some truly remarkable properties. In this section we will discuss some of the properties of feedback that can be understood intuitively. This intuition will be formalized in subsequent chapters.

Robustness to Uncertainty

One of the key uses of feedback is to provide robustness to uncertainty. By measuring the difference between the sensed value of a regulated signal and its desired value, we can supply a corrective action. If the system undergoes some change that affects the regulated signal, then we sense this change and try to force the system back to the desired operating point. This is precisely the effect that Watt exploited in his use of the centrifugal governor on steam engines.

As an example of this principle, consider the simple feedback system shown in Figure 1.10. In this system, the speed of a vehicle is controlled by adjusting the amount of gas flowing to the engine. Simple *proportional-integral* (PI) feedback is used to make the amount of gas depend on both the error between the current

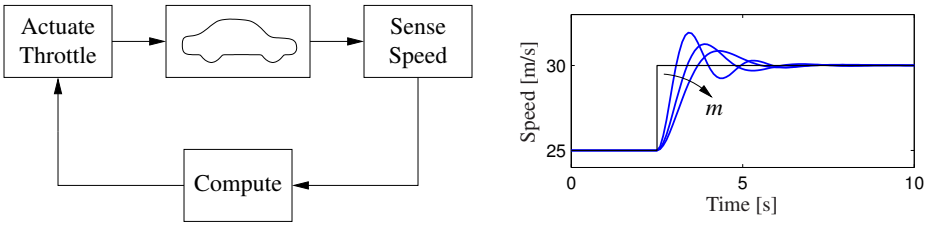


Figure 1.10: A feedback system for controlling the speed of a vehicle. In the block diagram on the left, the speed of the vehicle is measured and compared to the desired speed within the "Compute" block. Based on the difference in the actual and desired speeds, the throttle (or brake) is used to modify the force applied to the vehicle by the engine, drivetrain and wheels. The figure on the right shows the response of the control system to a commanded change in speed from 25 m/s to 30 m/s. The three different curves correspond to differing masses of the vehicle, between 1000 and 3000 kg, demonstrating the robustness of the closed loop system to a very large change in the vehicle characteristics.

and the desired speed and the integral of that error. The plot on the right shows the results of this feedback for a step change in the desired speed and a variety of different masses for the car, which might result from having a different number of passengers or towing a trailer. Notice that independent of the mass (which varies by a factor of 3!), the steady-state speed of the vehicle always approaches the desired speed and achieves that speed within approximately 5 s. Thus the performance of the system is robust with respect to this uncertainty.

Another early example of the use of feedback to provide robustness is the negative feedback amplifier. When telephone communications were developed, amplifiers were used to compensate for signal attenuation in long lines. A vacuum tube was a component that could be used to build amplifiers. Distortion caused by the nonlinear characteristics of the tube amplifier together with amplifier drift were obstacles that prevented the development of line amplifiers for a long time. A major breakthrough was the invention of the feedback amplifier in 1927 by Harold S. Black, an electrical engineer at Bell Telephone Laboratories. Black used *negative feedback*, which reduces the gain but makes the amplifier insensitive to variations in tube characteristics. This invention made it possible to build stable amplifiers with linear characteristics despite the nonlinearities of the vacuum tube amplifier.

Design of Dynamics

Another use of feedback is to change the dynamics of a system. Through feedback, we can alter the behavior of a system to meet the needs of an application: systems that are unstable can be stabilized, systems that are sluggish can be made responsive and systems that have drifting operating points can be held constant. Control theory provides a rich collection of techniques to analyze the stability and dynamic response of complex systems and to place bounds on the behavior of such systems by analyzing the gains of linear and nonlinear operators that describe their components.

An example of the use of control in the design of dynamics comes from the

area of flight control. The following quote, from a lecture presented by Wilbur Wright to the Western Society of Engineers in 1901 [McF53], illustrates the role of control in the development of the airplane:

> Men already know how to construct wings or airplanes, which when driven through the air at sufficient speed, will not only sustain the weight of the wings themselves, but also that of the engine, and of the engineer as well. Men also know how to build engines and screws of sufficient lightness and power to drive these planes at sustaining speed ... Inability to balance and steer still confronts students of the flying problem ... When this one feature has been worked out, the age of flying will have arrived, for all other difficulties are of minor importance.

The Wright brothers thus realized that control was a key issue to enable flight. They resolved the compromise between stability and maneuverability by building an airplane, the Wright Flyer, that was unstable but maneuverable. The Flyer had a rudder in the front of the airplane, which made the plane very maneuverable. A disadvantage was the necessity for the pilot to keep adjusting the rudder to fly the plane: if the pilot let go of the stick, the plane would crash. Other early aviators tried to build stable airplanes. These would have been easier to fly, but because of their poor maneuverability they could not be brought up into the air. The Wright Brothers were well aware of the compromise between stability and maneuverability when the designed they Wright Flyer [Dra55] and they made the first successful flight at Kitty Hawk in 1903. Modern fighter airplanes are also unstable in certain flight regimes, such as take-off and landing.

Since it was quite tiresome to fly an unstable aircraft, there was strong motivation to find a mechanism that would stabilize an aircraft. Such a device, invented by Sperry, was based on the concept of feedback. Sperry used a gyro-stabilized pendulum to provide an indication of the vertical. He then arranged a feedback mechanism that would pull the stick to make the plane go up if it was pointing down, and vice versa. The Sperry autopilot was the first use of feedback in aeronautical engineering, and Sperry won a prize in a competition for the safest airplane in Paris in 1914. Figure 1.11 shows the Curtiss seaplane and the Sperry autopilot. The autopilot is a good example of how feedback can be used to stabilize an unstable system and hence "design the dynamics" of the aircraft.

Creating Modularity

Feedback can be used to create modularity and shape well-defined relations between inputs and outputs in a structured hierarchical manner. A modular system is one in which individual components can be replaced without having to modify the entire system. By using feedback, it is possible to allow components to maintain their input/output properties in a manner that is robust to changes in its interconnections. A typical example is the electrical drive system shown in Figure 1.12, which has an architecture with three cascaded loops. The innermost loop is a cur-

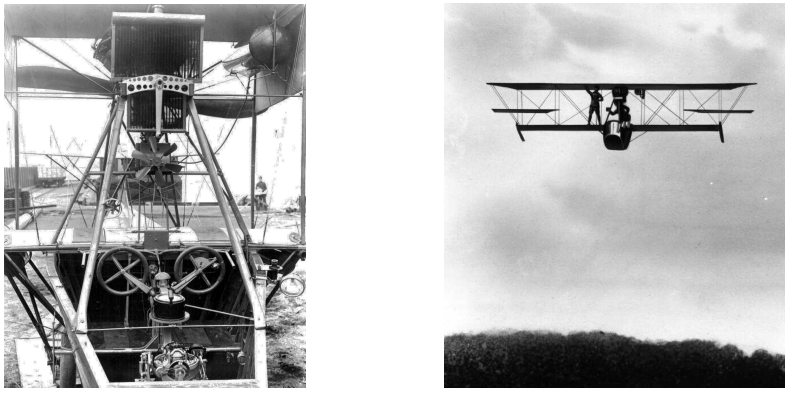


Figure 1.11: Aircraft autopilot system. The Sperry autopilot (left) contained a set of four gyros coupled to a set of air valves that controlled the wing surfaces. The 1912 Curtiss used an autopilot to stabilize the roll, pitch and yaw of the aircraft and was able to maintain level flight as a mechanic walked on the wing (right) [Hug93].

rent loop, the controller CC drives the amplifier so that the current to the motor follows the set point. The velocity loop with the controller VC drives the set point of the current controller so that velocity follows the set point of VC. The outer loop drives the set point of the velocity loop to follow the set point of the position controller

The control architecture with nested loops shown in Figure 1.12 is common. It simplifies both design, commissioning and operation. Consider for example the design of the velocity loop. With a well-designed current controller the motor current follows the set point of the controller CC. Since the motor velocity is proportional to the current the dynamics relating velocity to the input of the current controller is approximately an integrator, because force is proportional to current and angular acceleration is proportional to force. This simplified model can be used to design the velocity loop so that effects of friction and other disturbances are reduced. With a well-designed velocity loop, the design of the position loop is also simple. The loops can also be tuned sequentially starting with the inner loop. The architecture illustrates how feedback can be used to simplify modeling and create modular

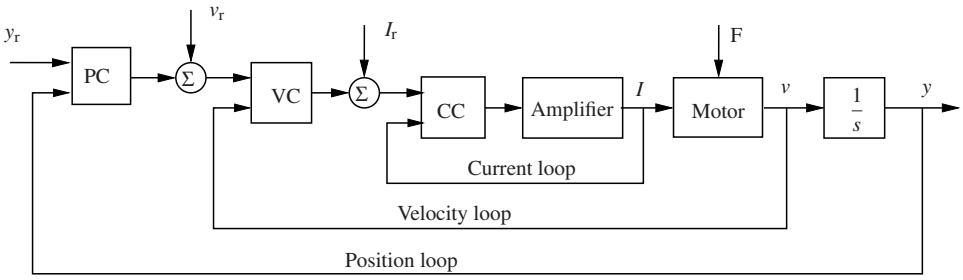


Figure 1.12: Block diagram of a system for position control. The system has three cascaded loops for control of current, velocity and position.

systems.

Challenges of Feedback

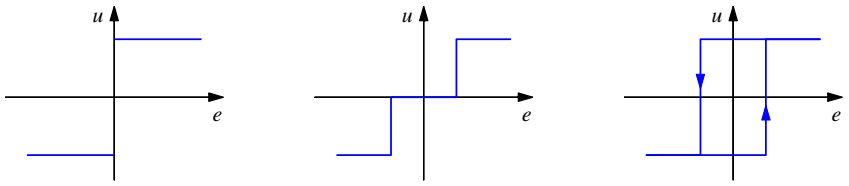
While feedback has many advantages, it also has some potential drawbacks. Chief among these is the possibility of instability if the system is not designed properly. We are all familiar with the effects of *positive feedback* when the amplification on a microphone is turned up too high in a room. This is an example of feedback instability, something that we obviously want to avoid. This is tricky because we must design the system not only to be stable under nominal conditions but also to remain stable under all possible perturbations of the dynamics.

In addition to the potential for instability, feedback inherently couples different parts of a system. One common problem is that feedback often injects measurement noise into the system. Measurements must be carefully filtered so that the actuation and process dynamics do not respond to them, while at the same time ensuring that the measurement signal from the sensor is properly coupled into the closed loop dynamics (so that the proper levels of performance are achieved).

Another potential drawback of control is the complexity of embedding a control system in a product. While the cost of sensing, computation and actuation has decreased dramatically in the past few decades, the fact remains that control systems are often complicated, and hence one must carefully balance the costs and benefits. An early engineering example of this is the use of microprocessor-based feedback systems in automobiles.The use of microprocessors in automotive applications began in the early 1970s and was driven by increasingly strict emissions standards, which could be met only through electronic controls. Early systems were expensive and failed more often than desired, leading to frequent customer dissatisfaction. It was only through aggressive improvements in technology that the performance, reliability and cost of these systems allowed them to be used in a transparent fashion. Even today, the complexity of these systems is such that it is difficult for an individual car owner to fix problems.

1.6 Simple Forms of Feedback

The idea of feedback to make corrective actions based on the difference between the desired and the actual values of a quantity can be implemented in many different ways. The benefits of feedback can be obtained by very simple feedback laws such as on-off control, proportional control and proportional-integral-derivative control. In this section we provide a brief preview of some of the topics that will be studied more formally in the remainder of the text.



(a) On-off control (b) Dead zone (c) Hysteresis

Figure 1.13: Input/output characteristics of on-off controllers. Each plot shows the input on the horizontal axis and the corresponding output on the vertical axis. Ideal on-off control is shown in (a), with modifications for a dead zone (b) or hysteresis (c). Note that for on-off control with hysteresis, the output depends on the value of past inputs.

On-Off Control

A simple feedback mechanism can be described as follows:

$$u = \begin{cases} u_{\max} & \text{if } e > 0 \\ u_{\min} & \text{if } e < 0, \end{cases} \tag{1.1}$$

where the *control error* $e = r - y$ is the difference between the reference signal (or command signal) r and the output of the system y and u is the actuation command. Figure 1.13a shows the relation between error and control. This control law implies that maximum corrective action is always used.

The feedback in equation (1.1) is called *on-off control*. One of its chief advantages is that it is simple and there are no parameters to choose. On-off control often succeeds in keeping the process variable close to the reference, such as the use of a simple thermostat to maintain the temperature of a room. It typically results in a system where the controlled variables oscillate, which is often acceptable if the oscillation is sufficiently small.

Notice that in equation (1.1) the control variable is not defined when the error is zero. It is common to make modifications by introducing either a dead zone or hysteresis (see Figure 1.13b and 1.13c).

PID Control

The reason why on-off control often gives rise to oscillations is that the system overreacts since a small change in the error makes the actuated variable change over the full range. This effect is avoided in *proportional control*, where the characteristic of the controller is proportional to the control error for small errors. This can be achieved with the control law

$$u = \begin{cases} u_{\max} & \text{if } e \geq e_{\max} \\ k_p e & \text{if } e_{\min} < e < e_{\max} \\ u_{\min} & \text{if } e \leq e_{\min}, \end{cases} \tag{1.2}$$

where k_p is the controller gain, $e_{\min} = u_{\min}/k_p$ and $e_{\max} = u_{\max}/k_p$. The interval $(e_{\min}, e_{\max})$ is called the *proportional band* because the behavior of the controller is linear when the error is in this interval:

$$u = k_p(r-y) = k_p e \qquad \text{if } e_{\min} \le e \le e_{\max}. \tag{1.3}$$

While a vast improvement over on-off control, proportional control has the drawback that the process variable often deviates from its reference value. In particular, if some level of control signal is required for the system to maintain a desired value, then we must have $e \neq 0$ in order to generate the requisite input.

This can be avoided by making the control action proportional to the integral of the error:

$$u(t) = k_i \int_0^t e(\tau) d\tau. \tag{1.4}$$

This control form is called *integral control*, and k_i is the integral gain. It can be shown through simple arguments that a controller with integral action has zero steady-state error (Exercise 1.5). The catch is that there may not always be a steady state because the system may be oscillating. In addition, if the control action has magnitude limits, as in equation (1.2), an effect known as "integrator windup" can occur and may result in poor performance unless appropriate "anti-windup" compensation is used. Despite the potential drawbacks, which can be overcome with careful analysis and design, the benefits of integral feedback in providing zero error in the presence of constant disturbances have made it one of the most used forms of feedback.

An additional refinement is to provide the controller with an anticipative ability by using a prediction of the error. A simple prediction is given by the linear extrapolation

$$e(t+T_d) \approx e(t) + T_d \frac{de(t)}{dt},$$

which predicts the error T_d time units ahead. Combining proportional, integral and derivative control, we obtain a controller that can be expressed mathematically as

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau + k_d \frac{de(t)}{dt}. \tag{1.5}$$

The control action is thus a sum of three terms: the past as represented by the integral of the error, the present as represented by the proportional term and the future as represented by a linear extrapolation of the error (the derivative term). This form of feedback is called a *proportional-integral-derivative (PID) controller* and its action is illustrated in Figure 1.14.

A PID controller is very useful and is capable of solving a wide range of control problems. More than 95% of all industrial control problems are solved by PID control, although many of these controllers are actually *proportional-integral* (PI) *controllers* because derivative action is often not included [DM02a]. There are also more advanced controllers, which differ from PID controllers by using more sophisticated methods for prediction.

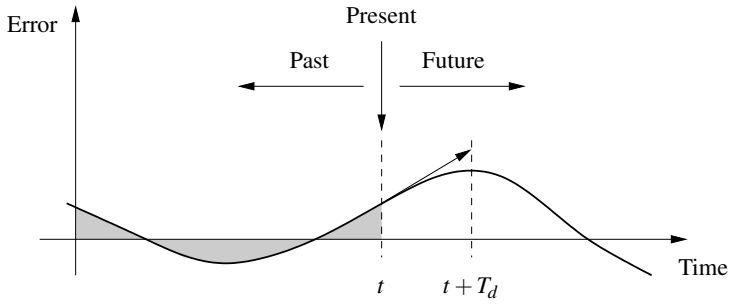


Figure 1.14: Action of a PID controller. At time t, the proportional term depends on the instantaneous value of the error. The integral portion of the feedback is based on the integral of the error up to time t (shaded portion). The derivative term provides an estimate of the growth or decay of the error over time by looking at the rate of change of the error. T_d represents the approximate amount of time in which the error is projected forward (see text).

1.7 Combining Feedback with Logic

The PID controller is a continuous time system. The on-off controller can be viewed both as a controller and a logic system. Continuous control is often combined with logic to cope with different operating conditions. Logic is typically related to changes in operating conditions, equipment protection, manual interaction and saturating actuators. One situation is when there is one variable that is of primary interest, but other variables may have to be controlled for equipment protection. For example, when controlling a compressor the outflow is the primary variable but it may be necessary to switch to a different mode to avoid compressor stall, which may damage the compressor. We illustrate some ways in which logic and feedback are combined by a few examples.

Cruise control

The basic control function in a cruise controller, such as the one shown in Figure 1.15, is to keep the velocity constant. It is typically done with a PI controller. The controller normally operates in automatic mode but it is is necessary to switch it off when braking, accelerating or changing gears. The cruise control system has a human–machine interface that allows the driver to communicate with the system. There are many different ways to implement this system; one version is illustrated in Figure 1.15. The system has four buttons: on-off, set/decelerate, resume/accelerate and cancel. The operation of the system is governed by a finite state machine that controls the modes of the PI controller and the reference generator; see Figure 1.15.

The finite state machine has four modes: off, standby, cruise and hold. The state changes depending on actions of the driver who can brake, accelerate and operate using the buttons. The on/off switch moves the states between off and standby. From standby the system can be moved to cruise by pushing the resume/accelerate button. The velocity reference is set as the velocity of the car when the button is

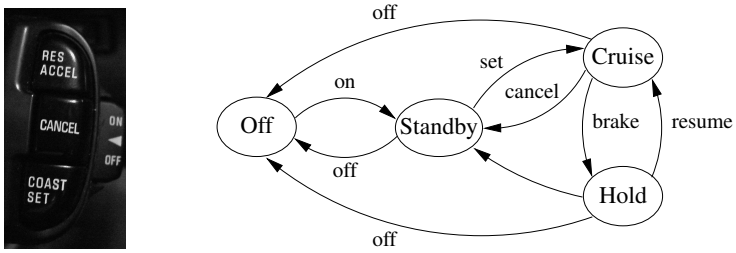


Figure 1.15: Finite state machine for cruise control system. The figure on the left shows some typical buttons used to control the system. The controller can be in one of four modes, corresponding to the nodes in the diagram on the right. Transition between the modes is controlled by pressing one of the five buttons on the cruise control interface: on, off, set, resume or cancel.

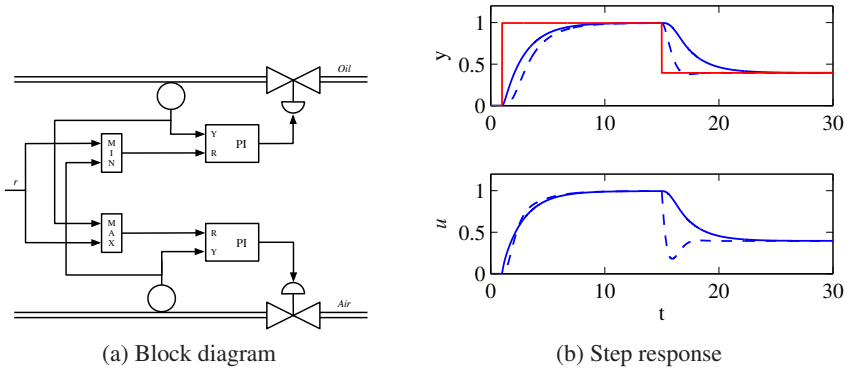
released. In the cruise state the operator can change the velocity reference; it is increased by the button resume/accelerate and decreased by the button set/coast. If the driver accelerates by pushing the gas pedal the speed increases but it will go back to the set velocity when the gas pedal is released. If the driver brakes the car brakes and the cruise controller goes into hold but it remembers the set point of the controller; it can be brought to the cruise state by pushing the res/accelerate button. The system also moves from cruise mode to standby if the cancel button is pushed. The reference for the velocity controller is remembered. The system goes into off mode by pushing off.

The PI controller is designed to have good regulation properties and to give good transient performance when switching between resume and control modes.

Server Farms

Server farms consist of a large number of computers for providing Internet services (cloud computing). Large server farms may have thousands of processors. Power consumption for driving the servers and for cooling them is a prime concern. The cost for energy can be more than 40% of the total cost for data centers, which is of the order of a million dollars per month for a large installation [EKR03]. The prime task of the server farm is to respond to a strongly varying computing demand. There are constraints given by electricity consumption and the available cooling capacity. The throughput of an individual server depends on the clock rate, which can be changed by the voltage applied to the system. Increasing the supply voltage increases the energy consumption and more cooling is required.

Control of server farms is often performed using a combination of feedback and logic. Capacity can be increased rapidly if a server is switched on simply by increasing the voltage to a server, but a server that is switched on consumes energy and requires cooling. Control of server farms is often performed by a combination of feedback and logic. To save energy it is advantageous to switch off servers that are not required, but it takes some time to switch on a new server. A control system for a server farm requires individual control of the voltage and cooling of each server and a strategy for switching servers on and off. Temperature is



(a) Block diagram (b) Step response

Figure 1.16: Air-fuel controller based on selectors. The left figure shows the system architecture. The letters r and y in the PI controller denotes the input ports for reference and measured signal respectively. The right figure shows a simulation where the power reference r (red) is changed at $t = 1$ and $t = 15$. Notice that the air flow (solid blue) is larger than the fuel flow (dashed) both for increasing and decreasing reference steps.

also important. Overheating will reduce the life time of the system and may even destroy it. The cooling system is complicated because cooling air goes through the servers in series and parallel. The measured value for the cooling system is therefore the server with the highest temperature.

Air-Fuel Control

Air-fuel control is an important problem for ship boilers. The control system consists of two loops for controlling air and oil flow and a *supervisory controller* that adjusts the air-fuel ratio. The ratio should be adjusted for optimal efficiency when the ships are on open sea but it is necessary to run the system with air excess when the ships are in the harbor, since generating black smoke will result in heavy penalties.

An elegant solution to the problem can be obtained by combining PI controllers with maximum and minimum selectors, as shown in Figure 1.16a. A *selector* is a static system with several inputs and one output: a maximum selector gives an output that is the largest of the inputs, a minimum selector gives an output that is the smallest of the inputs. Consider the situation when the power demand is increased: the reference r to the air controller is selected as the commanded power level by the maximum selector, and the reference to the oil flow controller is selected as the measured airflow. The oil flow will lag the air flow and there will be air excess. When the commanded power level is decreased, the reference of the oil flow controller is selected as the power demand by the minimum selector and the reference for the air flow controller is selected as the oil flow by the the maximum selector. The system then operates with air excess when power is decreased.

A simulation of the control logic is shown in Figure 1.16b, based on the process

model

$$\frac{dx_a}{dt} = -4x_a + 4u_a, \qquad \frac{dx_o}{dt} = -x_o + u_o,$$

where x_a and x_o are the states representing air and oil dynamics. The air dynamics are faster than the oil dynamics. The PI controllers are described by

$$u_a = -k_{pa}x_a + k_{ia}\int^t (r_a - y_a)dt, \qquad r_a = \max(r, x_0),$$

$$u_o = -k_{po}x_o + k_{io}\int^t (r_o - y_o)dt, \qquad r_o = \min(r, x_a).$$

The controller gains used in are $k_{pa} = 1$, $k_{ia} = 1$, $k_{po} = 2$ and $k_{io} = 4$.

Selectors are commonly used to implement logic in engines and power systems. They are also used for systems that require very high reliability: by introducing three sensors and only accepting values where two sensors agree it is possible to guard for the failure of a single sensor.

1.8 Control System Architectures

Most of the control systems we are investigating in this book will be relatively simple feedback loops. In this section we will try to give a glimpse of the fact that in reality the simple loops combine to form a complex system which often has an hierarchical structure with controllers, logic and optimization in different combinations. Figure 1.17 shows an example of such a hierarchy, exposing different "layers" of the control system. This class of systems is discussed in more detail in Chapter 14. We focus here on a few representative examples to illustrate some basic points.

Freight Train Trip Optimizer

An example of two of the layers represented in Figure 1.17 can be see in the control of modern locomotives developed by General Electric (GE). Typical requirements for operating a freight train is to arrive in time and to use as little fuel as possible. The key issue is to avoid unnecessary breaking. Figure 1.18 illustrates a system developed by GE. At the low layer the train has a speed regulator and a simple logic to avoid entering a zone where there is another train. The key disturbance for the speed control is the slope of the ground. The speed controller has a model of the track, a GPS sensor and an estimator. The set point for the speed controller is obtained from a trip optimizer, computes a driving plan that minimizes the fuel consumption while arriving at the desired arrival time. The arrival time is provided by a dispatch center which in turn may use some optimization. This optimization represents the second layer in Figure 1.17, with the top layer (decision-making) provided by the human operator.

Paper Mill

Figure 1.19a is a picture of a plant for making craft paper. The factory produces

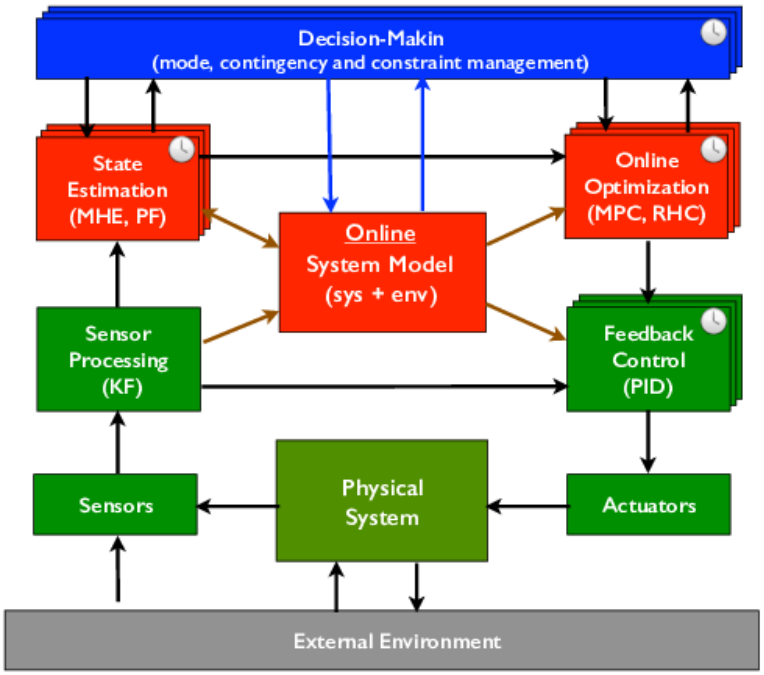
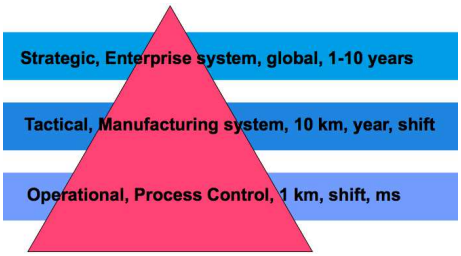


Figure 1.17: Layered decomposition of a control system.

paper for sacks and container board from logs of wood. There are three fiber lines and six paper machines. The plant has a few dozen mechanical and chemical production processes that convert the logs to a slurry of fibers in different steps and six paper machines that convert the fiber slurry to paper. There are several dozen tanks for storage of intermediate products. Each production unit has PI(D) controllers that control, flow, temperature and tank levels. The loops typically operate in a time scales from fractions of seconds to minutes. There is logic to make sure that

Figure 1.18: Freight locomotives carry massive loads of expensive diesel. GE's Trip Optimizer is a type of cruise control that combs through piles of data and synthesizes them for the driver in a way that allows him or her to steer the locomotive to maintain the most efficient speed at all times and reduce fuel burn.

(a) Paper mill in Gruvön, Sweden



(b) Enterprise control framework

Figure 1.19: A paper plant with enterprise control.
Update caption

the process is safe and there is sequencing for start, stop and production changes. The setpoints of the low level control loops are determined from production rates and recipes, sometimes using optimization. The operation of the system is governed by a supervisory system that measures tank levels and sets the production rates of the different production unit. This system performs optimization based on demanded production, measurement of tank levels and flows. The optimization is performed at the time scale of minutes to hours and it is constrained by the production rates of the different production units. At a higher level there is a system for distributing the product and for bringing in raw material using supply chain management. The manufacturing system may also be connected to the business system at an even higher level as illustrated in Figure 1.19b. There is also extensive communication because the production unit may cover an area of kilometers and the supply chains for raw material and customers a much larger range.

Autonomous Driving

The cruise controller in Figure 1.10 relieves the driver of one task to keep constant speed, but a driver still has many tasks to perform: plan the route, avoid collisions, decide the proper speed, plan the route, do lane changes, make turns, keep proper distance to the car ahead. Car manufacturers are continuously automating several of these functions going as far as automatic driving. Figure 1.20 shows a block diagram of the architecture of a typical autonomous car [BdH+07, CEHM10]. At a high level, autonomous vehicles decompose the driving problem into four basic subsystems: sensing, perception, planning and control.

The sensing subsystem is responsible for taking raw data measurements. For the vehicle, this included GPS, IMU (inertial measurement unit) and odometry measurements (or an off the shelf system that fused these together); several teams also included vision for lane and stop line detection. For perceiving the static and dynamic urban environment, measurements included laser range finders, radar and cameras. Many autonomous vehicles also segment the laser (e.g. clustering) and vision data (e.g. lane finding) in order to produce a data product of smaller size that is easier to process.

The perception subsystem is responsible for creating usable information about

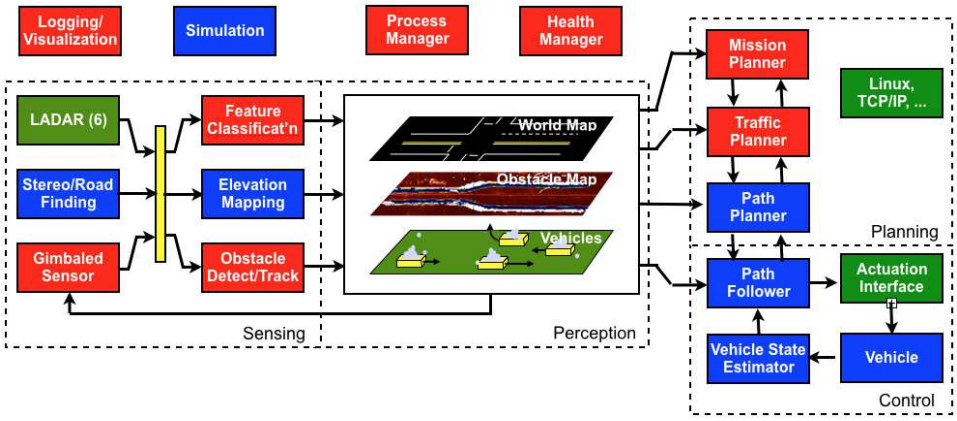


Figure 1.20: High level systems architecture for urban driving [BdH+07].

the vehicle and its environment. Vehicle estimation includes pose (inertial position, velocity, attitude, rates) as well as map relative information (e.g. the vehicle location within a lane or map); the latter typically uses vision or laser measurements to help to produce map relative estimates. Estimation of the environment can be accomplished in a number of ways, primarily because of the variations in sensors, computation and resources. For example, many vehicles use sensors mounted around the vehicle (front right, front left, side, etc.) and reason about elements such as the location, velocity, lane of other cars, and sensor occlusions.

The planning subsystem typically includes common components such as path planners, behavioral planners and route (map) planners. These vary, however, across different implementations. Some common approaches to path planning include graph search across a tree of possible trajectories and online optimization-based planners. Behavioral planners are usually built around finite-state machine logic. A key element in most planners is reasoning about the probabilistic information coming from the perception subsystem, which is typically accomplished with a finite-state machine. For special behaviors, such as operation at intersections, zones, and blockages, custom components are usually designed.

Finally, the control subsystem includes the actual actuators and commands to drive the car; information for the control law would come from some combination of the higher level planning (i.e. the proposed path), and/or direct sensing in some cases in order to increase the speed of response and avoid obstacles.

1.9 Further Reading

The material in the first half of this chapter draws from the report of the Panel on Future Directions on Control, Dynamics and Systems [Mur03]. Several additional papers and reports have highlighted the successes of control [NS99] and new vistas in control [Bro00, Kum01, Wis07]. The early development of control is described

by Mayr [May70] and in the books by Bennett [Ben79, Ben93], which cover the period 1800–1955. A fascinating examination of some of the early history of control in the United States has been written by Mindell [Min02]. A popular book that describes many control concepts across a wide range of disciplines is *Out of Control* by Kelly [Kel94].

There are many textbooks available that describe control systems in the context of specific disciplines. For engineers, the textbooks by Franklin, Powell and Emami-Naeini [FPEN05], Dorf and Bishop [DB04], Kuo and Golnaraghi [KG02] and Seborg, Edgar and Mellichamp [SEM04] are widely used. More mathematically oriented treatments of control theory include Sontag [Son98] and Lewis [Lew03]. The books by Hellerstein et al. [HDPT04] and Janert [Jan14] provide descriptions of the use of feedback control in computing systems. A number of books look at the role of dynamics and feedback in biological systems, including Milhorn [Mil66] (now out of print), J. D. Murray [Mur04] and Ellner and Guckenheimer [EG05]. The book by Fradkov [Fra07] and the tutorial article by Bechhoefer [Bec05] cover many specific topics of interest to the physics community.

Systems that combine continuous feedback with logic and sequencing are called hybrid systems [RST12]. The theory required to properly model and analyze such systems is outside the scope of this book. It is, however, very common that practical control systems combine feedback control with logic sequencing and selectors; many examples are given in [ÅH05].

Exercises

1.1 (Eye motion) Perform the following experiment and explain your results: Holding your head still, move one of your hands left and right in front of your face, following it with your eyes. Record how quickly you can move your hand before you begin to lose track of it. Now hold your hand still and shake your head left to right, once again recording how quickly you can move before losing track of your hand.

1.2 Identify five feedback systems that you encounter in your everyday environment. For each system, identify the sensing mechanism, actuation mechanism and control law. Describe the uncertainty with respect to which the feedback system provides robustness and/or the dynamics that are changed through the use of feedback.

1.3 (Balance systems) Balance yourself on one foot with your eyes closed for 15 s. Using Figure 1.3 as a guide, describe the control system responsible for keeping you from falling down. Note that the "controller" will differ from that in the diagram (unless you are an android reading this in the far future).

1.4 (Cruise control) Download the MATLAB code used to produce simulations for the cruise control system in Figure 1.10 from the companion web site. Using trial and error, change the parameters of the control law so that the overshoot in speed is not more than 1 m/s for a vehicle with mass $m = 1000$ kg.

1.5 (Integral action) We say that a system with a constant input reaches steady state if all system variables approach constant values as time increases. Show that a controller with integral action, such as those given in equations (1.4) and (1.5), gives zero error if the closed loop system reaches steady state. Notice that there is no saturation in the controller.

1.6 Search the web and pick an article in the popular press about a feedback and control system. Describe the feedback system using the terminology given in the article. In particular, identify the control system and describe (a) the underlying process or system being controlled, along with the (b) sensor, (c) actuator and (d) computational element. If the some of the information is not available in the article, indicate this and take a guess at what might have been used.

Chapter Two
Feedback Principles

Feedback - it is the fundamental principle that underlies all self-regulating systems, not only machines but also the processes of life and the tides of human affairs.

A. Tustin, "Feedback", *Scientific American*, 1952 [Tus52].

This chapter presents some examples that illustrate fundamental properties of feedback: disturbance attenuation, command signal following, robustness and shaping of behavior. Simple methods for analysis and design of low-order systems are introduced. After reading this chapter, readers should have some insight into the power of feedback, they should know about transfer functions and block diagrams, and they should be able to design simple feedback systems. The basic concepts described in this chapter are explained in more detail in the remainder of the text, and this chapter can be skipped for readers who prefer to move directly to the more detailed analysis and design techniques.

2.1 Mathematical Models

The fundamental properties of feedback will be illustrated using a collection of examples. We need a modest set of concepts and tools to analyze simple feedback systems: linear differential equations, transfer functions, block diagrams and block diagram algebra. In addition we need a simulation tool. In this section we will introduce some of these tools.

Linear Differential Equations and Transfer Functions

In many practical situations, the input/output behavior of a system can be modeled by a linear differential equation of the form

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n y = b_1 \frac{d^{n-1} u}{dt^{n-1}} + \cdots + b_n u, \tag{2.1}$$

where u is the input, y is the output and the coefficients a_k and b_k are real numbers. The differential equation (2.1) is characterized by two polynomials

$$a(s) = s^n + a_1 s^{n-1} + \cdots + a_n, \qquad b(s) = b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_n, \tag{2.2}$$

where $a(s)$ is the *characteristic polynomial* of the differential equation (2.1). We assume that the polynomials $a(s)$ and $b(s)$ do not have common roots.

Equation (2.2) represents a *time invariant* system because if the pair $u(t), y(t)$ satisfies the equation so does $u(t+\tau), y(t+\tau)$. The equation is also *linear* because

if $u_1(t), y_1(t)$ and $u_2(t), y_2(t)$ satisfy the equation so does $\alpha u_1(t) + \beta u_2(t), \alpha y_1(t) + \beta y_2(t)$, where α and β are real numbers. Systems that are linear and time invariant are often called *LTI systems*. We can visualize these systems as being characterized by a huge table of corresponding input/output pairs. An interesting property of an LTI system is that they can be characterized by a single carefully chosen pair, for example the response of the system to a step input.

The solution to equation (2.1) is the sum of two terms: the general solution to the *homogeneous equation*, which does not depend on the input, and a *particular solution*, which depends on the input. The homogeneous equation associated with equation (2.1) is

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_n y = 0. \tag{2.3}$$

Letting s_k represent the roots of the *characteristic equation* $a(s) = 0$, the solution to equation (2.3) is

$$y(t) = \sum_{k=1}^{n} C_k e^{s_k t} \tag{2.4}$$

if the characteristic does not have multiple roots s_k. The parameters $C_1, \ldots, C_n$ are constants that can be determined from the initial conditions at $t = 0$.

Since the coefficients a_k are real, the roots of the characteristic equation are either real-valued or occur in complex conjugate pairs. A real root s_k of the characteristic equation corresponds to the exponential function $e^{s_k t}$. This function decreases over time if s_k is negative, is constant if $s_k = 0$ and increases if s_k is positive, as shown in the top row of Figure 2.1. For real roots s_k the parameter $T = 1/s_k$ is called the *time constant*, because it tells how the signal decays.

A complex root $s_k = \sigma \pm i\omega$ corresponds to the time functions

$$e^{\sigma t} \sin(\omega t), \qquad e^{\sigma t} \cos(\omega t),$$

which have oscillatory behavior, as illustrated in the bottom row of Figure 2.1. The sine terms are shown as solid line; they have zero crossings with the spacing π/ω. The dashed lines show the envelopes, which correspond to the exponential function $\pm e^{\sigma t}$.

When the characteristic equation (2.4) has multiple roots, the solutions to the homogeneous equation (2.3) take the form

$$y(t) = \sum_{k=1}^{m} C_k(t) e^{s_k t}, \tag{2.5}$$

where $C_k(t)$ is a polynomial with degree less than the multiplicity of the root s_k. The solution (2.5) has $\sum_{k=1}^{m} (\deg C_k + 1) = n$ free parameters.

Having explored the solution to the homogeneous equation, we now turn to the input-dependent part of the solution. The solution to equation (2.1) for an exponential input is of particular interest, as will be shown in the following. We set $u(t) = e^{st}$, where $s \neq a_k$ is a complex number, and investigate if there is a unique particular solution of the form $y(t) = G(s)e^{st}$. Assuming this to be the case, we

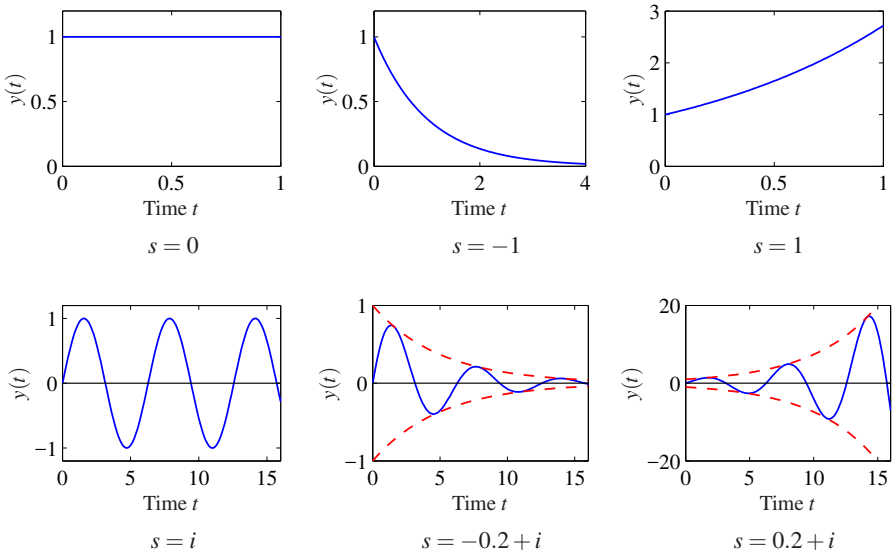


Figure 2.1: Examples of exponential signals. The top row corresponds to exponential signals with a real exponent, and the bottom row corresponds to those with complex exponents. The dashed line in the last two cases denotes the bounding envelope for the oscillatory signals. In each case, if the real part of the exponent is negative then the signal decays, while if the real part is positive then it grows.

find

$$\begin{aligned} \frac{du}{dt} &= se^{st}, & \frac{d^2u}{dt^2} &= s^2e^{st}, & \cdots \quad \frac{d^nu}{dt^n} &= s^ne^{st} \\ \frac{dy}{dt} &= sG(s)e^{st}, & \frac{d^2y}{dt^2} &= s^2G(s)e^{st}, & \cdots \quad \frac{d^ny}{dt^n} &= s^nG(s)e^{st}. \end{aligned} \tag{2.6}$$

Inserting these expressions into the differential equation (2.1) gives

$$(s^n + a_1s^{n-1} + \cdots + a_n)G(s)e^{st} = (b_1s^{n-1} + b_2s^{n-2} + \cdots + b_n)e^{st}$$

and hence

$$G(s) = \frac{b_1s^{n-1} + b_2s^{n-2} + \cdots + b_n}{s^n + a_1s^{n-1} + \cdots + a_n} = \frac{b(s)}{a(s)}. \tag{2.7}$$

This function is called the *transfer function* of the system. It describes a particular solution to the differential equation for the input e^{st}. Summarizing, we find that a solution to the differential equation (2.1) for the exponential input $u(t) = e^{st}$ is

$$y(t) = \sum_{k=1}^{m} C_k(t)e^{s_kt} + G(s)e^{st}. \tag{2.8}$$

The relation between the transfer function (2.7) and the differential equation (2.1) is clear: the transfer function (2.7) can be obtained by inspection from the differential equation (2.1), and conversely the differential equation can be obtained

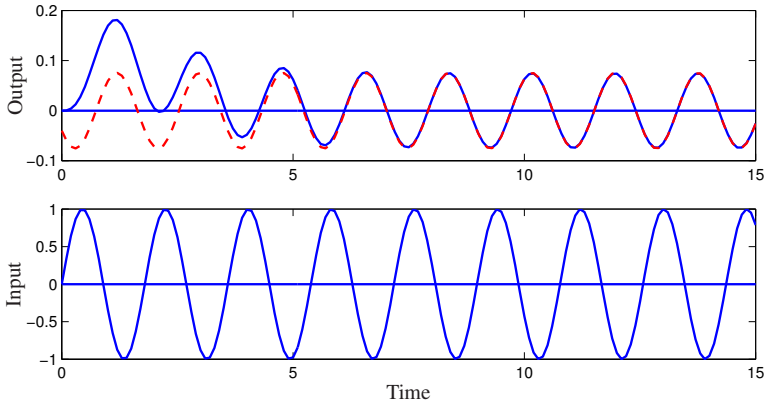


Figure 2.2: Two responses of a linear time-invariant system to a sinusoidal input. The dashed line shows the output when the initial conditions are chosen so that the output is purely sinusoidal. The full lines show the response response for the initial conditions $y(0) = 0$ and $y'(0) = 0$. The transfer function $G(s) = 1/(s+1)^2$.

from the transfer function. The transfer function can thus be regarded as a shorthand notation for the differential equation (2.1). It is a complete characterization of the differential equation even if it was derived as the response to a specific input $u(t) = e^{st}$. The input and the initial conditions must *both* be given to obtain the full solution of the differential equation, also refered to as the *response* of the system.

The particular solution for a constant input $u(t) = 1 = e^{0\cdot t}$ is $y(t) = G(0) = b_n/a_n$. The quantity $G(0)$ is called the *zero frequency gain*, the *static gain* or sometimes the *steady state gain*.

To deal with oscillatory signals, like those shown in the bottom row of Figure 2.1, we allow s to be a complex number. The transfer function is then a function $G : \mathbb{C} \to \mathbb{C}$ that maps complex numbers to complex numbers. We let arg represent the argument (phase, angle) of a complex number and $|\cdot|$ the magnitude, and note that the complex response to an input $u = e^{i\omega t} = \cos\omega t + i\sin\omega t$ is given by $G(i\omega t)$. Using just the imaginary parts of the signals, it follows that the particular solution for the input $u = \sin(\omega t) = \operatorname{Im} e^{i\omega t}$ is

$$\begin{aligned} y(t) &= \operatorname{Im}\left(G(i\omega)\, e^{i\omega t}\right) = \operatorname{Im}\left(|G(i\omega)|\, e^{i\arg G(i\omega)}\, e^{i\omega t}\right) \\ &= |G(i\omega)|\operatorname{Im} e^{i(\arg G(i\omega)+\omega t)} = |G(i\omega)|\sin(\omega t + \arg G(i\omega)). \end{aligned}$$

The input is thus amplified by $|G(i\omega)|$ and the phase shift between input and output is $\arg G(i\omega)$. The functions $G(i\omega)$, $|G(i\omega)|$ and $\arg G(i\omega)$ are called the *frequency response*, *gain* and *phase*. Gain and phase are also called *magnitude* and *angle*.

The full response to a sine function is the sum of a particular solution and a solution to the homogeneous equation that is determined by the initial conditions, as given in equation (2.8). An illustration is given in Figure 2.2. The dashed line, which is a pure sine wave, is the solution obtained when all C_k in equation (2.8) are zero. The full line shows the response obtained when the C_k are chosen so that

$y(0)$ and its derivatives $y^{(k)}(0)$, $k = 1, \ldots, n-1$ are all zero. Since all roots of the characteristic equation have negative real parts, the solution to the homogeneous equation (2.5) goes to zero as $t \to \infty$ and the general solution converges to the particular solution.

The transfer function has many physical interpretations that can be exploited for analysis and design. The roots s_k of the characteristic equation $a(s) = 0$ are called *poles* of the transfer function: the transfer function is infinite for $s = s_k$. The poles s_k appear as exponents in the general solution to the homogeneous equation, as seen in equations (2.4) and (2.5). Systems with poles that are "lightly damped" ($\mathrm{Re}(s_k)$ is negative but small) can exhibit resonances when a sinusoidal input is applied whose frequency is near the imaginary part of s_k.

The roots s_j of the polynomial $b(s)$ are called *zeros* of the transfer function. The reason is that if $b(s_j) = 0$ it follows that $G(s_j) = 0$, and the particular solution for the input $e^{s_k t}$ is then zero. A system theoretic interpretation is that the transmission of the exponential signal $e^{s_j t}$ is blocked by the zero $s = s_j$, which is therefore also called a *transmission zero*.

The transfer function can also convey a great deal of intuition: the approximations of $G(s)$ for small and large s capture the propagation of slow and fast signals respectively. Consider for example a spring-mass system with input u (force) and output q (position), whose dynamics satisfy the second-order differential equation

$$m\ddot{q} + c\dot{x} + ky = u.$$

The system has the transfer function

$$G(s) = \frac{1}{ms^2 + cs + k}.$$

For small s we have $G(s) \approx 1/k$. The corresponding input/output relation is $q = (1/k)u$, which implies that for low frequency inputs, the system behaves like a spring driven by a force. For large s we have $G(s) \approx 1/(ms^2)$. The corresponding differential equation is $m\ddot{q} = u$ and thus high frequency inputs the system thus behaves like mass driven by a force (a double integrator).

The frequency response of a stable system can be determined experimentally by exploring the steady state response of a system to sinusoidal signals. This is an alternative or a complement to physical modeling. A more elaborate treatment of transfer functions and the frequency response will be given in Chapter 9.

Stability: The Routh-Hurwitz Criterion

When using feedback there is always the danger that the system may become unstable, and it is therefore important to have a stability criterion. The differential equation (2.1) is called *stable* if all solutions of the homogeneous equation (2.3) go to zero for any initial condition. It follows from equation (2.5) that this requires that all the roots of the characteristic equation

$$a(s) = s^n + a_1 s^{n-1} + \cdots + a_n = 0$$

have negative real parts.

It can often be difficult to analytically compute the roots of a high-order polynomial. The *Routh-Hurwitz criterion* is a stability criterion that does not require explicit calculation of the roots, because it gives conditions in terms of the coefficients of the characteristic polynomial.

We illustrate the Routh-Hurwitz criterion by describing it for low-order differential equations. A first-order differential equation is stable when the coefficient a_1 of the characteristic polynomial is positive, since the root of the characteristic polynomial will be $s = -a_1 < 0$. A second-order polynomial has the roots

$$s = \frac{1}{2}\left(-a_1 \pm \sqrt{a_1^2 - 4a_2}\right),$$

and it is easy to verify that the real parts of the roots are both negative if and only if $a_1 > 0$ and $a_2 > 0$. A third order differential equation is more complicated, but the roots can be shown to have negative real parts if and only if

$$a_1, a_2, a_3 > 0, \quad \text{and} \quad a_1 a_2 > a_3. \tag{2.9}$$

The corresponding conditions for a fourth order differential equation are

$$a_1, a_2, a_3, a_4 > 0, \qquad a_1 a_2 > a_3, \quad \text{and} \quad a_1 a_2 a_3 > a_1^2 a_4 + a_3^2. \tag{2.10}$$

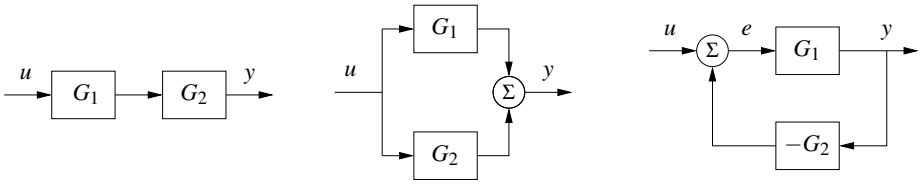
The Routh-Hurwitz criterion [Gan60] gives similar conditions for arbitrarily high order polynomials. Stability of a linear differential equation can thus be investigated just by analyzing the signs of various combinations of the coefficients of the characteristic polynomial.

Block Diagrams and Transfer Functions

As we saw already in Chapter 1, control systems are often described using block diagrams, such as the ones shown in Figures 1.1 and 1.3. If the behavior of the blocks are represented by transfer functions, the transfer function of a system can be obtained simply by algebraic manipulations. It follows from equation (2.8) that the transfer function can be derived from the particular solution for the input e^{st}. To derive the transfer function for a system composed of several blocks we assume that the input signal is an exponential $u(t) = e^{st}$, and compute the corresponding particular solutions for all blocks.

Consider for example the system in Figure 2.3a, which is a series connection of two systems with the transfer functions $G_1(s)$ and $G_2(s)$. Let the input of the system be $u(t) = e^{st}$ The output of the first block is then $y_1(t) = G_1(s)e^{st}$, which is also an exponential, and the output of the second system is $y(t) = G_2(s)G_1(s)u(t)$. The transfer function of the system is thus $G_{yu}(s) = G_2(s)G_1(s)$, where we use the convention that the right subscript is the input and the left subscript is the output, so that $y = G_{yu}u$.

Next we will will consider parallel connections of systems as shown in Figure 2.3b. Assuming that the input is $u(t) = e^{st}$, the exponential outputs of the blocks



(a) $G_{yu}(s) = G_2(s)G_1(s)$ (b) $G_{yu}(s) = G_1(s) + G_2(s)$ (c) $G_{yu}(s) = \dfrac{G_1(s)}{1 + G_1(s)G_2(s)}$

Figure 2.3: Interconnections of linear systems. Series (a), parallel (b) and feedback (c) connections are shown. The transfer functions for the composite systems can be derived by algebraic manipulations assuming exponential functions for all signals.

are $y_1(t) = G_1(s)e^{st}$ and $y_2(t) = G_2(s)e^{st}$. The output of the system is then

$$y(t) = \big(G_1(s)e^{st} + G_2(s)e^{st}\big) = \big(G_1(s) + G_2(s)\big)\, e^{st},$$

and the transfer function of a parallel connection of systems with the transfer functions $G_(s)$ and $G_2(s)$ is thus $G_{yu}(s) = G_1(s) + G_2(s)$.

Finally we will consider the feedback connection shown in Figure 2.3c. If the input $u(t) = e^{st}$ is an exponential we find

$$y(t) = G_1(s)e(t) = G_1(s)\big(u(t) - G_2(s)y(t)\big) = G_1(s)\big(e^{st} - G_2(s)y(t)\big).$$

Solving for $y(t)$ gives

$$y(t) = \frac{G_1(s)}{1 + G_1(s)G_2(s)}\, e^{st} = G_{yu}(s)\, e^{st}.$$

The transfer function of a feedback connection of systems with the transfer functions $G_1(s)$ and $G_2(s)$ is thus

$$G_{yu}(s) = \frac{G_1(s)}{G_1(s) + G_2(s)}. \tag{2.11}$$

By using polynomials and transfer functions the relations between signals in a feedback system can thus be obtained by algebra. With some practice the transfer functions can often be obtained by inspection, as we explore in more detail in Chapter 9.

2.2 Using Feedback to Improve Disturbance Attenuation

Reducing the effects of disturbances is a primary use of feedback. It was used by James Watt to make steam engines run at constant speed in spite of varying load and by electrical engineers to make generators driven by water turbines deliver electricity with constant frequency and voltage. Feedback is commonly used to alleviate effects of disturbances in the process industry, for machine tools and for engine and cruise control in cars. The human body exploits feedback to keep body

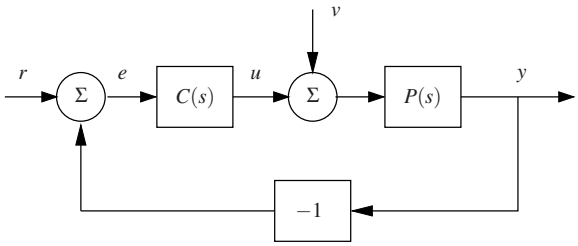


Figure 2.4: Block diagram of simple feedback system. The controller transfer function is $C(s)$ and the process transfer function is $P(s)$. The process output is y, the external signals are the reference r and the load disturbance v.

temperature, blood pressure and other important variables constant. For example the pupillary reflex guarantees that the light intensity of the retina is reasonably constant in spite of large variations in the ambient light intensity. Keeping variables close to a desired, constant reference value in spite of disturbances is called a *regulation problem*.

To discuss disturbance attenuation we consider the system in Figure 2.4. Since we will focus on the effects of a load disturbance v we will assume for now that the reference r is zero. To derive the transfer functions from the disturbance input v to the process output y, which we write as G_{yv}, we assume that the disturbance is an exponential function $v = e^{st}$. Applying block diagram algebra to Figure 2.4 gives

$$y(t) = P(s)\Big(e^{st} - C(s)P(s)y(t)\Big), \qquad y(t) = \frac{P(s)}{1+P(s)C(s)}\,e^{st}.$$

The transfer function relating the output y to the load disturbance v is thus

$$G_{yv}(s) = \frac{P(s)}{1+P(s)C(s)} \tag{2.12}$$

To explore the use of feedback to improve disturbance attentuation, we will focus on a simple process modeled by the first order differential equation

$$\frac{dy}{dt} + ay = bu, \qquad a > 0, \quad b > 0.$$

The corresponding transfer function is

$$P(s) = \frac{b}{s+a}. \tag{2.13}$$

This model is a reasonable approximation for a physical process if the storage of mass, momentum or energy can be captured by a single state variable. Typical examples are the velocity of a car on a road, the angular velocity of a rotating system and the fluid level of a tank.

Proportional Control

We will first investigate the case of proportional control, when the control signal is proportional to the output error: $u = k_p e$, as introduced already in Section 1.6. The controller transfer function is then $C(s) = k_p$. The process transfer function is given by equation (2.13) and the effect of the disturbance on the output is then described by the transfer function (2.12):

$$G_{yv}(s) = \frac{P(s)}{1+P(s)C(s)} = \frac{\frac{b}{s+a}}{1+\frac{bk_p}{s+a}} = \frac{b}{s+(a+bk_p)}.$$

The relation between the disturbance v and the output y is thus given by the differential equation

$$\frac{dy}{dt} + (a+bk_p)y = bv.$$

The closed loop system is stable if $a+bk_p > 0$. A constant disturbance $v = v_0$ then gives an output that exponentially approaches the value

$$y_0 = G_{yv}(0) = \frac{b}{a+bk_p} v_0$$

with the time constant $T = 1/(a+bk_p)$. Without feedback, $k_p = 0$ and for a constant disturbance v_0, the output will instead approach bv_0/a. The effect of the disturbance is thus reduced if $k_p > 0$.

We have thus shown that a constant disturbance gives an error that can be reduced by feedback using a proportional controller. The error decreases with increasing controller gain. Figure 2.5a shows the responses for a few values of the controller gain k_p.

Proportional-Integral (PI) Control

The PI controller, introduced in Section 1.6, is described by

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau. \tag{2.14}$$

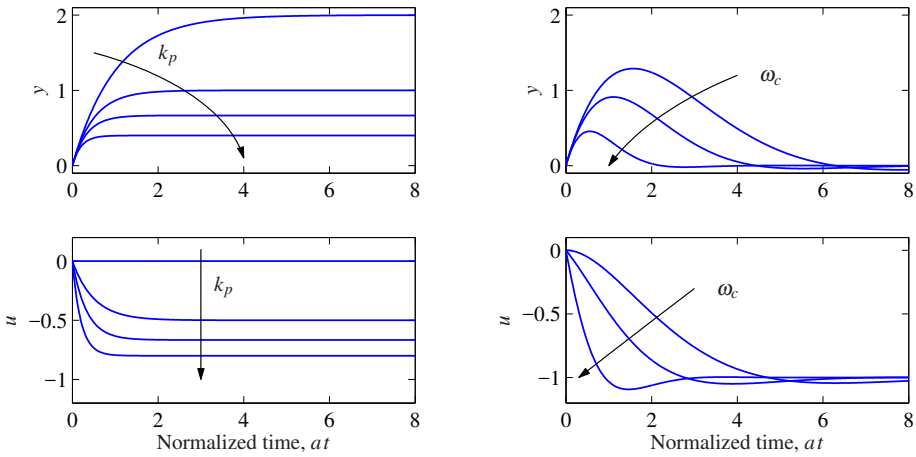
To determine the transfer function of the controller we differentiate to obtain

$$\frac{du}{dt} = k_p \frac{de}{dt} + k_i e,$$

and we find that the transfer function is $C(s) = k_p + k_i/s$. To investigate the effect of the disturbance v on the output we use the block diagram in Figure 2.4 and the transfer function from v to y is

$$G_{yv}(s) = \frac{P(s)}{1+P(s)C(s)} = \frac{bs}{s^2+(a+bk_p)s+bk_i}. \tag{2.15}$$

Using the relationship between transfer functions and differential equations given by equations (2.1) and (2.7), it follows that the relation between the load distur-



(a) Proportional control (b) Proportional-Integral (PI) control

Figure 2.5: Step responses for a first-order, closed loop system with proportional control (a) and PI control (b). The process transfer function is $P = 2/(s+1)$. The controller gains for proportional control are $k_p = 0$, 0.5, 1 and 2. The PI controller is designed using equation (2.19) with $\zeta_c = 0.707$ and $\omega_c = 0.707$, 1 and 2, which gives the controller parameters $k_p = 0$, 0.207, 0.914 and $k_i = 0.25$, 0.50 and 2.

bance and the output is given by the differential equation

$$\frac{d^2y}{dt^2} + (a + bk_p)\frac{dy}{dt} + bk_i y = b\frac{dv}{dt}. \tag{2.16}$$

Notice that since the disturbance enters as a derivative on the right hand side, a constant disturbance gives no steady state error. The same conclusion can be drawn from the observation that $G_{yv}(0) = 0$. This is consistent with the discussion of integral action and steady state error in Section 1.6.

To find suitable values of the controller parameters k_p and k_i, we consider the characteristic polynomial of the differential equation (2.16),

$$a_{cl}(s) = s^2 + (a + bk_p)s + bk_i. \tag{2.17}$$

We can assign arbitrary roots to the characteristic polynomial by choosing the controller gains k_p and k_i. The most common case is that we assign complex roots that give the characteristic polynomial

$$(s + \sigma_d + i\,\omega_d)(s + \sigma_d - i\,\omega_d) = s^2 + 2\sigma_d s + \sigma_d^2 + \omega_d^2. \tag{2.18}$$

This polynomial has roots at $s = -\sigma_d \pm i\,\omega_d$. The general solution to the homogeneous equation is then a linear combination of the terms

$$e^{-\sigma_d t}\sin(\omega_d t), \qquad e^{-\sigma_d t}\cos(\omega_d t),$$

which are damped sine and cosine functions, as shown in the lower middle plot in Figure 2.1. The coefficient σ_d determines the decay rate and the parameter ω_d gives the frequency of the decaying oscillation. Identifying coefficients of equal

powers of s in the polynomials (2.17) and (2.18) gives

$$k_p = \frac{2\sigma_d - a}{b}, \qquad k_i = \frac{\sigma_d^2 + \omega_d^2}{b}. \tag{2.19}$$

Instead of parameterizing the closed loop system in terms of σ_d and ω_d it is common practice to use the *undamped natural frequency* $\omega_c = \sqrt{\sigma_d^2 + \omega_d^2}$ and the *damping ratio* $\zeta_c = \sigma_d/\omega_c$. The closed loop characteristic polynomial is then

$$a_{cl}(s) = s^2 + 2\sigma_d s + \sigma_d^2 + \omega_d^2 = s^2 + 2\zeta_c\omega_c s + \omega_c^2.$$

This parameterization has the advantage that ζ_c, which is in the range $[-1,1]$, determines the shape of the response and ω_c gives the response speed.

Figure 2.5b shows the output y and the control signal u for $\zeta_c = 1/\sqrt{2} = 0.707$ and different values of the design parameter ω_c. Proportional control gives a steady-state error that decreases with increasing controller gain k_p. With PI control the steady-state error is zero. Both the decay rate and the peak error decrease when the design parameter ω_c is increased. Larger controller gains give smaller errors and control signals that react faster to the disturbance.

With the controller parameters (2.19), the transfer function (2.15) from disturbance v to process output y becomes

$$G_{yv}(s) = \frac{P(s)}{1 + P(s)C(s)} = \frac{bs}{s^2 + 2\zeta_c\omega_c s + \omega_c^2}.$$

For efficient attenuation of disturbances, it is desirable that $|G_{yv}(i\omega)|$ is small for all ω. For small values of ω we have $|G_{yv}(i\omega)| \approx b\omega/\omega_c$, while for large ω we have $|G_{yv}(i\omega)| \approx b/\omega$. The largest value of $|G_{yv}(i\omega)|$ is $b/(2\zeta_c\omega_c)$ for $\omega = \omega_c$. It thus follows that a large value of ω_c gives good load disturbance attenuation.

In summary, we find that the analysis gives a simple way to find the parameters of PI controllers for processes whose dynamics can be approximated by a first-order system. The technique can be generalized to more complicated systems but the controller will be more complex. To achieve the benefits of large control gains the model must be accurate over wide frequency ranges, as will be discussed next.

Unmodeled Dynamics

The analysis we have made so far indicates that there are no limits to the performance that can be achieved. Figure 2.5b shows that arbitrarily fast response can be obtained simply by making ω_c sufficiently large. In reality there are of course limitations on what is achievable. One reason is that the controller gains increase with ω_c: the proportional gain is $k_p = (2\zeta_c\omega_c - a)/b$ and the integral gain is $k_i = \omega_c^2/b$. A large value of ω_c thus gives large controller gains and the control signal may saturate. Another reason is that the model (2.13) is a simplification: it is only valid in a given frequency range. If the model is instead

$$P(s) = \frac{b}{(s+a)(1+sT)}, \tag{2.20}$$

where the term $1+sT$ represents the dynamics of sensors, actuators or other dynamics that were neglected when deriving equation (2.13)—so-called *unmodeled dynamics*—the closed loop characteristic polynomial for the closed loop system becomes

$$a_{cl} = s(s+a)(1+sT)+k_p s+k_i = s^3T+s^2(1+aT)+2\zeta_c\omega_c s+\omega_c^2.$$

It follows from the Routh-Hurwitz criterion (2.9) that the closed loop system is stable if $\omega_c^2 T < 2\zeta_c\omega_c(1+aT)$ or if

$$\omega_c T < 2\zeta_c(1+aT).$$

The frequency ω_c and the achievable response time are thus limited by the unmodeled dynamics represented by T, which typically is smaller than the time constant $1/a$ of the process. When models are developed for control it is therefore important to also consider the unmodeled dynamics.

The fact that unmodeled dynamics limit the performance of a feedback system is an important property and must be considered during the system design. It is common to use simplified models when designing components of complex systems and if the unmodeled dynamics of those components (or the other subsystems they interact with) are not properly taken into account, the implementation of the system can display poor behavior (of which instability is one extreme example). As we shall see in later chapters, it is the ability to reason about the effects of uncertainty that makes control theory a particularly powerful mathematical tool for systems design.

2.3 Using Feedback to Follow Command Signals

Another major application of feedback is to make a system output follow a reference value, which is called the *servo problem*. Cruise control, steering of a car, tracking a satellite with an antenna or a star with a telescope are some examples. Other examples are high performance audio amplifiers, machine tools and industrial robots.

To illustrate command signal following we will consider the system in Figure 2.4 where the process is a first-order system and the controller is a PI controller. The transfer functions of the process and the controller are

$$P(s) = \frac{b}{s+a}, \qquad C(s) = \frac{k_p s+k_i}{s}. \tag{2.21}$$

Since we will focus on following the command signal r, we will neglect the load disturbance and set $v=0$. Applying block diagram algebra to the system in Figure 2.4, we find that the transfer function from the command signal r to the output y is

$$G_{yr}(s) = \frac{P(s)C(s)}{1+P(s)C(s)} = \frac{bk_p s+bk_i}{s^2+(a+bk_p)s+bk_i}. \tag{2.22}$$

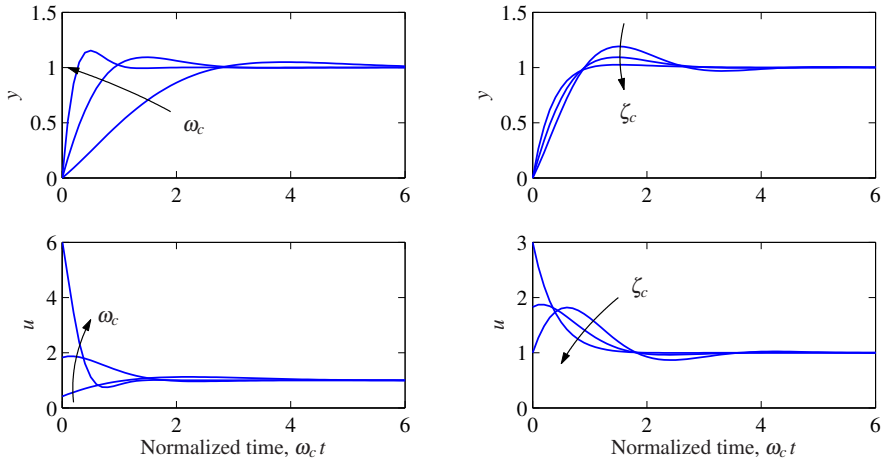


Figure 2.6: Responses to a unit step change in the command signal for different values of the design parameters ω_c and ζ_c. The left figure shows responses for fixed $\zeta_c = 0.707$ and $\omega_c = 1$, 2 and 5. The right figure shows responses for $\omega_c = 2$ and $\zeta_c = 0.5$, 0.707, and 1. The process parameters are $a = b = 1$. The initial value of the control signal is k_p.

Since $G_{yr}(0) = 1$ it follows that $r = y$ when r and y are constant, independent of the values of the parameters a and b, as long as the closed loop is stable. The steady state output is thus equal to the reference, a consequence of the integral action in the controller.

To determine suitable values of the controller parameters k_p and k_i, we proceed as in Section 2.2 by choosing controller parameters that make the closed-loop characteristic polynomial

$$a_{cl}(s) = s^2 + (a + bk_p)s + bk_i \tag{2.23}$$

equal to $s^2 + 2\zeta_c\omega_c s + \omega_c^2$ with $\zeta_c > 0$ and $\omega_c > 0$. Identifying coefficients of equal powers of s in these polynomials gives

$$k_p = \frac{2\zeta_c\omega_c - a}{b}, \qquad k_i = \frac{\omega_c^2}{b}, \tag{2.24}$$

which is equivalent to equation (2.19). Notice that integral gain increases with the square of ω_c. Figure 2.6 shows the output signal y and the control signal u for different values of the design parameters ζ_c and ω_c. The response time decreases with increasing ω_c and the initial value of the control signal also increases because it takes more effort to move rapidly. The overshoot decreases with increasing ζ_c. For $\omega_c = 2$, the design choice $\zeta_c = 1$ gives a short settling time and a response without overshoot.

It is desirable that the output y will track the reference signal r for time-varying references. This means that we would like the transfer function $G_{yr}(s)$ to be close to 1 for large frequency ranges. With the controller parameters (2.24), it follows

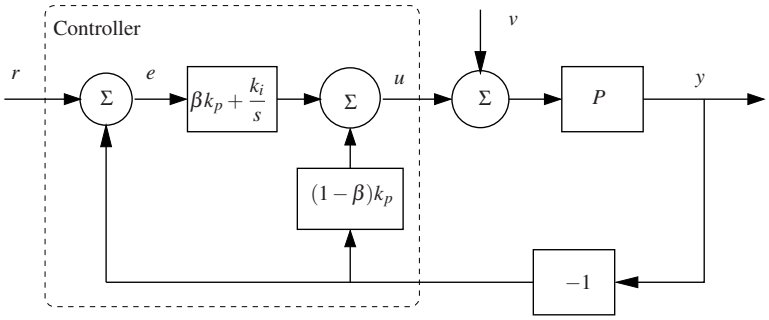


Figure 2.7: Block diagram of a closed-loop system with a PI controller having an architecture with two degrees of freedom.

from equation (2.22) that

$$G_{yr}(s) = \frac{P(s)C(s)}{1+P(s)C(s)} = \frac{(2\zeta_c\omega_c - a)s + \omega_c^2}{s^2 + 2\zeta_c\omega_c s + \omega_c^2}.$$

Since $G_{yr}(0) = 1$, tracking of constant inputs is perfect. In addition, if $s = i\omega$ is smaller in magnitude than ω_c, then we see that $G_{yr}(s)$ will be very close to one. The frequency ω_c thus determines the upper bound of the frequency of input signals that can be tracked with small error, and this bound is referred to as the *bandwidth* of the closed loop system. The frequency response of G_{yr} thus provides a quantitative representation of the tracking abilities.

Controllers with Two Degrees of Freedom

The control law in Figure 2.4 has *error feedback* because the control signal u is generated from the error $e = r - y$. With proportional control, a step in the reference signal r gives an immediate step change in the control signal u. This rapid reaction can be advantageous, but it may give large overshoot, which can be avoided by a replacing the PI controller in equation (2.14) with a controller of the form

$$u(t) = k_p\big(\beta r(t) - y(t)\big) + k_i \int_0^t (r(\tau) - y(\tau))d\tau. \tag{2.25}$$

In this modified PI algorithm, the proportional action only acts on the fraction β of the reference signal. The signal transmissions from reference r to u and from output y to u can be represented by the transfer functions

$$C_{ur}(s) = \beta k_p + \frac{k_i}{s}, \qquad C_{uy}(s) = k_p + \frac{k_i}{s} = C(s). \tag{2.26}$$

The controller (2.25) is called a controller with *two degrees of freedom* since the transfer functions $C_{ur}(s)$ and $C_{uy}(s)$ are different.

A block diagram of a closed loop system with a PI controller having two degrees of freedom is shown in Figure 2.7. Let the process transfer function be $P(s) = b/(s+a)$. The transfer functions from reference r and disturbance v to

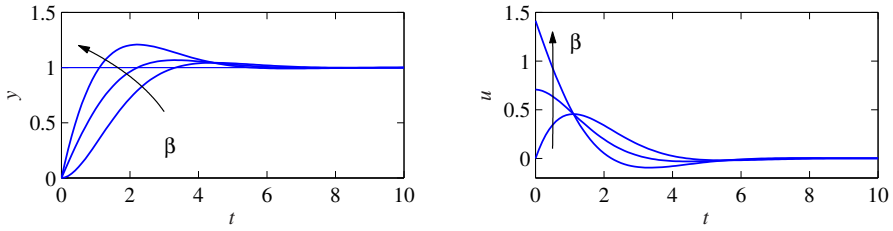


Figure 2.8: Response to a step change in the command signal for a system with a PI controller having two degrees of freedom. The process transfer function is $P(s) = 1/s$ and the controller gains are $k_p = 1.414$, $k_i = 1$ and $\beta = 0$, 0.5 and 1.

output y are

$$G_{yr}(s) = \frac{b\beta k_p s + bk_i}{s^2 + (a + bk_p)s + bk_i}, \qquad G_{yv}(s) = \frac{bs}{s^2 + (a + bk_p)s + bk_i}. \tag{2.27}$$

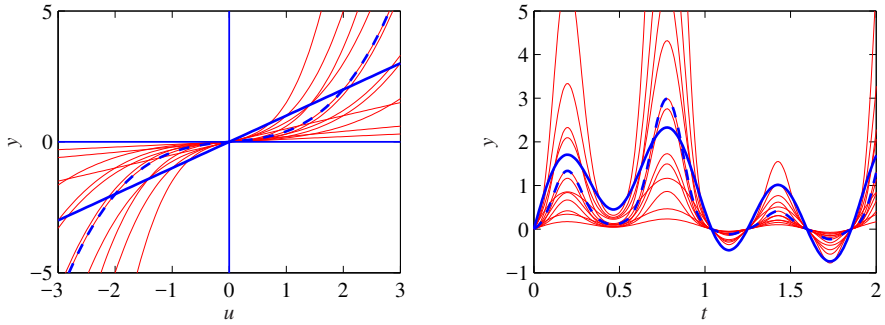
Comparing with the corresponding transfer function for a controller with error feedback in equations (2.15) and (2.22) we find that the responses to the load disturbances is the same but the response to reference signals is different.

A simulation of the closed loop system for $a = 0$ and $b = 1$ is shown in Figure 2.8. The figure shows that the parameter β has a significant effect on the responses. Comparing the system with error feedback ($\beta = 1$) to the system with smaller values of β we find that using a system with two degrees of freedom gives less overshoot and gentler control actions.

The example shows that command signal response can be improved by using a controller architecture having two degrees of freedom. In Section 2.6 we will further show that the responses to command signals and disturbances can be completely separated by using a more general system architecture. To use a system with two degrees of freedom both the reference signal r and the output signal y must be measured. There are situations where only the error signal $e = r - y$ can be measured; typical examples are DVD players, optical memories and atomic force microscopes.

2.4 Using Feedback to Provide Robustness

Feedback can be used to make good systems from poor components. Black's invention of the feedback amplifier for the telephone network is an early example [Bla77]. The vacuum tube, which was nonlinear and time varying, was the primary amplifier at the time, and Black's idea was to close a loop with negative feedback around the tube amplifier. In this way he could obtain a closed loop system with a linear input/output relation having constant gain. The general recipe is to localize the nonlinearities and the source of process variations, and to close feedback loops around them. We illustrate the approach with a simple model of an electronic amplifier.



(a) Input/output relationship (b) Output response

Figure 2.9: Responses of a static nonlinear system. The left figure shows the input/output relation of the open-loop system and the right figure shows responses to the input signal (2.29). The ideal response is shown solid bold lines. The nominal response of the nonlinear system is shown using dashed bold lines and the responses for different parameter values are shown using thin lines.

Reducing Effects of Parameter Variations and Nonlinearities

Consider an amplifier with a static, nonlinear input/output relation with considerable variability, as illustrated in Figure 2.9a. The nominal input/output characteristics is shown as a dashed bold line and examples of variations as thin lines. The nonlinearity in the figure is given by

$$y = f(u) = \alpha(u + \beta u^3), \qquad -3 \le u \le 3. \tag{2.28}$$

The nominal values corresponding to the dashed line are $\alpha = 0.2$ and $\beta = 1$. The variations of the parameters α and β are in the ranges $0.1 \le \alpha \le 0.5$, $0 \le \beta \le 2$. The responses of the system to the input $u = r$ with

$$r(t) = \sin(t) + \sin(\pi t) + \sin(\pi^2 t). \tag{2.29}$$

are shown in Figure 2.9b. The desired response $y = u$ is shown as a solid bold line and responses for a range of parameters are shown with thin lines. The nominal response of the nonlinear system is shown as a dashed bold line. It is distorted due to the nonlinearity. Notice in particular the heavy distortion both for small and large signal amplitudes.

The behavior of the system is clearly not satisfactory, but it can be improved significantly by introducing feedback. A block diagram of a system with a simple integral controller is shown in Figure 2.10. Figure 2.11 shows the behavior of the closed loop system with the same parameter variations as in Figure 2.9. The input/output plot in Figure 2.11a is a scatter plot of the inputs and the outputs of the feedback system. The input/output relation is practically linear and close to the desired response. There is some variability because of the dynamics introduced by the feedback. Figure 2.11b shows the responses to the reference signal; notice the dramatic improvement compared with Figure 2.9b. The tracking error is shown in

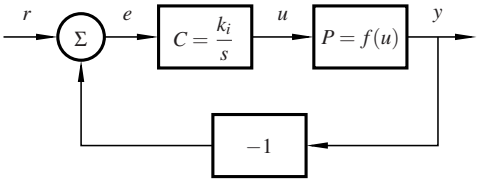


Figure 2.10: Block diagram of a nonlinear system with integral feedback.

Figure 2.11c.

Analysis

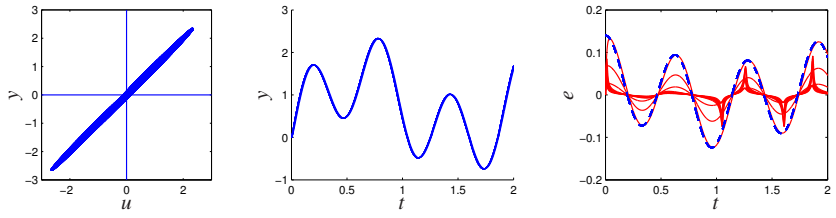
Analysis of the closed loop system is difficult because it is nonlinear. We can, however, obtain significant insight by using approximations. We first observe that the system is linear when $\beta = 0$. In other situations we can thus approximate the nonlinear function by a straight line around an operating point $u = u_0$. The slope of the nonlinear function at $u = u_0$ is $f'(u_0)$ and we will approximate the process with a linear system with the gain $f'(u_0)$. The transfer functions of the process and the controller are

$$P(s) = f'(u_0) = \alpha(1 + 3\beta u_0^2) = b, \qquad C(s) = \frac{k_i}{s}, \tag{2.30}$$

where u_0 denotes the operating condition. The process gain $b = \alpha(1 + 3\beta u_0^2)$ is in the range 0.1–27.5 depending on the values of α, β and u_0. It follows from equation (2.12) that the transfer functions relating the output y and the error e to the reference signal r are

$$G_{yr}(s) = \frac{bk_i}{s + bk_i}, \qquad G_{er}(s) = 1 - G_{yr} = \frac{s}{s + bk_i}. \tag{2.31}$$

The closed loop system is a first-order system with the pole $s = -bk_i$ and time constant $T_{cl} = 1/(bk_i)$. If the integral gain is chosen as $k_i = 1000$, the closed loop



(a) (b) (c)

Figure 2.11: Responses of the system with integral feedback ($k_i = 1000$). The left plot, is a scatter plot of inputs and outputs. The center plot shows the response of the closed loop system to the input signal r, and the right plot shows the control error. The parameter variations are the same as in Figure 2.9. Notice the dramatic improvement compared to Figure 2.9b. The dashed line in (c) corresponds to the approximate error given by equation (2.33).

pole ranges from 100 rad/s to 2.75×10^4 rad/s, which is fast compared to the high frequency component 9.86 rad/s of the input signal.

The error for the approximated system is described by the differential equation

$$\frac{de}{dt} = -bk_i e + \frac{dr}{dt}, \qquad \frac{dr}{dt} = \cos(t) + \pi\cos(\pi t) + \pi^2\cos(\pi^2 t). \tag{2.32}$$

The fast frequency component of the input (2.29) has the frequency $\pi^2 = 9.86$; it is slower than the process dynamics for all parameter variations. Neglecting the term de/dt in equation (2.32) gives

$$e \approx \frac{1}{bk_i}\frac{dr}{dt} \approx \frac{\pi^2}{bk_i}\cos(\pi^2 t). \tag{2.33}$$

An estimate of the largest error $e(t) \approx 0.1\cos(\pi^2 t)$ is obtained for the smallest value of $b = 0.1$. It is shown in dashed line in Figure 2.11c, and we see that it gives a good estimate of the maximum error across the uncertain parameter space.

This analysis is based on the assumption that the amplifier can be modeled by a constant gain. The closed loop system is however a dynamic system because the controller is an integrator. It follows from equation (2.31) that the closed loop dynamics has the time constant $T_{cl} = 1/(bk_i)$. If the amplifier has dynamics, its time constant must thus be small compared to T_{cl} in order to provide good tracking. It follows that the largest admissible integral gain k_i is determined by the unmodeled dynamics.

This example illustrates that feedback can be used to design an amplifier that has practically linear input/output relation even if the basic amplifier is nonlinear with strongly varying characteristics.

2.5 Positive Feedback

Most of this book is focused on negative feedback because of its amazingly good properties, which have been illustrated in the previous sections. In this section we will briefly discuss positive feedback, which has complementary properties. In spite of this, positive feedback has found good use in several contexts.

Systems with negative feedback can be well understood by linear analysis. To understand systems with positive feedback it is necessary to consider nonlinear effects, because without the nonlinearities the instability caused by positive feedback will grow without bound. The nonlinear elements can create interesting and useful effects by limiting the signals.

Positive feedback is common in many settings: encouraging a student or a coworker when they have performed well encourages them do to even better. In biology, it is standard to distinguish inhibitory connections (negative feedback) from excitatory feedback (positive feedback) as illustrated in Figure 2.12. Neurons use a combination of positive and negative feedback to generate spikes.

Positive feedback may cause instabilities. Exponential growth, where the rate

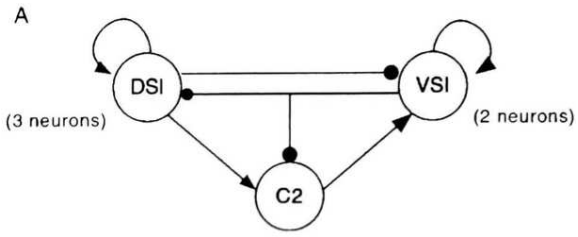


Figure 2.12: Schematic diagram of the neural network that controls swimming motions in the marine mollusk Tritonia, which has both positive and negative feedback [Wil99]. An exhibitory connection (positive feedback) is denoted with a line ending with an arrow, an inhibitory interaction (negative feedback) is denoted with an arrow ending with a circle. (Figure adapted from [Wil99].)

of change of a quantity x is proportional to x,

$$\frac{dx}{dt} = \alpha x,$$

is a typical example, which results in exponential growth $x(t) = e^{\alpha t}$. In nature, exponential growth of a species is limited by the finite amount of food. Another common example is when a microphone is placed close to a speaker in public address systems, resulting in a howling noise. Positive feedback can create stampedes in cattle heard, runs on banks and boom-bust behavior. In all these cases there is exponential growth that is finally limited by finite resources.

The notions of positive and negative feedback are clear if the feedback is static. If the feedback is dynamic its action can change from positive to negative depending on the frequency of the signals. Use of positive feedback will be illustrated by a few examples.

Hewlett's Oscillator

Since positive feedback may generate instability, it can be used to construct oscillators. To limit the exponential growth it is necessary to introduce some nonlinearity that limits the amplitude of the oscillation. An example is given in Figure 2.13, from William Hewlett's Masters thesis from Stanford University in 1938. Hewlett used two vacuum tubes with positive feedback and a nonlinear element in the form of a lamp to maintain constant amplitude of the oscillation. The positive feedback in the basic loop creates an oscillation. The resistance of the lamp decreases as the signal amplitude increases and the amplitude is limited. The oscillator HP200A, based on Hewlett's thesis, was the first product made by Hewlett-Packard, a company that Hewlett started with David Packard in 1939.

The Superregenerative Receiver

In the previous sections we have shown that negative feedback has some very useful properties. The negative feedback amplifier was an enabler for long-distance

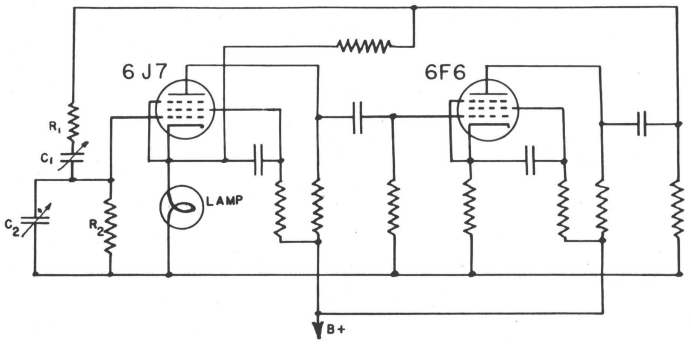


Figure 2.13: Circuit diagram of William Hewlett's oscillator that gives a stable oscillation by using positive feedback and a nonlinearity (lamp) that stabilized the amplitude of the oscillation, [].

telephony. A key idea was to design an amplifier with a very large open loop gain and to reduce the gain by negative feedback. The result was an amplifier that is robust and linear. Positive feedback has complementary properties: it is possible to create high gain but the closed loop system is sensitive to parameter variations.

To understand that positive feedback can generate high gains we consider an amplifier with gain A_{ol}. Referring back to our standard feedback loop in Figure 2.4, we can take the process dynamics to be A_{ol} and the controller dynamics to be k. Neglecting dynamics and closing the loop with *positive* feedback (instead of the negative feedback shown in the diagram), it can be shown that the closed loop system has gain

$$A_{cl} = \frac{A_{ol}}{1 - kA_{ol}}.$$

A very large closed loop gain A_{cl} can be obtained by selecting a feedback gain k that is just below the stability limit $1/A_{ol}$. Choosing $kA_{ol} = 0.99$ gives $A_{cl} = 100A_{ol}$. Armstrong constructed a "superregenerative" radio receiver with only one vacuum tube in 1914, when he was still undergraduate at Columbia University. The drawback by using positive feedback is that the system is highly sensitive and that the gain has to be adjusted carefully to avoid oscillations. It is still used in simple walkie-talkies, garage door openers and toys.

Implementation of Integral Action by Positive Feedback

Positive feedback was used in early controllers where integral action was provided by positive feedback around a system with first order dynamics, as shown in the block diagram of Figure 2.14. Intuitively the system can be explained as follows. Proportional feedback typically gives a steady state error. This can be overcome by adding a bias signal. In Figure 2.14 the bias is estimated by low pass filtering the control signal and adding it back into to the signal path. This serves to compensate for any error that is present.

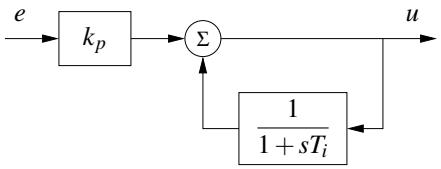


Figure 2.14: Implementation of integral action by *positive feedback* .

The circuit can be understood better by a little analysis. Using block diagram algebra we find that the transfer function of the system is

$$G_{ue} = \frac{k_p}{1 - \frac{1}{1+sT}} = k_p + \frac{k_p}{sT},$$

which is a transfer function of a PI controller. This way of implementing integral action is still used in many industrial regulators.

Positive Feedback Combined with Saturation

Systems with interesting and useful properties can be obtained by combining linear and nonlinear components with positive feedback. Consider the system in Figure 2.15, which consists of a linear block with first-order dynamics and a nonlinear block with positive feedback. Assume that the nonlinearity is

$$y = f(x) = \frac{x}{1+|x|}, \qquad x = f^{-1}(y) = \frac{y}{1-|y|}. \tag{2.34}$$

The system is described by the differential equation

$$\frac{dx}{dt} = -ax + b(r+y) = b(r - G(y)), \qquad G(y) = \frac{af^{-1}(y)}{b} - by. \tag{2.35}$$

The equilibria for a constant input r are given by

$$r = -y + \frac{a}{b} f^{-1}(y) = G(y) \qquad \text{where} \qquad \frac{dG(y)}{dy} = -1 + \frac{a}{b(1-|y|)^2}. \tag{2.36}$$

The graph of the function G is shown in Figure 2.16a for $a = 1$ and $b = 4$. The function has a local maximum $r_{max} = 1 + a/b - 2\sqrt{a/b} = 0.25$ at $y = -1/\sqrt{1+a/b} = -0.5$ and a local minimum $r_{min} = -0.25$ at $y = 0.5$. There is one unique equilibria if $|r| > 0.5$, two equilibria if $|r| = 0.5$ and three equilibria if $|r| < 0.5$. The set of possible equilibria as a function of r is shown in Figure 2.16b.

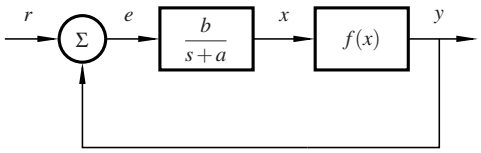
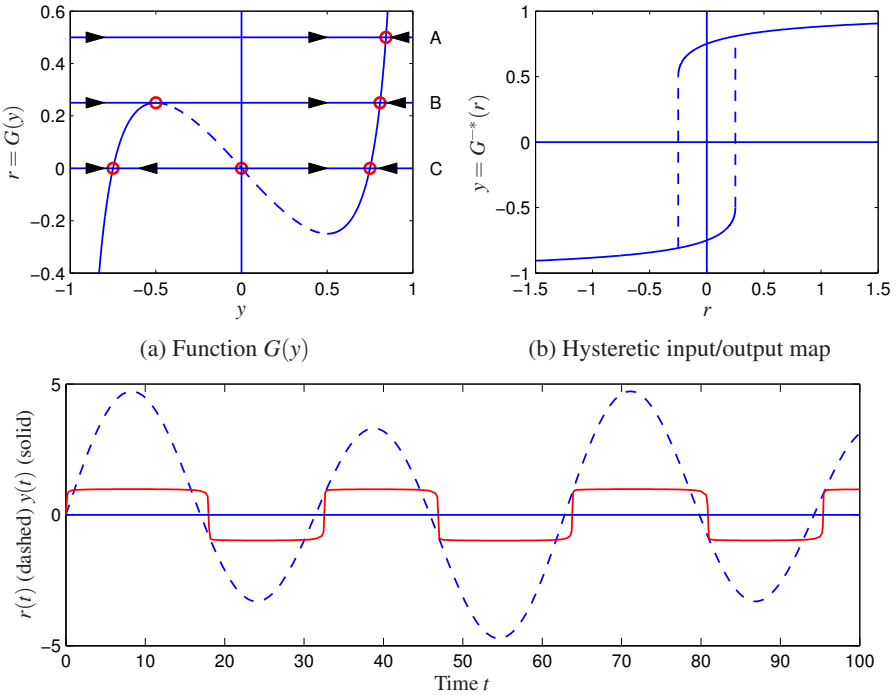


Figure 2.15: Block diagram of system with positive feedback and saturation. The parameters are $a = 1$ and $b = 10$.



(a) Function $G(y)$

(b) Hysteretic input/output map

(c) Input/output behavior

Figure 2.16: System with positive feedback and saturation, (a) show the function $G(y)$ used in the analysis, (b) shows the hysteretic input/output map and (c) shows a simulation of the system. The input r is the dashed curve and the output y is the solid line.

The differential equation (2.36) is of first order and the state $x \in \mathbb{R}$ is a value on the real line, so its behavior can be understood simply by investigating the sign of the derivative dx/dt in the intervals between the equilibria. Since x is a monotone function of y it follows follows from equation (2.35) that dx/dt is positive to the left of an equilibrium point where $G'(y)$ is positive, and negative to the right of this equilibrium point. The stable equilibria correspond to the values of y where the slope of $G(y)$ is positive, which is shown by solid lines in Figure 2.16b. The differential equation thus has two stable equilibria when $r_{min} < r < r_{max}$ and one stable equilibrium when $|r| \geq r_{max}$.

To understand the behavior of the system, we will explore what happens when the reference is changed. If the reference r is zero there are two stable equilibria: we assume that the system is in the stable left equilibrium, where y is negative. If the reference is increased slowly, the equilibrium moves slightly to the right. When the reference reaches the value 0.25, which corresponds an unstable equilibrium, the solution quickly moves to the stable equlibrium, where y is positive, as indicated by the line marked B in Figure 2.16a. If the value of r is increased further, the output y also increases. The input/output relation of the system has

the hysteretic behavior shown in Figure 2.16b, where the dashed line indicate the switches, which occur for $r = \pm r_{max} = \pm(1 + a/b - 2\sqrt{a/b})$.

The temporal behavior of the system is illustrated by the simulations in Figure 2.16c, where the input r is dashed and the output y is solid. The shapes of the signals depend on the parameters; the values $a = 5$, $b = 50$ were used in the figure. The hysteresis width is $2r_{max}$ and the parameter a gives the sharpness of the corners of the output.

The circuit shown in the Figure 2.15 is commonly used as a trigger to detect changes in a signal (known as a Schmitt trigger). It is also used as a memory element in solid state memories. It illustrates that feedback can be used to obtain discrete behavior.

2.6 Feedback and Feedforward

Feedback and feedforward have complementary properties as was discussed in Section 1.5. Feedback only acts when there are deviations between the actual and the desired behaviors, feedforward acts on planned behavior. Feedback and feedforward can be combined to improve the response to command signals and to reduce the effect of disturbances that can be measured. In this section we will discuss feedforward control and how it can be combined with feedback.

Feedforward and System Inversion

To explore feedforward control we will first investigate command signal following. Consider the system modeled by the differential equation (2.1):

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_n y = b_1 \frac{d^{n-1} u}{dt^{n-1}} + \ldots + b_n u.$$

Assume that we want to find a control signal $u(t)$ that gives the response $y_{\mathrm{m}}(t)$, a desired output. It follows from equation (2.1) that the desired control signal satisfies

$$b_1 \frac{d^{n-1} u}{dt^{n-1}} + \ldots + b_n u = \frac{d^n y_{\mathrm{m}}}{dt^n} + a_1 \frac{d^{n-1} y_{\mathrm{m}}}{dt^{n-1}} + \ldots + a_n y_{\mathrm{m}}, \tag{2.37}$$

where we consider y_{m} as the input and u as the output. Equation (2.37) is called the *inverse* of equation (2.1) because it is obtained by exchanging inputs and outputs. If the transfer function of the original system is P, the transfer function of the inverse system is simply $1/P$.

More generally, consider a system with the transfer function P and assume that we want to find a feedforward controller so that the response to command signals is given by the transfer function F_{m}, as shown in Figure 2.17. The feedforward transfer function is then

$$F_{\mathrm{u}} = P^{-1} F_{\mathrm{m}} = \frac{F_{\mathrm{m}}}{P}. \tag{2.38}$$

Design of a feedforward compensator is thus closely related to system inversion.

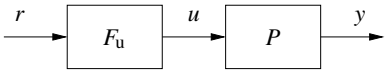


Figure 2.17: Block diagram of a process with pure feedforward compensation. The process transfer function is P, the desired response is characterized by the transfer function F_m the feedforward compensator has the transfer function F_u and the combined system has transfer function $F_m = PF_u$.

Feedforward control has the advantage that it cannot create feedback instabilities, but it also has some disadvantages. If the process transfer function has zeros in the right half plane, the feedforward transfer function given by equation (2.38) will be unstable and the computed response y_m may be unbounded. If the process has time delays, system inversion requires an ideal predictor. The transfer function $F_u(s)$ can also generate very large input signals. Assume for example that

$$P(s) = \frac{0.5}{(s+1)^2}, \quad F_m(s) = \frac{1}{s+1}.$$

It then follows from equation (2.38) that $F_r(s) = 2(s+1)$. To generate the feedforward signal it is thus necessary to differentiate the command signal, which can result in very large control signals if the command signal changes rapidly. In this particular case we must require that the rate of change of the control signal is not too large.

Another drawback of feedforward control is that good process models are required. To understand the effects of errors in the process model we assume that a feedforward F_r is designed for a nominal process model P. It follows from equation (2.38) that the transfer function from reference to process output is $F_m = F_u P$. To investigate what happens when there are small errors dP in the process model, we differentiate F_m with respect to P, which gives

$$\frac{dF_m}{dP} = F_u = \frac{PF_u}{P} = \frac{F_m}{P} \quad \Longrightarrow \quad \frac{dF_m}{F_m} = \frac{dP}{P}. \tag{2.39}$$

The relative error in F_m is thus equal to the relative error in P, which is an analytic justification of the entry in Table 1.1 in Section 1.5 that says that feedforward is sensitive to modeling errors.

Combining Feedforward with Feedback

It seems natural to combine feedforward with feedback. An architecture of such a controller is shown in the block diagram of Figure 2.18. The controller has three blocks: the transfer function F_m that describes the desired response, the feedback transfer function C and the feedforward transfer function F_u.

The controller architecture in Figure 2.18 is highly intuitive. The feedforward signal u_{ff} generates the ideal output $y = y_m$. If there are no disturbances and no modeling errors the feedback error e is then zero and the feedback signal u_{fb} is also zero. All control is thus handled by the feedforward action. If there are disturbances and modeling errors, the error e will not be zero and the feedback controller

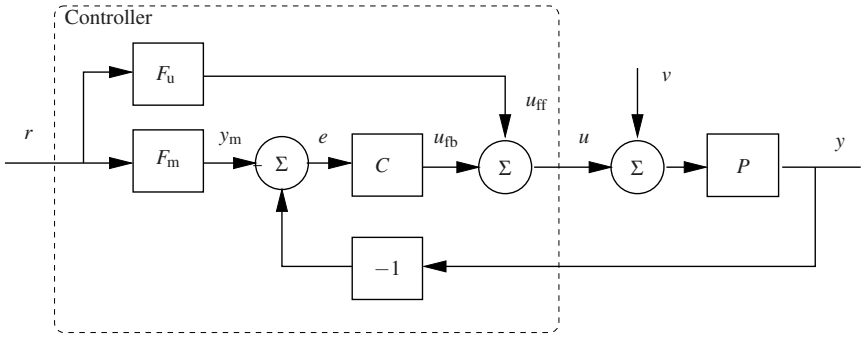


Figure 2.18: Block diagram of a closed loop system where the controller has an architecture with two degrees of freedom. The desired response y_m and the feedforward signal $u_{\rm ff}$ are generated from the reference r. The feedback controller C acts on the control error $e = y_{\rm m} - y$ and generates the feedback control signal $u_{\rm fb}$.

C will make appropriate corrections.

The controller architecture in Figure 2.18 is a generalization of the controller with two degrees of freedom introduced in Section 2.3 (see Figure 2.7). A nice property is that it separates command signal following, robustness and disturbance attenuation. Command signal following is dealt with by design of the transfer function $F_{\rm m}$, which gives the desired response, and the feedforward transfer function $F_{\rm u}$, which generates the nominal input that is required. Robustness and disturbance attenuation is dealt with by design of the feedback transfer function C.

The transfer function from r to y for the system in Figure 2.18 is

$$G_{yr} = \frac{P(F_{\rm u} + CF_{\rm m})}{1 + PC} = F_{\rm m} + \frac{PF_{\rm u} - F_{\rm m}}{1 + PC}. \tag{2.40}$$

The transfer function G_{yr} from reference r to output y equals $F_{\rm m}$ if the feedforward transfer function $F_{\rm u}$ is chosen to be $F_{\rm u} = P^{-1}F_{\rm m}$, which is the same as the condition (2.38) for pure feedforward in an open loop system.

The transfer functions relating the output y and the feedback signal u to the disturbances v are

$$G_{yv}(s) = \frac{P(s)}{1 + P(s)C(s)}, \qquad G_{uv}(s) = -\frac{C(s)}{1 + P(s)C(s)}, \tag{2.41}$$

which do not depend on the transfer functions $F_{\rm m}$ and $F_{\rm u}$. The controller architecture in Figure 2.18 thus admits a decoupling of the response to command signals and the response to disturbances. The feedback controller $C(s)$ provides robustness to process variations and attenuation of load disturbances. The transfer functions $F_{\rm m}$ and $F_{\rm u}$ determine the desired response to command signals.

To find the effects of small modeling errors on $G_{\rm yr}$ we differentiate equation (2.40) with respect to P, which gives

$$\frac{dG_{yr}}{dP} = \frac{G_{yr}}{P} - \frac{G_{yr}C}{1 + PC}$$

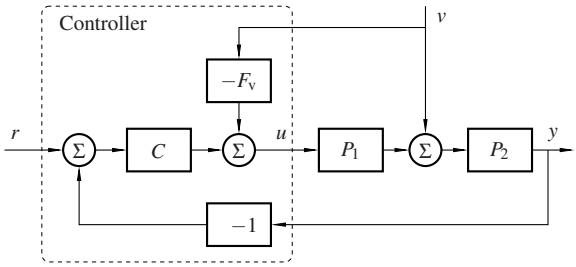


Figure 2.19: Block diagram of a system with a control architecture that combines feedback with feedforward from a disturbance v that can be measured.

Rearranging the equation to compute the relative error gives

$$\frac{dG_{yr}}{G_{yr}} = \frac{dP}{P} - \frac{CdP}{1+PC} = \frac{1}{1+PC}\frac{dP}{P} = S\frac{dP}{P}. \tag{2.42}$$

Comparing this with equation (2.39), we find that the relative errors in the process are reduced by the factor $S = 1/(1+PC)$. Combining feedback and feedforward thus has significant advantages. The function S, called the *sensitivity function*, will be discussed in Section 12.2.

Using Feedforward to Attenuate Measured Disturbances

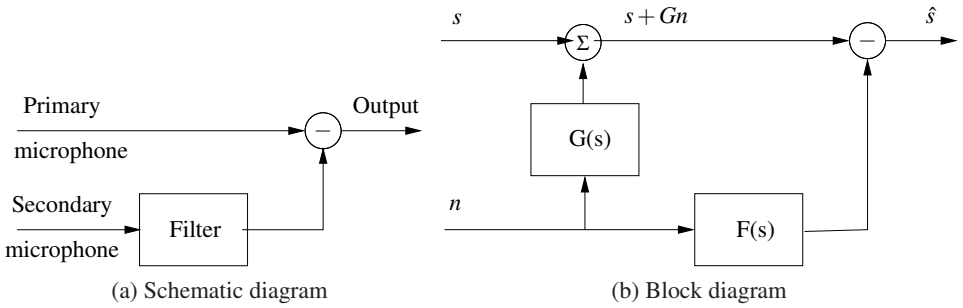
Feedforward can also be used to mitigate the effect of disturbances that can be measured. Such a scheme is shown in Figure 2.19. The process transfer function P is composed of two factors, $P = P_1P_2$. A measured disturbance v enters at the input of process section P_2. The measured disturbance is fed to the process input via the feedforward transfer function F_v.

The transfer function from the disturbance v to process output y is

$$G_{yv} = \frac{P_2(1-P_1F_v)}{1+PC} \tag{2.43}$$

This equation shows that there are two ways of reducing the disturbance. The transfer function $1-P_1F_v$ can be made small by a proper choice of the feedforward transfer function F_v. In feedback compensation the effect of the disturbance is instead reduced by making the the loop transfer function PC large. Feedforward thus makes the error small by subtraction, while feedback instead makes the error small by dividing with $1+PC$. An immediate consequence is that feedforward is more sensitive than feedback since we are trying to match two terms. Feedback gives better robustness but there is a risk of instability. Feedback and feedforward are therefore complementary, and we again can see that it is useful to combine them.

Feedforward is most effective when the disturbance v enters early in the process. This occurs when most of the dynamics are in process section P_2. When $P_2 = P$, and therefore $P_1 = 1$, the feedforward compensator is simply a proportional controller.



(a) Schematic diagram (b) Block diagram

Figure 2.20: Schematic and block diagrams for noise cancellation.

Noise Cancellation

Noise cancellation is a common example of the use of feedforward to cancel effects of disturbances. Consider, for example, a pilot that has to communicate in a noisy cabin. The environmental noise will seriously deteriorate the communication because the pilots microphone will pick up ambient noise. The noise can be reduced significantly by using two microphones as illustrated in Figure 2.20a. The primary microphone is directed towards the pilot. It picks up the pilots voice and ambient noise. The second microphone is directed away from the pilot and it picks up the ambient noise. The effect of the noise can be reduced by filtering the signal from the secondary microphone and subtracting it from the signal from the primary microphone. A block diagram of the system is shown in Figure 2.20b. The transfer function $G(s)$ represents the dynamics of the acoustic transmission from the secondary microphone to the first microphone. The transfer function $F(s)$ is the transfer function of the filter. To cancel the effect of the noise the transfer function $F(s)$ should be close to $G(s)$. Since the noise transmission depends on the situation, for example how the pilot turns his head, it is common to let the filter $F(s)$ be adaptive so that it can adjust, as described later in Example 5.16. Noise cancellation has many applications: in headphones, to create noise-free spaces by active noise control, or to measure electrocardiograms and separate the heartbeats of mother and fetus.

2.7 Using Feedback to Shape Behavior

The regulation and servo problems discussed in Sections 2.2 and 2.3 are classical applications of feedback. In Section 2.4 it was shown that feedback can be used to shape linear input/output behavior from nonlinear components. Feedback can also be used to shape many other behaviors. In robotics feedback is used to obtain collision avoidance. In automotive control it is used to create behaviors that avoid locking the brakes and skidding. Bacteria use simple feedback mechanisms to search for areas where there is high concentration of food or light. The principle is to sense a variable and to make exploratory moves to see if the concentration increases. A similar mechanism can be used to avoid harmful substances. Opti-

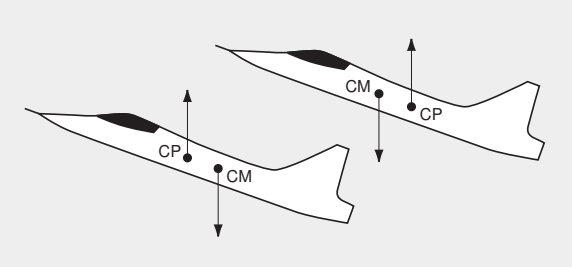


Figure 2.21: Schematic diagram of two aircraft. The aircraft at the top is stable because it has the center of pressure CP behind the center of mass CM. The aircraft at the bottom is unstable because the positions of center of mass and center of pressure are reversed.

mization is also used in computer systems to maintain high throughput of servers. In this section we elaborate on a few cases.

Stabilization

Stabilizing an unstable system is a typical example of how feedback can be used to change behavior. Many systems are naturally unstable. The ability to stand upright, walk and run has given humans many advantages but it requires stabilization. Stability and maneuverability are conflicting goals in vehicle design. The ship designer Minorsky realized that there was a trade-off between maneuverability and stability and he emphasized that a stable ship is difficult to steer. The Wright Flyer, which was maneuverable but unstable, inspired Sperry to design an autopilot. Feedback has been used extensively in aircraft, from simple systems for stability augmentation to systems that provide full autonomy.

Military airplanes gain significant competitive advantage by being made unstable. Schematic pictures of two airplanes are shown in Figure 2.21. The positions of the center of mass CM and the center of pressure CP are key elements. To be stable, the center of pressure must be behind of the center of mass. The center of pressure of an aircraft shifts backwards when a plane goes supersonic. If the plane is stable at subsonic speeds it becomes even more stable at supersonic speeds because of the long distance between CM and CP. Large forces and large control surfaces are then required to maneuver the airplane and the plane will be more sluggish. A more balanced design is obtained by placing the center of pressure in front of the center of mass at subsonic speeds. Such an airplane will have superior performance, but it is unstable at subsonic speeds, typically at takeoff and landing. The control system that stabilizes the aircraft in these operating conditions is mission-critical, with strong requirements on robustness and reliability.

The evolutionary biologist John Maynard Smith [Smi52a] has claimed that while early flying animals were inherently stable, they later developed unstable configurations when their sensory and nervous systems became more sophisticated and able to stabilize. The unstable configuration had significant advantages in maneuverability both for predator and prey.

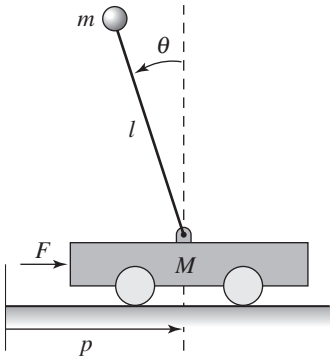


Figure 2.22: Inverted pendulum on a cart.

Keeping an inverted pendulum in the upright position is another example of stabilization. Consider the cart–pendulum system shown in Figure 2.22. Neglecting damping, assuming that the cart is much heavier than the pendulum and assuming that the tilt angle θ is small, the dynamics of the system can be approximated by the differential equation

$$J_t\ddot{\theta} - mgl\theta = u \tag{2.44}$$

(a more detailed derivation is given in Example 3.1). The transfer function of the open loop system is

$$G_{\theta u} = \frac{1}{J_t s^2 - mgl}, \qquad a_{cl}(s) = J_t s^2 - mgl.$$

The system is unstable because it has a pole $s = \sqrt{mgl/J_t} = \omega_0$ that is positive. It can be stabilized with a *proportional-derivative (PD) controller* that has the transfer function

$$C(s) = k_d s + k_p. \tag{2.45}$$

The closed-loop characteristic polynomial is

$$a_{cl}(s) = J_t s^2 + k_d s + (k_p - mgl),$$

and all of its roots are in the left half plane if $k_p > mgl$.

One way to find controller parameters is to choose the controller gains so that the characteristic polynomial has natural frequency ω_c and damping ratio ζ_c, hence

$$k_d = 2\zeta_c\omega_c J_t, \qquad k_p = J_t\omega_c^2 + mgl.$$

Choosing $\omega_c = \omega_0$ gives the closed loop poles $-\zeta_c\omega_0 \pm i\sqrt{1-\zeta_c^2}\omega_0$ which can be compared with the open loop poles at $\pm\omega_0$. The controller gains required are $k_p = 2mgl$ and $k_d = 2\zeta_c\sqrt{mglJ_t}$. The control law (2.45) stabilizes the pendulum but is does not stabilize the motion of the cart. To do this it is necessary to introduce feedback from cart position and cart velocity.

Figure 2.23: Haptic devices, the left figure shows the PHANTOM® and the right a system is developed by Quanser.

Impedance Control and Haptics

Changing behavior of a mechanical system is common in robotics and haptics. Position control is not sufficient when industrial robots are used for grinding, polishing and assembly. The robot can be brought into proximity with the workspace by position control, but to carry out the operations it is desirable to shape how the force depends on the distance between the tool and the workspace. A spring-like behavior is an example. The general problem is to create a behavior specified by a given differential equation between force and motion, a procedure called *impedance control*. Similar situations occur in teleoperation in hazardous environment or in telesurgery. In this situation the workpiece is operated remotely using a joystick. It is useful for the operator to have some indication of the forces between the tool and the workpiece. This can be accomplished by generating a force on the operator's joystick that mimics the force on the workpiece.

Figure 2.23 shows two haptic input devices. The systems are pen-like with levers or gimbals containing angle sensors and force actuation. By sensing position and orientation, and then generating a force depending on position and velocity, it is possible to create a behavior that simulates touching real or virtual objects. Forces that simulate friction and surface structure can also be generated.

We illustrate the principle by considering a joystick having a low friction joint. Let J be the moment of inertia, and let the actuation torque and the external torque from the operator be T_a and T, respectively. The equation of motion is

$$J\frac{d^2\theta}{dt^2} = T + T_a.$$

By measuring the angle θ and its first two derivatives we can create the feedback

$$T_a = k_p(\theta_r - \theta) - k_d\frac{d\theta}{dt} - k_a\frac{d^2\theta}{dt^2}.$$

The closed loop system is then

$$(J + k_a)\frac{d^2\theta}{dt^2} + k_d\frac{d\theta}{dt} + k_p(\theta - \theta_r) = T.$$

The feedback has thus provided virtual inertia $J + k_a$, virtual damping k_d and virtual spring action k_p. If no torque T is applied, the joystick will assume the orientation given by the reference signal θ_r. If a the user applies a torque, the joystick will behave like a damped spring–mass system.

2.8 Further Reading

The books by Bennett [Ben79, Ben93] and Mindel [Min02, Min08] give interesting perspective on the development of control. Much of the material touched upon in this chapter is classical control; see [CM51], [JNP47] and [Tru55]. The notion of controllers with two degrees of freedom was introduced by Horowitz [Hor63]. The analysis will be elaborated in the rest of the book. Transfer functions and other descriptions of dynamics are discussed in Chapters 6 and 9, methods to investigate stability in Chapter 10. The simple method to find parameters of controllers based on matching of coefficients of the closed loop characteristic polynomial is developed further in Chapters 7, 8 and 13. Feedforward control is discussed in Section 8.5 and Section 12.2.

Exercises

2.1 Let $y \in \mathbb{R}$ and $u \in \mathbb{R}$. Solve the differential equations

$$\frac{dy}{dt} + ay = bu, \qquad \frac{d^2y}{dt^2} + 2\frac{dy}{dt} + y = 2\frac{du}{dt} + u,$$

for $t > 0$. Determine the responses to a unit step $u(t) = 1$ and the exponential signal $u(t) = e^{st}$ when the initial condition is zero. Derive the transfer functions of the systems.

2.2 Let $y_0(t)$ be the response of a system with the transfer function $G_0(s)$ to a given input. The transfer function $G(s) = (1 + sT)G_0(s)$ has the same zero frequency gain but it has an additional zero at $z = -1/T$. Let $y(t)$ be the response of the system with the transfer function $G(s)$ and show that

$$y(t) = y_0(t) + T\frac{dy_0}{dt}, \tag{2.46}$$

Next consider the system with the transfer function

$$G(s) = \frac{s + a}{a(s^2 + 2s + 1)},$$

which has unit zero-frequency-gain ($G(0) = 1$). Use the result in equation (2.46) to explore the effect of the zero $s = -1/T$ on the step response of a system

2.3 Consider a closed loop system with process dynamics and a PI controller modeled by

$$\frac{dy}{dt} + ay = bu, \qquad u = k_p(r-y) + k_i \int_0^t (r(\tau) - y(\tau))d\tau,$$

where r is the reference, u the control variable and y the process output. a) Derive a differential equation relating the output y to the difference by direct manipulation of the equations. b) Draw a block diagram of the system. c) Derive the transfer functions of the process and the controller. d) Compute the transfer function from reference r to output y of the closed loop system. Make the derivations both by direct manipulation of the system equations and by polynomial algebra. Compare the results with a direct determination of the transfer functions by inspection of the block diagram.

2.4 The dynamics of the pupillary reflex is approximated by a linear system with the transfer function

$$P(s) = \frac{0.2(1-0.1s)}{(1+0.1s)^3}.$$

Assume that the nerve system that controls the pupil opening is modeled as a proportional controller with the gain k. Use Routh-Hurwitz theorem to determine the largest gain that gives a stable closed loop system.

2.5 A simple model for the relation between speed v and throttle u for a car is given by the transfer function

$$G_{vu} = \frac{b}{s+a}$$

where $b = 1$ m/s^2 and $a = 0.025$ rad/s (see Section 4.1). The control signal is normalized to the range $0 \leq u \leq 1$. Design a PI controller for the system that gives a closed loop system with the characteristic polynomial

$$a_{cl}(s) = s^2 + 2\zeta\omega_c s + \omega_c^2.$$

What are the consequences of choosing different values of the design parameters ζ and ω_c? Use your judgment to find suitable values. Hint: Investigate maximum acceleration and maximum velocity for step changes in the velocity reference.

2.6 Consider the feedback system in Figure 2.4. Let the disturbance $v = 0$, $P(s) = 1$ and $C(s) = k_i/s$. Determine the transfer function G_{yr} from reference r to output y. Also determine how much G_{yr} is changed when the process gain changes by 10%.

2.7 The calculations in Section 2.2 can be interpreted as a design method for a PI controller for a first-order system. A similar calculation can be made for PID control of the second order system. Let the transfer functions of the process and the controller be

$$P(s) = \frac{b}{s^2 + a_1 s + a_2}, \qquad C(s) = k_p + \frac{k_i}{s} + k_d s.$$

Show that the controller parameters

$$k_p = \frac{(1+2\alpha\zeta)\omega_c^2 - a_2}{b}, \qquad k_i = \frac{\alpha\omega_c^3}{b}, \qquad k_d = \frac{(\alpha+2\zeta)\omega_c - a_1}{b}.$$

give a closed loop system with the characteristic polynomial

$$(s^2 + 2\zeta\omega_c s + \omega_c^2)(s+\alpha\omega_c).$$

2.8 Consider an open loop system with the nonlinear input-output relation $y = f(u)$. Assume that the system is closed with the proportional controller $u = k(r - y)$. Show that the input-output relation of the closed loop system is

$$y + \frac{1}{k} f^{-1}(y) = r.$$

Estimate the largest deviation from ideal linear response $y = r$. Illustrate by plotting the input output responses for a) $f(u) = \sqrt{u}$ and b) $f(u) = u^2$ with $0 \le u \le 1$ and $k = 5, 10$ and 100.

2.9 Consider the system in Section 2.2 where the controller was designed to give a closed loop system characterized by $\omega_c = 1$ and $\zeta = 0.707$. The transfer functions of the process and the controller are

$$P(s) = \frac{2}{s+1}, \qquad C(s) = \frac{0.207s + 0.5}{s}.$$

The response of the closed loop system to step command signals has a settling time (time required to stay within 2% of the final value, see Figure 6.9) of $4/\zeta\omega_c \approx 5.66$. Assume that the attenuation of the load disturbances is satisfactory but that we want a closed loop system system that responds five times faster to command signals without overshoot. Determine the transfer functions of a controller with the architecture in Figure 2.18 that gives a response to command signals with a first-order dynamics. Simulate the system in the nominal case of a perfect model and explore the effects of modeling errors by simulation.

2.10 Consider a queuing system modeled by

$$\frac{dx}{dt} = \lambda - \mu_{\max}\frac{x}{x+1}.$$

The model is nonlinear and the dynamics of the system changes significantly with the queuing length; see Example 3.12. Investigate the situation when a PI controller is used for admission control. The arrival intensity λ is then given by

$$\lambda = k_p(r - x) + k_i \int^t (r(t) - x(t))dt.$$

The controller parameters are determined from the approximate model

$$\frac{dx}{dt} = \lambda.$$

Find controller parameters that give the closed loop characteristic polynomial $s^2 + 2s + 1$ for the approximate model. Investigate the behavior of the control strategy for the nonlinear model by simulation for the input $r = 5 + 4\sin(0.1t)$.

Chapter Three
System Modeling

> *... I asked Fermi whether he was not impressed by the agreement between our calculated numbers and his measured numbers. He replied, "How many arbitrary parameters did you use for your calculations?" I thought for a moment about our cut-off procedures and said, "Four." He said, "I remember my friend Johnny von Neumann used to say, with four parameters I can fit an elephant, and with five I can make him wiggle his trunk."*
>
> Freeman Dyson on describing the predictions of his model for meson-proton scattering to Enrico Fermi in 1953 [Dys04].

A model is a precise representation of a system's dynamics used to answer questions via analysis and simulation. The model we choose depends on the questions we wish to answer, and so there may be multiple models for a single dynamical system, with different levels of fidelity depending on the phenomena of interest. In this chapter we provide an introduction to the concept of modeling and present some basic material on two specific methods commonly used in feedback and control systems: differential equations and difference equations.

3.1 Modeling Concepts

A *model* is a mathematical representation of a physical, biological or information system. Models allow us to reason about a system and make predictions about how a system will behave. In this text, we will mainly be interested in models of dynamical systems describing the input/output behavior of systems, and we will often work in "state space" form. As pointed out already in Chapter 1, when using models it is important to keep in mind that they are an *approximation* of the underlying system. Analysis and design using models must always be done carefully to insure that the limits of the model are respected.

Roughly speaking, a dynamical system is one in which the effects of actions do not occur immediately. For example, the velocity of a car does not change immediately when the gas pedal is pushed nor does the temperature in a room rise instantaneously when a heater is switched on. Similarly, a headache does not vanish right after an aspirin is taken, requiring time for it to take effect. In business systems, increased funding for a development project does not increase revenues in the short term, although it may do so in the long term (if it was a good investment). All of these are examples of dynamical systems, in which the behavior of the system evolves with time.

In the remainder of this section we provide an overview of some of the key concepts in modeling. The mathematical details introduced here are explored more

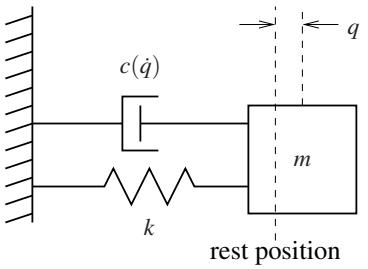


Figure 3.1: Spring–mass system with nonlinear damping. The position of the mass is denoted by q, with $q = 0$ corresponding to the rest position of the spring. The forces on the mass are generated by a linear spring with spring constant k and a damper with force dependent on the velocity $\dot{q}$.

fully in the remainder of the chapter.

The Heritage of Mechanics

The study of dynamics originated in attempts to describe planetary motion. The basis was detailed observations of the planets by Tycho Brahe and the results of Kepler, who found empirically that the orbits of the planets could be well described by ellipses. Newton embarked on an ambitious program to try to explain why the planets move in ellipses, and he found that the motion could be explained by his law of gravitation and the formula stating that force equals mass times acceleration. In the process he also invented calculus and differential equations.

One of the triumphs of Newton's mechanics was the observation that the motion of the planets could be predicted based on the current positions and velocities of all planets. It was not necessary to know the past motion. The *state* of a dynamical system is a collection of variables that completely captures the past motion of a system for the purpose of predicting future motion. For a system of planets the state is simply the positions and the velocities of the planets. We call the set of all possible states the *state space*.

A common class of mathematical models for dynamical systems is ordinary differential equations (ODEs). In mechanics, one of the simplest such differential equations is that of a spring–mass system with damping:

$$m\ddot{q} + c(\dot{q}) + kq = 0. \tag{3.1}$$

This system is illustrated in Figure 3.1. The variable $q \in \mathbb{R}$ represents the position of the mass m with respect to its rest position. We use the notation $\dot{q}$ to denote the derivative of q with respect to time (i.e., the velocity of the mass) and $\ddot{q}$ to represent the second derivative (acceleration). The spring is assumed to satisfy Hooke's law, which says that the force is proportional to the displacement. The friction element (damper) is taken as a nonlinear function $c(\dot{q})$, which can model effects such as stiction and viscous drag. The position q and velocity $\dot{q}$ represent the instantaneous state of the system. We say that this system is a *second-order system* since it has two states which we combine in the *state vector* $x = (q, \dot{q})$.

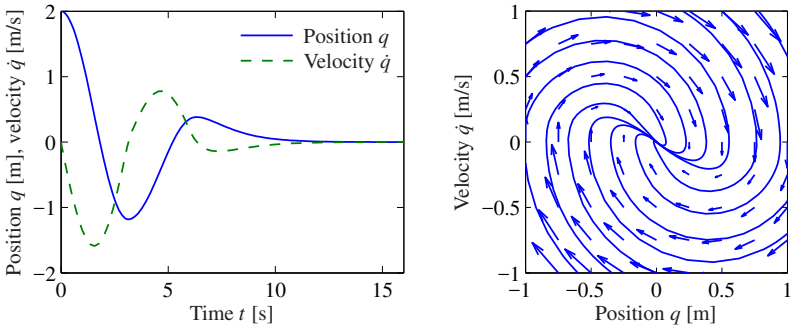


Figure 3.2: Illustration of a state model. A state model gives the rate of change of the state as a function of the state. The plot on the left shows the evolution of the state as a function of time. The plot on the right, called a *phase portrait*, shows the evolution of the states relative to each other, with the velocity of the state denoted by arrows.

The evolution of the position and velocity can be described using either a time plot or a phase portrait, both of which are shown in Figure 3.2. The *time plot*, on the left, shows the values of the individual states as a function of time. The *phase portrait*, on the right, shows the traces of some of the states from different initial conditions: it illustrates how the states move in the state space. In the phase portrait we have also shown arrows that represent the velocity $\dot{x}$ of the state x in a few points. The phase portrait gives a strong intuitive representation of the equation as a vector field or a flow.While systems of second order (two states) can be represented in this way, unfortunately it is difficult to visualize equations of higher order using this approach.

The differential equation (3.1) is called an *autonomous* system because there are no external influences. (Note that this usage of "autonomous" is slightly different than in the phrase "autonomous vehicle".) Such a model is natural for use in celestial mechanics because it is difficult to influence the motion of the planets. In many examples, it is useful to model the effects of external disturbances or controlled forces on the system. One way to capture this is to replace equation (3.1) by

$$m\ddot{q} + c(\dot{q}) + kq = u, \tag{3.2}$$

where u represents the effect of external inputs. The model (3.2) is called a *forced* or *controlled differential equation*. It implies that the rate of change of the state can be influenced by the input $u(t)$. Adding the input makes the model richer and allows new questions to be posed. For example, we can examine what influence external disturbances have on the trajectories of a system. Or, in the case where the input variable is something that can be modulated in a controlled way, we can analyze whether it is possible to "steer" the system from one point in the state space to another through proper choice of the input.

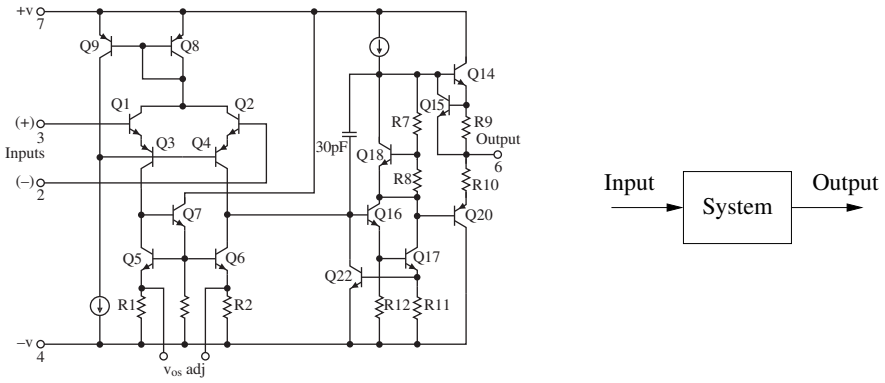


Figure 3.3: Illustration of the input/output view of a dynamical system. The figure on the left shows a detailed circuit diagram for an electronic amplifier; the one on the right is its representation as a block diagram.

The Heritage of Electrical Engineering

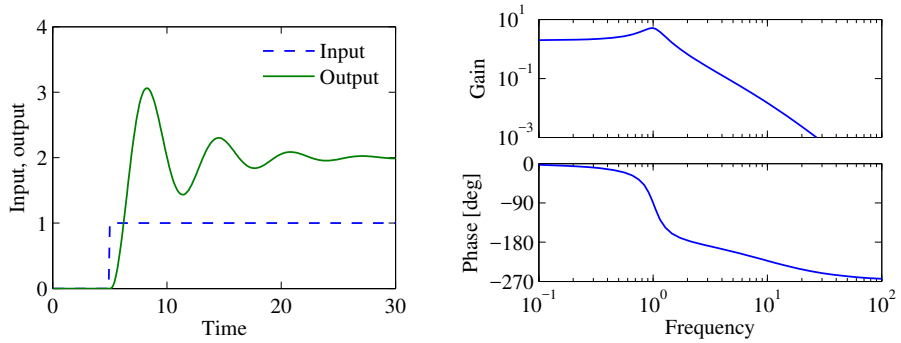
A different view of dynamics emerged from electrical engineering, where the design of electronic amplifiers led to a focus on input/output behavior. A system was considered a device that transforms inputs to outputs, as illustrated in Figure 3.3. Conceptually an input/output model can be viewed as a giant table of inputs and outputs. Given an input signal $u(t)$ over some interval of time, the model should produce the resulting output $y(t)$.

The input/output framework is used in many engineering disciplines since it allows us to decompose a system into individual components connected through their inputs and outputs. Thus, we can take a complicated system such as a radio or a television and break it down into manageable pieces such as the receiver, demodulator, amplifier and speakers. Each of these pieces has a set of inputs and outputs and, through proper design, these components can be interconnected to form the entire system.

The input/output view is particularly useful for the special class of *linear time-invariant systems*. This term will be defined more carefully later in this chapter, but roughly speaking a system is linear if the superposition (addition) of two inputs yields an output that is the sum of the outputs that would correspond to individual inputs being applied separately. A system is time-invariant if the output response for a given input does not depend on when that input is applied.

Many electrical engineering systems can be modeled by linear time-invariant systems, and hence a large number of tools have been developed to analyze them. One such tool is the *step response*, which describes the relationship between an input that changes from zero to a constant value abruptly (a step input) and the corresponding output. As we shall see later in the text, the step response is very useful in characterizing the performance of a dynamical system, and it is often used to specify the desired dynamics. A sample step response is shown in Figure 3.4a.

Another way to describe a linear time-invariant system is to represent it by its



(a) Step response (b) Frequency response

Figure 3.4: Input/output response of a linear system. The step response (a) shows the output of the system due to an input that changes from 0 to 1 at time $t = 5$ s. The frequency response (b) shows the amplitude gain and phase change due to a sinusoidal input at different frequencies.

response to sinusoidal input signals. This is called the *frequency response*, and a rich, powerful theory with many concepts and strong, useful results has emerged. The results are based on the theory of complex variables and Laplace transforms. The basic idea behind frequency response is that we can completely characterize the behavior of a system by its steady-state response to sinusoidal inputs. Roughly speaking, this is done by decomposing any arbitrary signal into a linear combination of sinusoids (e.g., by using the Fourier transform) and then using linearity to compute the output by combining the response to the individual frequencies. A sample frequency response is shown in Figure 3.4b.

The input/output view lends itself naturally to experimental determination of system dynamics, where a system is characterized by recording its response to particular inputs, e.g., a step or a set of sinusoids over a range of frequencies.

The Control View

When control theory emerged as a discipline in the 1940s, the approach to dynamics was strongly influenced by the electrical engineering (input/output) view. A second wave of developments in control, starting in the late 1950s, was inspired by mechanics, where the state space perspective was used. The emergence of space flight is a typical example, where precise control of the orbit of a spacecraft is essential. These two points of view gradually merged into what is today the state space representation of input/output systems. In the 1970s the development was influenced by advances in automation, which emphasized the need to include logic and sequencing.

The development of state space models involved modifying the models from mechanics to include external actuators and sensors and utilizing more general

forms of equations. In control, the model given by equation (3.2) was replaced by

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x,u), \tag{3.3}$$

where x is a vector of state variables, u is a vector of control signals and y is a vector of measurements. The term dx/dt represents the derivative of the vector x with respect to time, and f and h are (possibly nonlinear) mappings of their arguments to vectors of the appropriate dimension. For mechanical systems, the state consists of the position and velocity of the system, so that $x = (q, \dot{q})$ in the case of a damped spring–mass system. Note that in the control formulation we model dynamics as first-order differential equations, but we will see that this can capture the dynamics of higher-order differential equations by appropriate definition of the state and the maps f and h.

Adding inputs and outputs has increased the richness of the classical problems and led to many new concepts. For example, it is natural to ask if possible states x can be reached with the proper choice of u (reachability) and if the measurement y contains enough information to reconstruct the state (observability). These topics will be addressed in greater detail in Chapters 7 and 8.

A final development in building the control point of view was the emergence of disturbances and model uncertainty as critical elements in the theory. The simple way of modeling disturbances as deterministic signals like steps and sinusoids has the drawback that such signals cannot be predicted precisely. A more realistic approach is to model disturbances as random signals. This viewpoint gives a natural connection between prediction and control. The dual views of input/output representations and state space representations are particularly useful when modeling systems with uncertainty since state models are convenient to describe a nominal model but uncertainties are easier to describe using input/output models (often via a frequency response description). Uncertainty will be a constant theme throughout the text and will be studied in particular detail in Chapter 13.

An interesting observation in the design of control systems is that feedback systems can often be analyzed and designed based on comparatively simple models. The reason for this is the inherent robustness of feedback systems. However, other uses of models may require more complexity and more accuracy. One example is feedforward control strategies, where one uses a model to precompute the inputs that cause the system to respond in a certain way. Another area is system validation, where one wishes to verify that the detailed response of the system performs as it was designed. Because of these different uses of models, it is common to use a hierarchy of models having different complexity and fidelity.

Multidomain Modeling

Modeling is an essential element of many disciplines, but traditions and methods from individual disciplines can differ from each other, as illustrated by the previous discussion of mechanical and electrical engineering. A difficulty in systems engineering is that it is frequently necessary to deal with heterogeneous systems

from many different domains, including chemical, electrical, mechanical and information systems.

To model such multidomain systems, we start by partitioning a system into smaller subsystems. Each subsystem is represented by balance equations for mass, energy and momentum, or by appropriate descriptions of information processing in the subsystem. The behavior at the interfaces is captured by describing how the variables of the subsystem behave when the subsystems are interconnected. These interfaces act by constraining variables within the individual subsystems to be equal (such as mass, energy or momentum fluxes). The complete model is then obtained by combining the descriptions of the subsystems and the interfaces.

Using this methodology it is possible to build up libraries of subsystems that correspond to physical, chemical and informational components. The procedure mimics the engineering approach where systems are built from subsystems that are themselves built from smaller components. As experience is gained, the components and their interfaces can be standardized and collected in model libraries. In practice, it takes several iterations to obtain a good library that can be reused for many applications.

State models or ordinary differential equations are not suitable for component-based modeling of this form because states may disappear when components are connected. This implies that the internal description of a component may change when it is connected to other components. As an illustration we consider two capacitors in an electrical circuit. Each capacitor has a state corresponding to the voltage across the capacitors, but one of the states will disappear if the capacitors are connected in parallel. A similar situation happens with two rotating inertias, each of which is individually modeled using the angle of rotation and the angular velocity. Two states will disappear when the inertias are joined by a rigid shaft.

This difficulty can be avoided by replacing differential equations by *differential algebraic equations*, which have the form

$$F(z,\dot{z}) = 0,$$

where $z \in \mathbb{R}^n$. A simple special case is

$$\dot{x} = f(x,y), \qquad g(x,y) = 0, \tag{3.4}$$

where $z = (x,y)$ and $F = (\dot{x} - f(x,y), g(x,y))$. The key property is that the derivative $\dot{z}$ is not given explicitly and there may be pure algebraic relations between the components of the vector z. Modeling using differential algebraic equations is also called *equation-based modeling*, *acausal modeling* or *behavioral modeling*.

The model (3.4) captures the examples of the parallel capacitors and the linked rotating inertias. For example, when two capacitors are connected, we simply add the algebraic equation expressing that the voltages across the capacitors are the same.

Modelica is a language that has been developed to support component-based modeling. Differential algebraic equations are used as the basic description, and object-oriented programming is used to structure the models. Modelica is used to

model the dynamics of technical systems in domains such as mechanical, electrical, thermal, hydraulic, thermofluid and control subsystems. Modelica is intended to serve as a standard format so that models arising in different domains can be exchanged between tools and users. A large set of free and commercial Modelica component libraries are available and are used by a growing number of people in industry, research and academia. For further information about Modelica, see http://www.modelica.org or Tiller [Til01].

Finite State Machines and Hybrid Systems

A final type of modeling has been developed within the computer-controlled systems community. A hybrid system (also called a *cyberphysical system*) is one that combines continuous dynamics with discrete logic. The discrete portion of the system represents logical variables that reside in a computer, such as the mode of a system (on, off, degraded, etc.).

Discrete state dynamics are often represented using a *finite state machine* that consists of a finite set of discrete states $\alpha \in \mathbb{Q}$. We can think of α as the "mode" of the system. The dynamics of a finite state machine are defined in terms of transitions between the states. One convenient representation is as a *guarded transition system*:

$$g_i(\alpha, \beta) \implies \alpha' = r_i(\alpha), \qquad i = 1, \ldots, N.$$

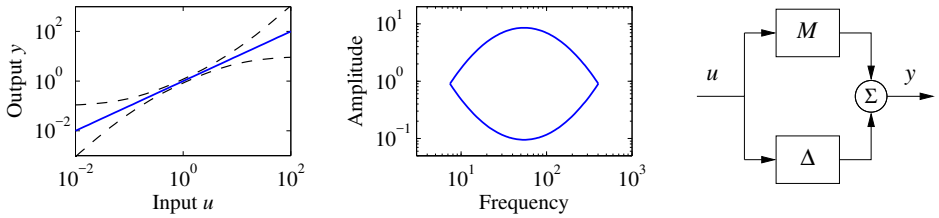
Here the function g is a Boolean (true/false) function that depends on the current system mode α and an input β, which might represent an environmental event (button press, component failure, etc). If the guard g_i is true then the system transitions from the current state α to a new state α', determined by the rule (transition map) r_i. A guarded transition system can have many different rules, depending on the system state and external input.

It is also possible to combine systems that have finite states with those having discrete states, creating a *hybrid system*. For example, if a system has a continuous state x and discrete state α, we might write the overall system dynamics as

$$\frac{dx}{dt} = f_\alpha(x, u, v), \quad g_i(x, \alpha, \beta) \implies \alpha' = r_i(x, \alpha), \qquad i = 1, \ldots, N.$$

In this representation, the continuous dynamics (with state x) are governed by an ordinary differential equation that may depend on the system mode α (indicated by the subscript in f_α). The discrete transition system is also influenced by the continuous state, so that the guards g_i and rules r_i now depend on the continuous state.

Many other representations are possible for hybrid systems, including models that allow a non-continuous change in the continuous variables when a change in the discrete state occurs (so-called *reset logic*). Computer modeling packages for hybrid systems include StateFlow (part of the MATLAB suite of tools), Modelica and Ptolemy [Pto14].



(a) Static uncertainty (b) Uncertainty lemon (c) Model uncertainty

Figure 3.5: Characterization of model uncertainty. Uncertainty of a static system is illustrated in (a), where the solid line indicates the nominal input/output relationship and the dashed lines indicate the range of possible uncertainty. The uncertainty lemon [GPD59] in (b) is one way to capture uncertainty in dynamical systems emphasizing that a model is valid only in the amplitude and frequency ranges within the shaded region. In (c) a model is represented by a nominal model M and another model Δ representing the uncertainty analogous to the representation of parameter uncertainty.

Model Uncertainty

Reducing uncertainty is one of the main reasons for using feedback, and it is therefore important to characterize uncertainty. When making measurements, there is a good tradition to assign both a nominal value and a measure of uncertainty. It is useful to apply the same principle to modeling, but unfortunately it is often difficult to express the uncertainty of a model quantitatively.

For a static system whose input/output relation can be characterized by a function, uncertainty can be expressed by an uncertainty band as illustrated in Figure 3.5a. At low signal levels there are uncertainties due to sensor resolution, friction and quantization. For example, some models for queuing systems or cells are based on averages that exhibit significant variations for small populations. At large signal levels there are saturations or even system failures. The signal ranges where a model is reasonably accurate vary dramatically between applications, but it is rare to find models that are accurate for signal ranges larger than 10^4.

Characterization of the uncertainty of a dynamic model is much more difficult. We can try to capture uncertainties by assigning uncertainties to parameters of the model, but this is often not sufficient. There may be errors due to phenomena that have been neglected, e.g., small time delays. In control the ultimate test is how well a control system based on the model performs, and time delays can be important. There is also a frequency aspect. There are slow phenomena, such as aging, that can cause changes or drift in the systems. There are also high-frequency effects: a resistor will no longer be a pure resistance at very high frequencies, and a beam has stiffness and will exhibit additional dynamics when subject to high-frequency excitation. The *uncertainty lemon* [GPD59] shown in Figure 3.5b is one way to conceptualize the uncertainty of a system. It illustrates that a model is valid only in certain amplitude and frequency ranges.

We will introduce some formal tools for representing uncertainty in Chapter 13

using figures such as Figure 3.5c. These tools make use of the concept of a transfer function, which describes the frequency response of an input/output system. For now, we simply note that one should always be careful to recognize the limits of a model and not to make use of models outside their range of applicability. For example, one can describe the uncertainty lemon and then check to make sure that signals remain in this region. In early analog computing, a system was simulated using operational amplifiers, and it was customary to give alarms when certain signal levels were exceeded. Similar features can be included in digital simulation.

Algebraic Loops

When analyzing or simulating a system described by a block diagram, we need to form the differential equations that describe the complete system. In many cases the equations can be obtained by combining the differential equations that describe each subsystem and substituting variables. This simple procedure cannot be used when there are closed loops of subsystems that all have a direct connection between inputs and outputs, known as an *algebraic loop*. A *direct connection* means that a change in the output u gives an instantaneous change in the output y.

To see what can happen, consider a system with two blocks, a first-order nonlinear system,

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x), \tag{3.5}$$

and a proportional controller described by $u = -ky$. There is no direct connection since the function h does not depend on u. In that case we can obtain the equation for the closed loop system simply by replacing u by $-ky = -kh(x)$ in equation (3.5) to give

$$\frac{dx}{dt} = f(x,-kh(x)), \qquad y = h(x),$$

which is an ordinary differential equation.

The situation is more complicated if there is a direct connection. If $y = h(x,u)$, then replacing u by $-ky$ gives

$$\frac{dx}{dt} = f(x,-ky), \qquad y = h(x,-ky).$$

To obtain a differential equation for x, the algebraic equation $y = h(x,-ky)$ must be first be solved to give $y = \alpha(x)$, which in general is a complicated task.

When algebraic loops are present, it is necessary to solve algebraic equations to obtain the differential equations for the complete system. The resulting model becomes a set of differential algebraic equation, similar to equation 3.4. Resolving algebraic loops is a nontrivial problem because it requires the symbolic solution of algebraic equations. Most block diagram-oriented modeling languages cannot handle algebraic loops, and they simply give a diagnosis that such loops are present. In the era of analog computing, algebraic loops were eliminated by introducing

fast dynamics between the loops. This created differential equations with fast and slow modes that are difficult to solve numerically. Advanced modeling languages like Modelica use several sophisticated methods to resolve algebraic loops.

3.2 State Space Models

In this section we introduce the two primary forms of models that we use in this text: differential equations and difference equations. Both make use of the notions of state, inputs, outputs and dynamics to describe the behavior of a system. We also briefly discuss modeling of finite state systems.

Ordinary Differential Equations

The state of a system is a collection of variables that summarize the past of a system for the purpose of predicting the future. For a physical system the state is composed of the variables required to account for storage of mass, momentum and energy. A key issue in modeling is to decide how accurately this storage has to be represented. The state variables are gathered in a vector $x \in \mathbb{R}^n$ called the *state vector*. The control variables are represented by another vector $u \in \mathbb{R}^p$, and the measured signal by the vector $y \in \mathbb{R}^q$. A system can then be represented by the differential equation

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x,u), \tag{3.6}$$

where $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^q$ are smooth mappings. We call a model of this form a *state space model*.

The dimension of the state vector is called the *order* of the model. The model given in equation (3.6) is called *time-invariant* because the functions f and h do not depend explicitly on time t; there are more general time-varying systems where the functions do depend on time. The model consists of two functions: the function f gives the rate of change of the state vector as a function of state x and control u, and the function h gives the measured values as functions of state x and control u.

A model is called a *linear* state space model (or often just a "linear system") if the functions f and h are linear in x and u. A linear state space model can thus be represented by

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du, \tag{3.7}$$

where A, B, C and D are constant matrices. Such a model is said to be *linear and time-invariant*, or LTI for short. (In this text we will usually omit the term time-invariant and just say the model is linear.) The matrix A is called the *dynamics matrix*, the matrix B is called the *control matrix*, the matrix C is called the *sensor matrix* and the matrix D is called the *direct term*. Frequently models will not have a direct term, indicating that the control signal does not influence the output directly.

A different form of linear differential equations, generalizing the second-order dynamics from mechanics, is an equation of the form

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n y = u, \tag{3.8}$$

where t is the independent (time) variable, $y(t)$ is the dependent (output) variable and $u(t)$ is the input. The notation $d^k y/dt^k$ is used to denote the kth derivative of y with respect to t, sometimes also written as $y^{(k)}$. The controlled differential equation (3.8) is said to be an nth-order model. This model can be converted into state space form by defining

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} d^{n-1}y/dt^{n-1} \\ d^{n-2}y/dt^{n-2} \\ \vdots \\ dy/dt \\ y \end{pmatrix},$$

and the state space equations become

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} -a_1 x_1 - \cdots - a_n x_n \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{pmatrix} + \begin{pmatrix} u \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \qquad y = x_n.$$

With the appropriate definitions of A, B, C and D, this equation is in linear state space form.

An even more general model is obtained by letting the output be a linear combination of the states of the model, i.e.,

$$y = b_1 x_1 + b_2 x_2 + \cdots + b_n x_n + du.$$

This model can be represented in state space as

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u, \tag{3.9}$$

$$y = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix} x + du.$$

This particular form of a linear state space model is called *reachable canonical form* and will be studied in more detail in later chapters. Many other representations for a model are possible and we shall see several of these in Chapters 6–8. It is also possible to expand the form of equation (3.8) to allow derivatives of the input to appear, as we saw briefly in Chapter 2.

(a) Segway (b) Saturn rocket (c) Cart–pendulum system

Figure 3.6: Balance systems. (a) Segway Personal Transporter, (b) Saturn rocket and (c) inverted pendulum on a cart. Each of these examples uses forces at the bottom of the system to keep it upright.

Example 3.1 Balance systems

An example of a type of system that can be modeled using ordinary differential equations is the class of *balance systems*. A balance system is a mechanical system in which the center of mass is balanced above a pivot point. Some common examples of balance systems are shown in Figure 3.6. The Segway® Personal Transporter (Figure 3.6a) uses a motorized platform to stabilize a person standing on top of it. When the rider leans forward, the transportation device propels itself along the ground but maintains its upright position. Another example is a rocket (Figure 3.6b), in which a gimbaled nozzle at the bottom of the rocket is used to stabilize the body of the rocket above it. Other examples of balance systems include humans or other animals standing upright or a person balancing a stick on their hand.

Balance systems are a generalization of the spring–mass system we saw earlier. We can write the dynamics for a mechanical system in the general form

$$M(q)\ddot{q} + C(q,\dot{q}) + K(q) = B(q)u,$$

where $M(q)$ is the inertia matrix for the system, $C(q,\dot{q})$ represents the Coriolis forces as well as the damping, $K(q)$ gives the forces due to potential energy and $B(q)$ describes how the external applied forces couple into the dynamics. Note that q may be a vector, rather than just a scalar, and represents the configuration variables of the system. The specific form of the equations can be derived using Newtonian mechanics. Each of the terms depends on the configuration of the system q and these terms are often nonlinear in the configuration variables.

Figure 3.6c shows a simplified diagram for a balance system consisting of an inverted pendulum on a cart. To model this system, we choose state variables that represent the position and velocity of the base of the system, p and $\dot{p}$, and the angle and angular rate of the structure above the base, θ and $\dot{\theta}$. We let F represent the force applied at the base of the system, assumed to be in the horizontal direc-

tion (aligned with p), and choose the position and angle of the system as outputs. With this set of definitions, the dynamics of the system can be computed using Newtonian mechanics and have the form

$$\begin{pmatrix} (M+m) & -ml\cos\theta \\ -ml\cos\theta & (J+ml^2) \end{pmatrix} \begin{pmatrix} \ddot{p} \\ \ddot{\theta} \end{pmatrix} + \begin{pmatrix} c\dot{p} + ml\sin\theta\,\dot{\theta}^2 \\ \gamma\dot{\theta} - mgl\sin\theta \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}, \qquad (3.10)$$

where M is the mass of the base, m and J are the mass and moment of inertia of the system to be balanced, l is the distance from the base to the center of mass of the balanced body, c and γ are coefficients of viscous friction and g is the acceleration due to gravity.

We can rewrite the dynamics of the system in state space form by defining the state as $x = (p, \theta, \dot{p}, \dot{\theta})$, the input as $u = F$ and the output as $y = (p, \theta)$. If we define the total mass and total inertia as

$$M_t = M + m, \qquad J_t = J + ml^2,$$

the equations of motion then become

$$\frac{d}{dt}\begin{pmatrix} p \\ \theta \\ \dot{p} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{p} \\ \dot{\theta} \\ \dfrac{-mls_\theta\dot{\theta}^2 + mg(ml^2/J_t)s_\theta c_\theta - c\dot{p} - (\gamma/J_t)mlc_\theta\dot{\theta} + u}{M_t - m(ml^2/J_t)c_\theta^2} \\ \dfrac{-ml^2 s_\theta c_\theta\dot{\theta}^2 + M_t gls_\theta - clc_\theta\dot{p} - \gamma(M_t/m)\dot{\theta} + lc_\theta u}{J_t(M_t/m) - m(lc_\theta)^2} \end{pmatrix},$$

$$y = \begin{pmatrix} p \\ \theta \end{pmatrix},$$

where we have used the shorthand $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$.

In many cases, the angle θ will be very close to 0, and hence we can use the approximations $\sin\theta \approx \theta$ and $\cos\theta \approx 1$. Furthermore, if $\dot{\theta}$ is small, we can ignore quadratic and higher terms in $\dot{\theta}$. Substituting these approximations into our equations, we see that we are left with a *linear* state space equation

$$\frac{d}{dt}\begin{pmatrix} p \\ \theta \\ \dot{p} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & m^2l^2g/\mu & -cJ_t/\mu & -\gamma lm/\mu \\ 0 & M_t mgl/\mu & -clm/\mu & -\gamma M_t/\mu \end{pmatrix} \begin{pmatrix} p \\ \theta \\ \dot{p} \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ J_t/\mu \\ lm/\mu \end{pmatrix} u,$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} x,$$

where $\mu = M_t J_t - m^2 l^2$. ∇

Example 3.2 Inverted pendulum

A variation of the previous example is one in which the location of the base p does not need to be controlled. This happens, for example, if we are interested only in stabilizing a rocket's upright orientation without worrying about the location of the

base of the rocket. The dynamics of this simplified system are given by

$$\frac{d}{dt}\begin{pmatrix}\theta\\ \dot\theta\end{pmatrix} = \begin{pmatrix}\dot\theta\\ \dfrac{mgl}{J_t}\sin\theta - \dfrac{\gamma}{J_t}\dot\theta + \dfrac{l}{J_t}u\cos\theta\end{pmatrix}, \qquad y=\theta, \tag{3.11}$$

where γ is the coefficient of rotational friction, $J_t = J + ml^2$ and u is the force applied at the base. This system is referred to as an *inverted pendulum*. ∇

Difference Equations

In some circumstances, it is more natural to describe the evolution of a system at discrete instants of time rather than continuously in time. If we refer to each of these times by an integer $k = 0,1,2,\ldots$, then we can ask how the state of the system changes for each k. Just as in the case of differential equations, we define the state to be the set of variables that summarizes the past of the system for the purpose of predicting its future. Systems described in this manner are referred to as *discrete-time systems*.

The evolution of a discrete-time system can be written in the form

$$x[k+1] = f(x[k],u[k]), \qquad y[k] = h(x[k],u[k]), \tag{3.12}$$

where $x[k] \in \mathbb{R}^n$ is the state of the system at time k (an integer), $u[k] \in \mathbb{R}^p$ is the input and $y[k] \in \mathbb{R}^q$ is the output. As before, f and h are smooth mappings of the appropriate dimension. We call equation (3.12) a *difference equation* since it tells us how $x[k+1]$ differs from $x[k]$. The state $x[k]$ can be either a scalar- or a vector-valued quantity; in the case of the latter we write $x_j[k]$ for the value of the jth state at time k.

Just as in the case of differential equations, it is often the case that the equations are linear in the state and input, in which case we can describe the system by

$$x[k+1] = Ax[k] + Bu[k], \qquad y[k] = Cx[k] + Du[k].$$

As before, we refer to the matrices A, B, C and D as the dynamics matrix, the control matrix, the sensor matrix and the direct term. The solution of a linear difference equation with initial condition $x[0]$ and input $u[0],\ldots,u[T]$ is given by

$$\begin{aligned} x[k] &= A^k x[0] + \sum_{j=0}^{k-1} A^{k-j-1}Bu[j], \\ y[k] &= CA^k x[0] + \sum_{j=0}^{k-1} CA^{k-j-1}Bu[j] + Du[k], \end{aligned} \qquad k > 0. \tag{3.13}$$

Difference equations are also useful as an approximation of differential equations, as we will show later.

Example 3.3 Predator–prey

As an example of a discrete-time system, consider a simple model for a predator–prey system. The predator–prey problem refers to an ecological system in which

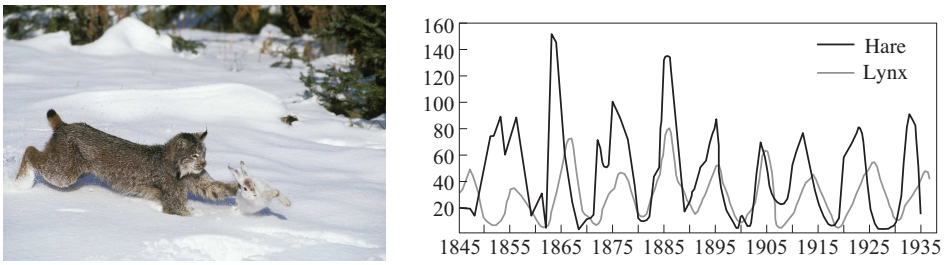


Figure 3.7: Predator versus prey. The photograph on the left shows a Canadian lynx and a snowshoe hare, the lynx's primary prey. The graph on the right shows the populations of hares and lynxes between 1845 and 1935 in a section of the Canadian Rockies [Mac37]. The data were collected on an annual basis over a period of 90 years. (Photograph copyright Tom and Pat Leeson.)

we have two species, one of which feeds on the other. This type of system has been studied for decades and is known to exhibit interesting dynamics. Figure 3.7 shows a historical record taken over 90 years for a population of lynxes versus a population of hares [Mac37]. As can been seen from the graph, the annual records of the populations of each species are oscillatory in nature.

A simple model for this situation can be constructed using a discrete-time model to keep track of the rate of births and deaths of each species. Letting H represent the population of hares and L represent the population of lynxes, we can describe the state in terms of the populations at discrete periods of time. Letting k be the discrete-time index (e.g., the day or month number), we can write

$$\begin{aligned} H[k+1] &= H[k] + b_r(u)H[k] - aL[k]H[k], \\ L[k+1] &= L[k] + cL[k]H[k] - d_f L[k], \end{aligned} \tag{3.14}$$

where $b_r(u)$ is the hare birth rate per unit period and is a function of the food supply u, d_f is the lynx mortality rate and a and c are the interaction coefficients. The interaction term $aL[k]H[k]$ models the rate of predation, which is assumed to be proportional to the rate at which predators and prey meet and is hence given by the product of the population sizes. The interaction term $cL[k]H[k]$ in the lynx dynamics has a similar form and represents the rate of growth of the lynx population. This model makes many simplifying assumptions—such as the fact that hares decrease in number only through predation by lynxes—but it often is sufficient to answer basic questions about the system.

To illustrate the use of this system, we can compute the number of lynxes and hares at each time point from some initial population. This is done by starting with $x[0] = (H_0, L_0)$ and then using equation (3.14) to compute the populations in the following period. By iterating this procedure, we can generate the population over time. The output of this process for a specific choice of parameters and initial conditions is shown in Figure 3.8. While the details of the simulation are different from the experimental data (to be expected given the simplicity of our assumptions), we see qualitatively similar trends and hence we can use the model to help

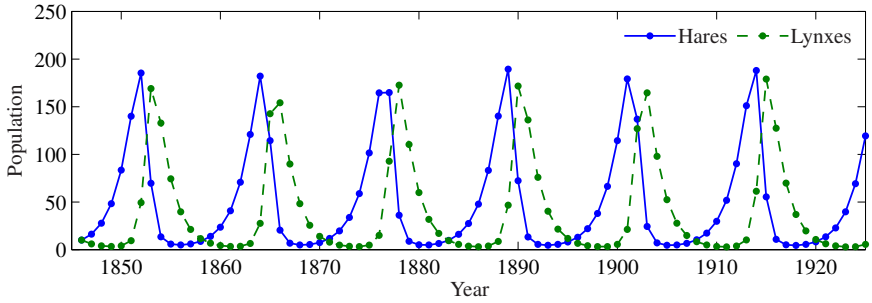


Figure 3.8: Discrete-time simulation of the predator–prey model (3.14). Using the parameters $a = c = 0.014$, $b_r(u) = 0.6$ and $d = 0.7$ in equation (3.14) with daily updates, the period and magnitude of the lynx and hare population cycles approximately match the data in Figure 3.7.

explore the dynamics of the system. ∇

Example 3.4 E-mail server

The IBM Lotus server is a collaborative software system that administers users' e-mail, documents and notes. Client machines interact with end users to provide access to data and applications. The server also handles other administrative tasks. In the early development of the system it was observed that the performance was poor when the central processing unit (CPU) was overloaded because of too many service requests, and mechanisms to control the load were therefore introduced.

The interaction between the client and the server is in the form of remote procedure calls (RPCs). The server maintains a log of statistics of completed requests. The total number of requests being served, called `RIS` (RPCs in server), is also measured. The load on the server is controlled by a parameter called `MaxUsers`, which sets the total number of client connections to the server. This parameter is controlled by the system administrator. The server can be regarded as a dynamical system with `MaxUsers` as the input and `RIS` as the output. The relationship between input and output was first investigated by exploring the steady-state performance and was found to be linear.

In [HDPT04] a dynamic model in the form of a first-order difference equation is used to capture the dynamic behavior of this system. Using system identification techniques, they construct a model of the form

$$y[k+1] = ay[k] + bu[k],$$

where $u = \texttt{MaxUsers} - \overline{\texttt{MaxUsers}}$ and $y = \texttt{RIS} - \overline{\texttt{RIS}}$. The parameters $a = 0.43$ and $b = 0.47$ are parameters that describe the dynamics of the system around the operating point, and $\overline{\texttt{MaxUsers}} = 165$ and $\overline{\texttt{RIS}} = 135$ represent the nominal operating point of the system. The number of requests was averaged over a sampling period of 60 s. ∇

Another application of difference equation is in the implementation of control systems on computers. Early controllers were analog physical systems, which can

be modeled by differential equations. When implementing a controller described by a differential equation using a computer it is necessary to do approximations. A simple way is to approximate derivatives by finite differences, as illustrated by the following example.

Example 3.5 Difference approximation of a PI controller
Consider the PI controller

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)d\tau = k_p e(t) + x(t), \qquad x = k_i \int_0^t e(\tau)d\tau,$$

where the controller state is given by the differential equation

$$\frac{dx}{dt} = k_i e(t) \tag{3.15}$$

Assume that the error is measured at regular sampling intervals $t = h, 2h, 3h, \ldots$. Approximating the derivative in equation (3.15) by differences gives

$$\frac{x(kh+h) - x(kh)}{h} = k_i e(kh),$$

and the controller is then given by the difference equation

$$x[k+1] = x[k] + hk_i e[k], \qquad u[k] = k_p e[k] + x[k],$$

where $x[k] = x(kh)$, $e[k] = e(kh)$ and $u[k] = u(kh)$ represent the discrete-time state, error and input sampled at each time interval. This controller is easy to implement on a computer since it consists of just addition and multiplication. ∇

The approximation in the example works well provided that the sampling interval is so short that the variable $e(t)$ changes very little over a sampling interval.

Finite State Machines

A finite state machine is a model in which the states of the system are chosen from a finite list of "modes". The dynamics of a finite state machine are given by transitions between these modes, possibly in response to external signals. We illustrate this concept with a simple example.

Example 3.6 Traffic light controller
Consider a finite state machine model of a traffic light control system, as shown in Figure 3.6. We represent the state of the system in terms of the set of traffic lights that are turned on (either east–west or north–south). In addition, once a light is turned on it should stay that way for a certain minimum time, and then only change when a car comes up to the intersection in the opposite direction. This gives us two states for each direction of the lights: waiting for a car to arrive and waiting for the timer to expire. Thus, we have four states for the system, as shown in Figure 3.6.

The dynamics for the light describe how the system transitions from one state to another. Starting at the left most state, we assume that the lights are set to allow

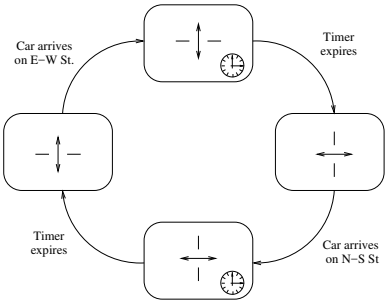


Figure 3.9: A simple model for a traffic light. The diagram on the right is a finite state machine model of the traffic light controller.

traffic in the north–south direction. When a car arrives on the east–west street, we transition to the state at the top of the diagram, where a timer is started. Once the timer reaches the designated amount of time, we transition to the state on the right side of the diagram and turn on the lights in the east–west direction. From here we wait until a car arrives on the north–south street and continue the cycle.

Viewed as a control system, this model has a state space consisting of four discrete states: north–south waiting, north–south countdown, east–west waiting, and east–west countdown. The inputs to the controller consist of the signals that indicate whether a car is present at the roads leading up to the intersection. The outputs from the controller are the signals that change the colors of the traffic light. Finally, the dynamics of the controller are the transition diagram that controls how the states (or modes) of the system change in time. ∇

More formally, a finite state machine can be represented as a finite set of discrete states $\alpha \in \mathbb{Q}_{\text{sys}}$, where $\mathbb{Q}_{\text{sys}}$ is a discrete set. The dynamics of the system are described by transitions between the discrete states, as in the finite state machine described in the previous example. These transitions can depend on external inputs or measurements and can generate output actions on transition into or out of a given state. If we let $\beta \in \mathbb{Q}_{\text{in}}$ represent (discrete) input events (button press, component failure, etc) and $\gamma \in \mathbb{Q}_{\text{out}}$ represent (discrete) output actions (such as turning off of device), then the dynamics of the finite state machine can be written as a guarded command system

$$g_i(\alpha,\beta) \implies \begin{array}{l} \alpha' = r_i(\alpha,\beta), \\ \gamma = a_i(\alpha,\beta), \end{array} \qquad i = 1,\ldots,N. \tag{3.16}$$

Here the function g is a Boolean (true/false) function that depends on the current system mode α and an external input β. If the guard g_i is true then the system transitions from the current state α to a new state α', determined by the rule (transition map) r_i and the external input. The output action γ is similarly dependent on the current state and external input. A guarded transition system can have many different rules, depending on the system state and external input.

The dynamics of a transition system is similar in many ways to the discrete

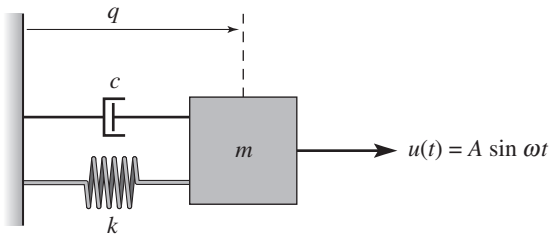


Figure 3.10: A driven spring–mass system with damping. Here we use a linear damping element with coefficient of viscous friction c. The mass is driven with a sinusoidal force of amplitude A.

time dynamics in equation (3.12). The major difference is that the transitions do not necessarily occur at regularly spaced intervals of time. Indeed, there is no strict notion of time in a transition system as we have described it here: it is only the sequence of events that is kept track of (through the evolution of the discrete state).

Specifications for finite transition systems are often written as logical functions describing the conditions that should be imposed on the system. For example, we might wish to say that if a specific sensor is not operating, then the system cannot transition to a mode that requires the use of that sensor. This could be written as the logical formula

$$\alpha \in \{\text{states with sensor } k \text{ not functioning}\} \implies \alpha' \notin \{\text{states requiring sensor } k\}.$$

The formula of the form $p \implies q$ where p and q are Boolean propositions can be written as the logical function $(!p)\,||\,(p\,\&\&\,q)$, which asserts that if proposition p is true then proposition q must be true. In the sensor example, p and q are represented by whether the system mode α is in some set of states.

Finite state machines are very useful for describing logical operations and are often combined with continuous state models (differential or different equations) to create a hybrid system model. The study of hybrid systems is beyond the scope of this text, but excellent references include Lee and Seshia [LS15] and Alur [Alu15].

Simulation and Analysis

State space models can be used to answer many questions. One of the most common, as we have seen in the previous examples, involves predicting the evolution of the system state from a given initial condition. While for simple models this can be done in closed form, more often it is accomplished through computer simulation.

Consider again the damped spring–mass system from Section 3.1, but this time with an external force applied, as shown in Figure 3.10. We wish to predict the motion of the system for a periodic forcing function, with a given initial condition, and determine the amplitude, frequency and decay rate of the resulting motion.

We choose to model the system with a linear ordinary differential equation.

Using Hooke's law to model the spring and assuming that the damper exerts a force that is proportional to the velocity of the system, we have

$$m\ddot{q} + c\dot{q} + kq = u, \tag{3.17}$$

where m is the mass, q is the displacement of the mass, c is the coefficient of viscous friction, k is the spring constant and u is the applied force. In state space form, using $x = (q, \dot{q})$ as the state and choosing $y = q$ as the output, we have

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ -\dfrac{c}{m}x_2 - \dfrac{k}{m}x_1 + \dfrac{u}{m} \end{pmatrix}, \qquad y = x_1.$$

We see that this is a linear second-order differential equation with one input u and one output y.

We now wish to compute the response of the system to an input of the form $u = A\sin\omega t$. Although it is possible to solve for the response analytically, we instead make use of a computational approach that does not rely on the specific form of this system. Consider the general state space system

$$\frac{dx}{dt} = f(x,u).$$

Given the state x at time t, we can approximate the value of the state at a short time $h > 0$ later by assuming that the rate of change $f(x,u)$ is constant over the interval t to $t+h$. This gives

$$x(t+h) = x(t) + hf(x(t),u(t)). \tag{3.18}$$

Iterating this equation, we can thus solve for x as a function of time. This approximation is known as Euler integration and is in fact a difference equation if we let h represent the time increment and write $x[k] = x(kh)$, as we saw in Example 3.5. Although modern simulation tools such as MATLAB and Mathematica use more accurate methods than Euler integration, they still have some of the same basic trade-offs.

Returning to our specific example, Figure 3.11 shows the results of computing $x(t)$ using equation (3.18), along with the analytical computation. We see that as h gets smaller, the computed solution converges to the exact solution. The form of the solution is also worth noticing: after an initial transient, the system settles into a periodic motion. The portion of the response after the transient is called the *steady-state response* to the input.

In addition to generating simulations, models can also be used to answer other types of questions. Two that are central to the methods described in this text concern the stability of an equilibrium point and the input/output frequency response. We illustrate these two computations through the examples below and return to the general computations in later chapters.

Returning to the damped spring–mass system, the equations of motion with no

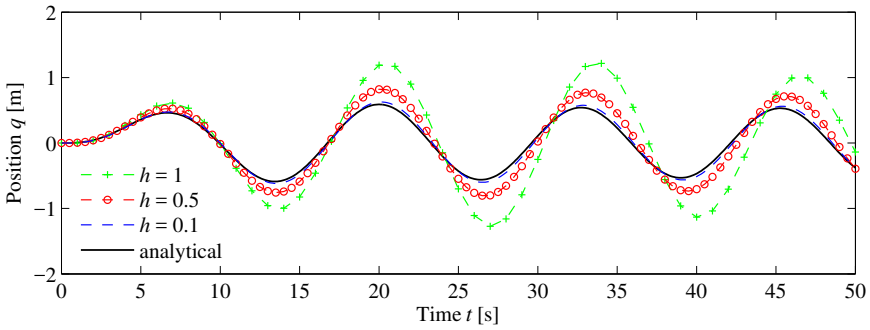


Figure 3.11: Simulation of the forced spring–mass system with different simulation time constants. The solid line represents the analytical solution. The dashed lines represent the approximate solution via the method of Euler integration, using decreasing step sizes.

input forcing are given by

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ -\frac{c}{m}x_2 - \frac{k}{m}x_1 \end{pmatrix}, \tag{3.19}$$

where x_1 is the position of the mass (relative to the rest position) and x_2 is its velocity. We wish to show that if the initial state of the system is away from the rest position, the system will return to the rest position eventually (we will later define this situation to mean that the rest position is *asymptotically stable*). While we could heuristically show this by simulating many, many initial conditions, we seek instead to prove that this is true for *any* initial condition.

To do so, we construct a function $V : \mathbb{R}^n \to \mathbb{R}$ that maps the system state to a positive real number. For mechanical systems, a convenient choice is the energy of the system,

$$V(x) = \frac{1}{2}kx_1^2 + \frac{1}{2}mx_2^2. \tag{3.20}$$

If we look at the time derivative of the energy function, we see that

$$\frac{dV}{dt} = kx_1\dot{x}_1 + mx_2\dot{x}_2 = kx_1x_2 + mx_2(-\frac{c}{m}x_2 - \frac{k}{m}x_1) = -cx_2^2,$$

which is always either negative or zero. Hence $V(x(t))$ is never increasing and, using a bit of analysis that we will see formally later, the individual states must remain bounded.

If we wish to show that the states eventually return to the origin, we must use a slightly more detailed analysis. Intuitively, we can reason as follows: suppose that for some period of time, $V(x(t))$ stops decreasing. Then it must be true that $\dot{V}(x(t)) = 0$, which in turn implies that $x_2(t) = 0$ for that same period. In that case, $\dot{x}_2(t) = 0$, and we can substitute into the second line of equation (3.19) to obtain

$$0 = \dot{x}_2 = -\frac{c}{m}x_2 - \frac{k}{m}x_1 = -\frac{k}{m}x_1.$$

Thus we must have that x_1 also equals zero, and so the only time that $V(x(t))$ can stop decreasing is if the state is at the origin (and hence this system is at its rest position). Since we know that $V(x(t))$ is never increasing (because $\dot{V} \leq 0$), we therefore conclude that the origin is stable (for *any* initial condition).

This type of analysis, called Lyapunov stability analysis, is considered in detail in Chapter 5. It shows some of the power of using models for the analysis of system properties.

Another type of analysis that we can perform with models is to compute the output of a system to a sinusoidal input, known as the *frequency response*. We again consider the spring–mass system, but this time keeping the input and leaving the system in its original form:

$$m\ddot{q} + c\dot{q} + kq = u. \tag{3.21}$$

We wish to understand how the system responds to a sinusoidal input of the form

$$u(t) = A \sin \omega t.$$

We will see how to do this analytically in Chapter 7, but for now we make use of simulations to compute the answer.

We first begin with the observation that if $q(t)$ is the solution to equation (3.21) with input $u(t)$, then applying an input $2u(t)$ will give a solution $2q(t)$ (this is easily verified by substitution). Hence it suffices to look at an input with unit magnitude, $A = 1$. A second observation, which we will prove in Chapter 6, is that the long-term response of the system to a sinusoidal input is itself a sinusoid at the same frequency, and so the output has the form

$$q(t) = g(\omega) \sin(\omega t + \varphi(\omega)),$$

where $g(\omega)$ is called the *gain* of the system and $\varphi(\omega)$ is called the *phase* (or phase offset).

To compute the frequency response numerically, we can simulate the system at a set of frequencies $\omega_1, \ldots, \omega_N$ and plot the gain and phase at each of these frequencies. An example of this type of computation is shown in Figure 3.12. For linear systems the frequency response does not depend on the amplitude A of the input signal. Frequency response can also be applied to nonlinear systems but the gain and phase then depend on the A.

3.3 Modeling Methodology

To deal with large, complex systems, it is useful to have different representations of the system that capture the essential features and hide irrelevant details. In all branches of science and engineering it is common practice to use some graphical description of systems, called *schematic diagrams*. They can range from stylistic pictures to drastically simplified standard symbols. These pictures make it possible to get an overall view of the system and to identify the individual components.

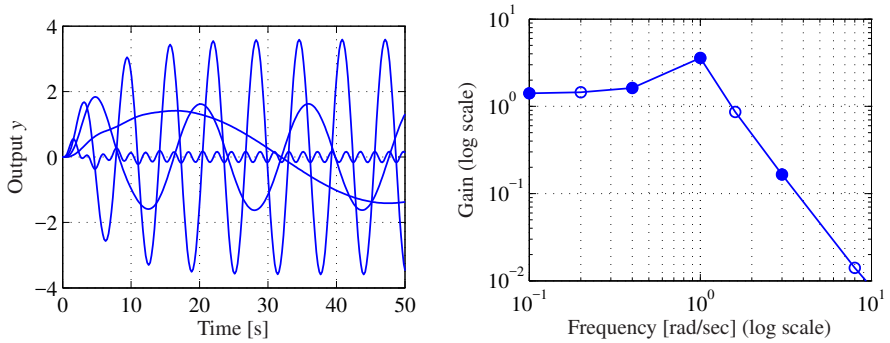
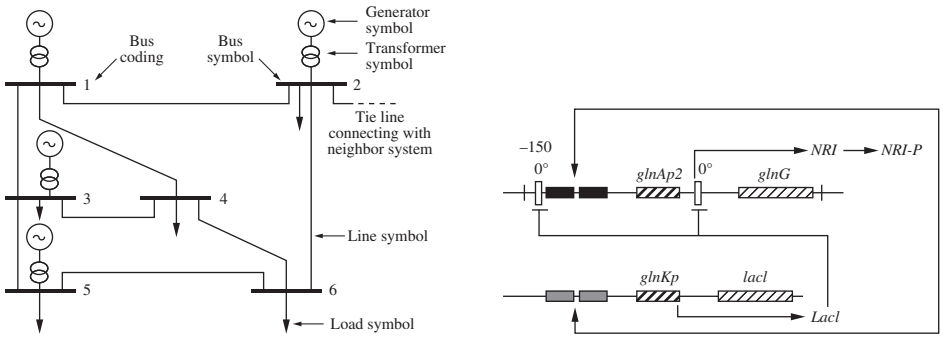
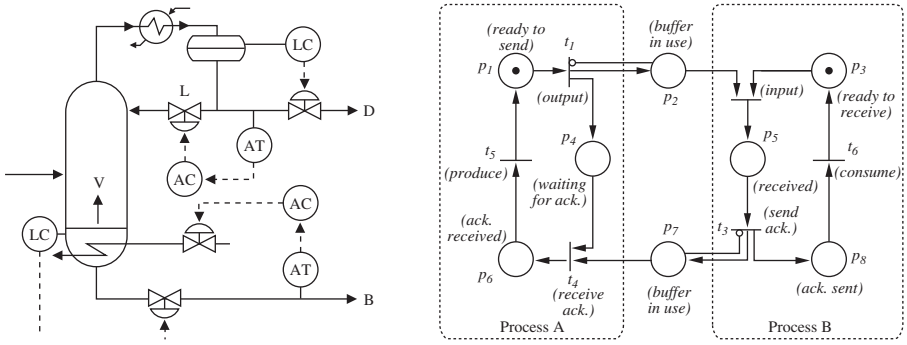


Figure 3.12: A frequency response (gain only) computed by measuring the response of individual sinusoids. The figure on the left shows the response of the system as a function of time to a number of different unit magnitude inputs (at different frequencies). The figure on the right shows this same data in a different way, with the magnitude of the response plotted as a function of the input frequency. The filled circles correspond to the particular frequencies shown in the time responses.



(a) Power electronics (b) Cell biology



(c) Process control (d) Networking

Figure 3.13: Schematic diagrams for different disciplines. Each diagram is used to illustrate the dynamics of a control system: (a) electrical schematics for a power system [Kun93], (b) a biological circuit diagram for a synthetic clock circuit [ASMN03], (c) a process diagram for a distillation column [SEM04] and (d) a Petri net description of a communication protocol.

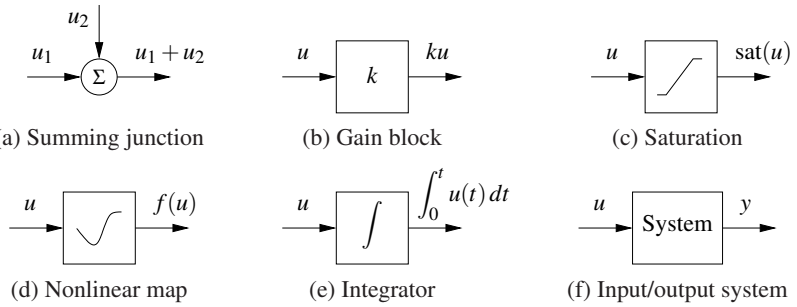


Figure 3.14: Standard block diagram elements. The arrows indicate the the inputs and outputs of each element, with the mathematical operation corresponding to the blocked labeled at the output. The system block (f) represents the full input/output response of a dynamical system.

Examples of such diagrams are shown in Figure 3.13. Schematic diagrams are useful because they give an overall picture of a system, showing different subprocesses and their interconnection and indicating variables that can be manipulated and signals that can be measured.

Block Diagrams

A special graphical representation called a *block diagram* has been developed in control engineering. The purpose of a block diagram is to emphasize the information flow and to hide details of the system. In a block diagram, different process elements are shown as boxes, and each box has inputs denoted by lines with arrows pointing toward the box and outputs denoted by lines with arrows going out of the box. The inputs denote the variables that influence a process, and the outputs denote the signals that we are interested in or signals that influence other subsystems. Block diagrams can also be organized in hierarchies, where individual blocks may themselves contain more detailed block diagrams.

Figure 3.14 shows some of the notation that we use for block diagrams. Signals are represented as lines, with arrows to indicate inputs and outputs. The first diagram is the representation for a summation of two signals. An input/output response is represented as a rectangle with the system name (or mathematical description) in the block. Two special cases are a proportional gain, which scales the input by a multiplicative factor, and an integrator, which outputs the integral of the input signal.

Figure 3.15 illustrates the use of a block diagram, in this case for modeling the flight response of a fly. The flight dynamics of an insect are incredibly intricate, involving careful coordination of the muscles within the fly to maintain stable flight in response to external stimuli. One known characteristic of flies is their ability to fly upwind by making use of the optical flow in their compound eyes as a feedback mechanism. Roughly speaking, the fly controls its orientation so that the point of contraction of the visual field is centered in its visual field.

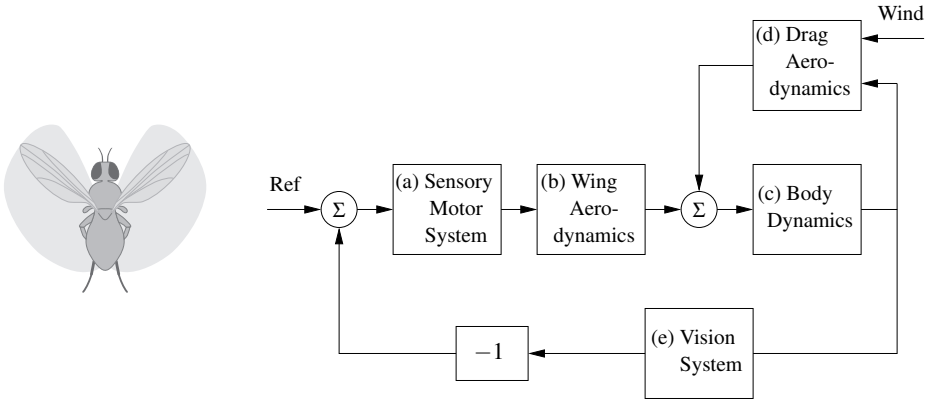


Figure 3.15: A block diagram representation of the flight control system for an insect flying against the wind. The mechanical portion of the model consists of the rigid-body dynamics of the fly, the drag due to flying through the air and the forces generated by the wings. The motion of the body causes the visual environment of the fly to change, and this information is then used to control the motion of the wings (through the sensory motor system), closing the loop.

To understand this complex behavior, we can decompose the overall dynamics of the system into a series of interconnected subsystems (or *blocks*). Referring to Figure 3.15, we can model the insect navigation system through an interconnection of five blocks. The sensory motor system (a) takes the information from the visual system (e) and generates muscle commands that attempt to steer the fly so that the point of contraction is centered. These muscle commands are converted into forces through the flapping of the wings (b) and the resulting aerodynamic forces that are produced. The forces from the wings are combined with the drag on the fly (d) to produce a net force on the body of the fly. The wind velocity enters through the drag aerodynamics. Finally, the body dynamics (c) describe how the fly translates and rotates as a function of the net forces that are applied to it. The insect position, speed and orientation are fed back to the drag aerodynamics and vision system blocks as inputs.

Each of the blocks in the diagram can itself be a complicated subsystem. For example, the visual system of a fruit fly consists of two complicated compound eyes (with about 700 elements per eye), and the sensory motor system has about 200,000 neurons that are used to process information. A more detailed block diagram of the insect flight control system would show the interconnections between these elements, but here we have used one block to represent how the motion of the fly affects the output of the visual system, and a second block to represent how the visual field is processed by the fly's brain to generate muscle commands. The choice of the level of detail of the blocks and what elements to separate into different blocks often depends on experience and on the questions that one wants to answer using the model. One of the powerful features of block diagrams is their ability to hide information about the details of a system that may not be needed to gain an understanding of the essential dynamics of the system.

Modeling from Experiments

Since control systems are provided with sensors and actuators, it is also possible to obtain models of system dynamics from experiments on the process. The models are restricted to input/output models since only these signals are accessible to experiments, but modeling from experiments can also be combined with modeling from physics through the use of feedback and interconnection.

A simple way to determine a system's dynamics is to observe the response to a step change in the control signal. Such an experiment begins by setting the control signal to a constant value; then when steady state is established, the control signal is changed quickly to a new level and the output is observed. The experiment gives the step response of the system, and the shape of the response gives useful information about the dynamics. It immediately gives an indication of the response time, and it tells if the system is oscillatory or if the response is monotone.

Example 3.7 Spring–mass system
The dynamics of the spring–mass system in Section 3.1 are given by

$$m\ddot{q} + c\dot{q} + kq = u. \tag{3.22}$$

We wish to determine the constants m, c and k by measuring the response of the system to a step input of magnitude F_0.

We will show in Chapter 7 that when $c^2 < 4km$, the step response for this system from the rest configuration is given by

$$q(t) = \frac{F_0}{k}\left(1 - \frac{1}{\omega_d}\sqrt{\frac{k}{m}}\exp\left(-\frac{ct}{2m}\right)\sin(\omega_d t + \varphi)\right),$$

$$\omega_d = \frac{\sqrt{4km - c^2}}{2m}, \qquad \varphi = \tan^{-1}\left(\frac{\sqrt{4km - c^2}}{c}\right).$$

From the form of the solution, we see that the shape of the step response is determined by the parameters of the system. Hence, by measuring certain features of the step response we can determine the parameter values.

Figure 3.16 shows the response of the system to a step of magnitude $F_0 = 20$ N, along with some measurements. We start by noting that the steady-state position of the mass (after the oscillations die down) is a function of the spring constant k:

$$q(\infty) = \frac{F_0}{k}, \tag{3.23}$$

where F_0 is the magnitude of the applied force ($F_0 = 1$ for a unit step input). The parameter $1/k$ is called the *gain* of the system. The period of the oscillation can be measured between two peaks and must satisfy

$$\frac{2\pi}{T} = \frac{\sqrt{4km - c^2}}{2m}. \tag{3.24}$$

Finally, the rate of decay of the oscillations is given by the exponential factor in

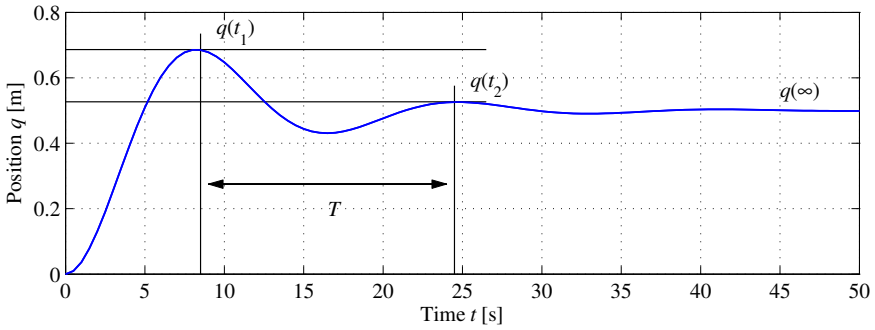


Figure 3.16: Step response for a spring–mass system. The magnitude of the step input is $F_0 = 20$ N. The period of oscillation T is determined by looking at the time between two subsequent local maxima in the response. The period combined with the steady-state value $q(\infty)$ and the relative decrease between local maxima can be used to estimate the parameters in a model of the system.

the solution. Measuring the amount of decay between two peaks, we have

$$\log\left(q(t_1) - \frac{F_0}{k}\right) - \log\left(q(t_2) - \frac{F_0}{k}\right) = \frac{c}{2m}(t_2 - t_1). \tag{3.25}$$

Using this set of three equations, we can solve for the parameters and determine that for the step response in Figure 3.16 we have $m \approx 250$ kg, $c \approx 60$ N s/m and $k \approx 40$ N/m. ∇

Modeling from experiments can also be done using many other signals. Sinusoidal signals are commonly used (particularly for systems with fast dynamics) and precise measurements can be obtained by exploiting correlation techniques. An indication of nonlinearities can be obtained by repeating experiments with input signals having different amplitudes. Modeling based on sinusoidal signals is very time consuming for systems with slow dynamics. In such situations it is advantageous to used signals that switch between two different levels. There is a whole subfield of control called *system identification* that deals with experimental determination of models. Questions like optimal inputs, experiments in open and closed loop, model accuracy and fundamental limitations are dealt with extensively.

Normalization and Scaling

When deriving a model, it is often useful to introduce dimension-free variables. Such a procedure can often simplify the equations for a system by reducing the number of parameters. It can also reveal interesting properties of the model. It is also useful to normalize variables by scaling to improve numerics and allow faster and more accurate simulations.

The procedure of scaling is straightforward in principle: choose units for each independent variable and introduce new variables by dividing the variables by the chosen normalization unit. We illustrate the procedure with two examples.

Example 3.8 Spring–mass system
Consider again the spring–mass system introduced earlier. Neglecting the damping, the system is described by

$$m\ddot{q} + kq = u.$$

The model has two parameters m and k. To normalize the model we introduce dimension-free variables $x = q/l$ and $\tau = \omega_0 t$, where $\omega_0 = \sqrt{k/m}$ and l is the chosen length scale. We scale force by $ml\omega_0^2$ and introduce $v = u/(ml\omega_0^2)$. The scaled equation then becomes

$$\frac{d^2x}{d\tau^2} = \frac{d^2q/l}{d(\omega_0 t)^2} = \frac{1}{ml\omega_0^2}(-kq+u) = -x+v,$$

which is the normalized undamped spring–mass system. Notice that the normalized model has no parameters, while the original model had two parameters m and k. Introducing the scaled, dimension-free state variables $z_1 = x = q/l$ and $z_2 = dx/d\tau = \dot{q}/(l\omega_0)$, the model can be written as

$$\frac{d}{d\tau}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} 0 \\ v \end{pmatrix}.$$

This simple linear equation describes the dynamics of any spring–mass system, independent of the particular parameters, and hence gives us insight into the fundamental dynamics of this oscillatory system. To recover the physical frequency of oscillation or its magnitude, we must invert the scaling we have applied. ∇

Example 3.9 Balance system
Consider the balance system described in Example 3.1. Neglecting damping by putting $c = 0$ and $\gamma = 0$ in equation (3.10), the model can be written as

$$(M+m)\frac{d^2p}{dt^2} - ml\cos\theta\frac{d^2\theta}{dt^2} + ml\sin\theta\left(\frac{d\theta}{dt}\right)^2 = F,$$
$$-ml\cos\theta\frac{d^2p}{dt^2} + (J+ml^2)\frac{d^2\theta}{dt^2} - mgl\sin\theta = 0.$$

Let $\omega_0 = \sqrt{mgl/(J+ml^2)}$, choose the length scale as l, let the time scale be $1/\omega_0$, choose the force scale as $(M+m)l\omega_0^2$ and introduce the scaled variables $\tau = \omega_0 t$, $x = p/l$ and $u = F/((M+m)l\omega_0^2)$. The equations then become

$$\frac{d^2x}{d\tau^2} - \alpha\cos\theta\frac{d^2\theta}{d\tau^2} + \alpha\sin\theta\left(\frac{d\theta}{d\tau}\right)^2 = u, \quad -\beta\cos\theta\frac{d^2x}{d\tau^2} + \frac{d^2\theta}{d\tau^2} - \sin\theta = 0,$$

where $\alpha = m/(M+m)$ and $\beta = ml^2/(J+ml^2)$. Notice that the original model has five parameters m, M, J, l and g but the normalized model has only two parameters α and β. If $M \gg m$ and $ml^2 \gg J$, we get $\alpha \approx 0$ and $\beta \approx 1$ and the model can be approximated by

$$\frac{d^2x}{d\tau^2} = u, \qquad \frac{d^2\theta}{d\tau^2} - \sin\theta = u\cos\theta.$$

The model can be interpreted as a mass combined with an inverted pendulum driven by the same input. ∇

For large systems scaling is not so easy: there are many choices and good selection of variables and normalization units require good understanding of the physics of the system and the numerical methods that will be used for analysis, scaling of large systems is therefor still an art.

3.4 Modeling Examples

In this section we introduce additional examples that illustrate some of the different types of systems for which one can develop differential equation and difference equation models. These examples are specifically chosen from a range of different fields to highlight the broad variety of systems to which feedback and control concepts can be applied. A more detailed set of applications that serve as running examples throughout the text are given in Chapter 4.

Motion Control Systems

Motion control systems involve the use of computation and feedback to control the movement of a mechanical system. Motion control systems range from nanopositioning systems (atomic force microscopes, adaptive optics), to control systems for the read/write heads in a disk drive of a DVD player, to manufacturing systems (transfer machines and industrial robots), to automotive control systems (antilock brakes, suspension control, traction control), to air and space flight control systems (airplanes, satellites, rockets and planetary rovers).

Example 3.10 Vehicle steering
A common problem in motion control is to control the trajectory of a vehicle through an actuator that causes a change in the orientation. A steering wheel on an automobile and the front wheel of a bicycle are two examples, but similar dynamics occur in the steering of ships or control of the pitch dynamics of an aircraft. In many cases, we can understand the basic behavior of these systems through the use of a simple model that captures the basic kinematics of the system.

Consider a vehicle with two wheels as shown in Figure 3.17. For the purpose of steering we are interested in a model that describes how the velocity of the vehicle depends on the steering angle δ. To be specific, let b be the wheel base and consider the velocity v at the center of mass, a distance a from the rear wheel, as shown in Figure 3.17. Let x and y be the coordinates of the center of mass, θ the heading angle and α the angle between the velocity vector v and the centerline of the vehicle. The point O is at the intersection of the normals to the front and rear wheels. The distance from O to the center of mass is r_a.

Assuming no slipping of the wheels the motion is a rotation around the point O in the figure. Since $b = r_a \tan\delta$ and $a = r_a \tan\alpha$, it follows that $\tan\alpha = (a/b)\tan\delta$

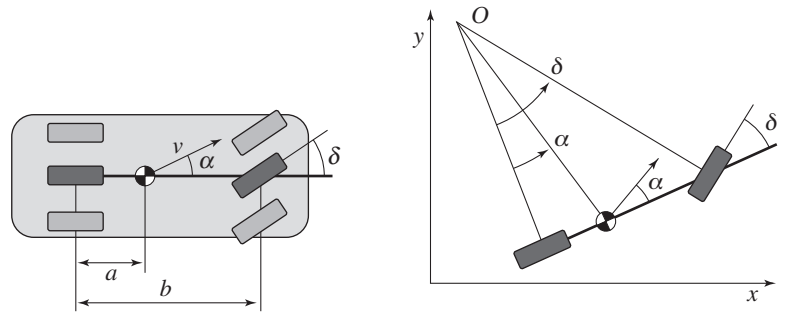


Figure 3.17: Vehicle steering dynamics. The left figure shows an overhead view of a vehicle with four wheels. The wheel base is b and the center of mass at a distance a forward of the rear wheels. By approximating the motion of the front and rear pairs of wheels by a single front wheel and a single rear wheel, we obtain an abstraction called the *bicycle model*, shown on the right. The steering angle is δ and the velocity at the center of mass has the angle α relative the length axis of the vehicle. The position of the vehicle is given by (x,y) and the orientation (heading) by θ.

and we get the following relation between α and the steering angle δ:

$$\alpha = \arctan\left(\frac{a\tan\delta}{b}\right). \tag{3.26}$$

Assume that the wheels are rolling without slip and that the velocity of the rear wheel is v_0. The vehicle speed at its center of mass is $v = v_0/\cos\alpha$, and we find that the motion of this point is given by

$$\begin{aligned}\frac{dx}{dt} &= v\cos(\alpha+\theta) = v_0\frac{\cos(\alpha+\theta)}{\cos\alpha},\\ \frac{dy}{dt} &= v\sin(\alpha+\theta) = v_0\frac{\sin(\alpha+\theta)}{\cos\alpha}.\end{aligned} \tag{3.27}$$

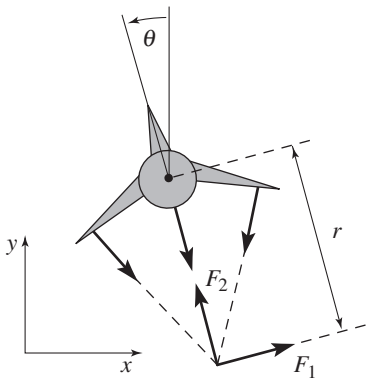
To see how the angle θ is influenced by the steering angle, we observe from Figure 3.17 that the vehicle rotates with the angular velocity v_0/r_a around the point O. Hence

$$\frac{d\theta}{dt} = \frac{v_0}{r_a} = \frac{v_0}{b}\tan\delta. \tag{3.28}$$

Equations (3.26)–(3.28) can be used to model an automobile under the assumptions that there is no slip between the wheels and the road and that the two front wheels can be approximated by a single wheel at the center of the car. This model is often called the *bicycle model*. The assumption of no slip can be relaxed by adding an extra state variable, giving a more realistic model. Such a model also describes the steering dynamics of ships as well as the pitch dynamics of aircraft and missiles. It is also possible to choose coordinates so that the reference point is at the rear wheels (corresponding to setting $\alpha = 0$), a model often referred to as the *Dubins car* [Dub57] or the bicycle model.

Figure 3.17 represents the situation when the vehicle moves forward and has

(a) Harrier "jump jet"



(b) Simplified model

Figure 3.18: Vectored thrust aircraft. The Harrier AV-8B military aircraft (a) redirects its engine thrust downward so that it can "hover" above the ground. Some air from the engine is diverted to the wing tips to be used for maneuvering. As shown in (b), the net thrust on the aircraft can be decomposed into a horizontal force F_1 and a vertical force F_2 acting at a distance r from the center of mass.

front-wheel steering. The figure shows that the model also applies to rear wheel steering if the sign of the velocity is reversed. ∇

Example 3.11 Vectored thrust aircraft

Consider the motion of vectored thrust aircraft, such as the Harrier "jump jet" shown Figure 3.18a. The Harrier is capable of vertical takeoff by redirecting its thrust downward and through the use of smaller maneuvering thrusters located on its wings. A simplified model of the Harrier is shown in Figure 3.18b, where we focus on the motion of the vehicle in a vertical plane through the wings of the aircraft. We resolve the forces generated by the main downward thruster and the maneuvering thrusters as a pair of forces F_1 and F_2 acting at a distance r below the aircraft (determined by the geometry of the thrusters).

Let (x, y, θ) denote the position and orientation of the center of mass of the aircraft. Let m be the mass of the vehicle, J the moment of inertia, g the gravitational constant and c the damping coefficient. Then the equations of motion for the vehicle are given by

$$\begin{aligned} m\ddot{x} &= F_1 \cos\theta - F_2 \sin\theta - c\dot{x}, \\ m\ddot{y} &= F_1 \sin\theta + F_2 \cos\theta - mg - c\dot{y}, \\ J\ddot{\theta} &= rF_1. \end{aligned} \tag{3.29}$$

It is convenient to redefine the inputs so that the origin is an equilibrium point of the system with zero input. Letting $u_1 = F_1$ and $u_2 = F_2 - mg$, the equations become

$$\begin{aligned} m\ddot{x} &= -mg\sin\theta - c\dot{x} + u_1\cos\theta - u_2\sin\theta, \\ m\ddot{y} &= mg(\cos\theta - 1) - c\dot{y} + u_1\sin\theta + u_2\cos\theta, \\ J\ddot{\theta} &= ru_1. \end{aligned} \tag{3.30}$$

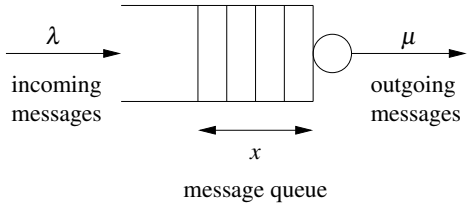


Figure 3.19: Schematic diagram of a queuing system. Messages arrive at rate λ and are stored in a queue. Messages are processed and removed from the queue at rate μ. The average length of the queue is given by $x \in \mathbb{R}$.

These equations describe the motion of the vehicle as a set of three coupled second-order differential equations. ∇

Information Systems

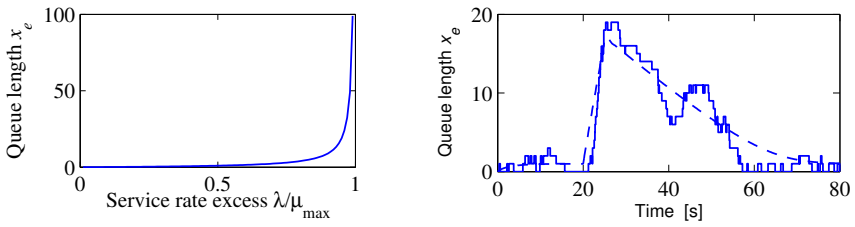
Information systems range from communication systems like the Internet to software systems that manipulate data or manage enterprise-wide resources. Feedback is present in all these systems, and designing strategies for routing, flow control and buffer management is a typical problem. Many results in queuing theory emerged from design of telecommunication systems and later from development of the Internet and computer communication systems [BG87, Kle75, Sch87]. Management of queues to avoid congestion is a central problem and we will therefore start by discussing the modeling of queuing systems.

Example 3.12 Queuing systems
A schematic picture of a simple queue is shown in Figure 3.19. Requests arrive and are then queued and processed. There can be large variations in arrival rates and service rates, and the queue length builds up when the arrival rate is larger than the service rate. When the queue becomes too large, service is denied using an admission control policy.

The system can be modeled in many different ways. One way is to model each incoming request, which leads to an event-based model where the state is an integer that represents the queue length. The queue changes when a request arrives or a request is serviced. The statistics of arrival and servicing are typically modeled as random processes. In many cases it is possible to determine statistics of quantities like queue length and service time, but the computations can be quite complicated.

A significant simplification can be obtained by approximating the discrete queue length by a continuous variable. Instead of keeping track of each request we instead view service and requests as continuous flows. The model obtained is called a *flow model* because of the analogy with fluid dynamics where motion of molecules are replace by continuous flows. Hence, if the queue length x is a continuous variable and the arrivals and services are flows with rates λ and μ, the system can be modeled by the first-order differential equation

$$\frac{dx}{dt} = \lambda - \mu = \lambda - \mu_{\max} f(x), \quad x \geq 0, \tag{3.31}$$



(a) Steady-state queue length (b) Overload condition

Figure 3.20: Queuing dynamics. (a) The steady-state queue length as a function of $\lambda/\mu_{\max}$. (b) The behavior of the queue length when there is a temporary overload in the system. The solid line shows a realization of an event-based simulation, and the dashed line shows the behavior of the flow model (3.32). The maximum service rate is $\mu_{\max} = 1$, and the arrival rate starts at $\lambda = 0.5$. The arrival rate is increased to $\lambda = 4$ at time 20, and it returns to $\lambda = 0.5$ at time 25.

proposed by Agnew [Agn76]. The service rate μ depends on the queue length; if there are no capacity restrictions we have $\mu = x/T$ where T is the time it takes to serve one customer. The service rate thus increases linearly with the queue length. In reality the growth will be slower because longer queues require more resources, and the service rate has an upper limit $\mu_{\max}$. These effects are captured by by modeling the service rate as $\mu_{\max} f(x)$ in equation (3.32). The function $f(x)$ is monotone, approximately linear for small x and $f(\infty) = 1$.

For a particular queue, the function can be determined empirically by measuring the queue length for different arrival and service rates. A simple choice is $f(x) = x/(1+x)$, which gives the model

$$\frac{dx}{dt} = \lambda - \mu_{\max}\frac{x}{x+1}. \tag{3.32}$$

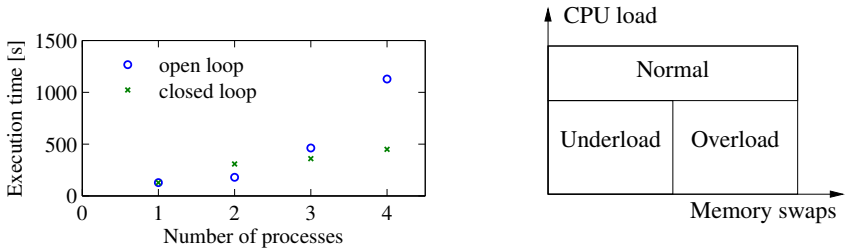
It was shown by Tipper [TS90], that if arrival and service processes are Poisson processes, then average queue length is given by equation (3.32). Furthermore the equation equation (3.32) is a good approximation even for short queue lengths.

To explore the properties of the model (3.32) we will first investigate the equilibrium value of the queue length when the arrival rate λ is constant. Setting the derivative dx/dt to zero in equation (3.32) and solving for x, we find that the queue length x approaches the steady-state value

$$x_e = \frac{\lambda}{\mu_{\max} - \lambda}. \tag{3.33}$$

Figure 3.20a shows the steady-state queue length as a function of $\lambda/\mu_{\max}$, the effective service rate excess. Notice that the queue length increases rapidly as λ approaches $\mu_{\max}$. To have a queue length less than 20 requires $\lambda/\mu_{\max} < 0.95$. The average time to service a request can be shown to be $T_s = (x+1)/\mu_{\max}$, and it increases dramatically as λ approaches $\mu_{\max}$.

Figure 3.20b illustrates the behavior of the server in a typical overload situation. The figure shows that the queue builds up quickly and clears very slowly. Since the



(a) System performance (b) System state

Figure 3.21: Illustration of feedback in the virtual memory system of the IBM/370. (a) The effect of feedback on execution times in a simulation, following [BG68]. Results with no feedback are shown with ○, and results with feedback with ×. Notice the dramatic decrease in execution time for the system with feedback. (b) How the three states are obtained based on process measurements.

response time is proportional to queue length, it means that the quality of service is poor for a long period after an overload. This behavior is called the *rush-hour effect* and has been observed in web servers and many other queuing systems such as automobile traffic.

The dashed line in Figure 3.20b shows the behavior of the flow model, which describes the average queue length. The simple model captures behavior qualitatively, but there are variations from sample to sample when the queue length is short. ∇

Many complex systems use discrete control actions. Such systems can be modeled by characterizing the situations that correspond to each control action, as illustrated in the following example.

Example 3.13 Virtual memory paging control

An early example of the use of feedback in computer systems was applied in the operating system OS/VS for the IBM 370 [BG68, Cro75]. The system used virtual memory, which allows programs to address more memory than is physically available as fast memory. Data in current fast memory (random access memory, RAM) is accessed directly, but data that resides in slower memory (disk) is automatically loaded into fast memory. The system is implemented in such a way that it appears to the programmer as a single large section of memory. The system performed very well in many situations, but very long execution times were encountered in overload situations, as shown by the open circles in Figure 3.21a. The difficulty was resolved with a simple discrete feedback system. The load of the central processing unit (CPU) was measured together with the number of page swaps between fast memory and slow memory. The operating region was classified as being in one of three states: normal, underload or overload. The normal state is characterized by high CPU activity, the underload state is characterized by low CPU activity and few page replacements, the overload state has moderate to low CPU load but many page replacements; see Figure 3.21b. The boundaries between the regions and the time for measuring the load were determined from simulations using typ-

ical loads. The control strategy was to do nothing in the normal load condition, to exclude a process from memory in the overload condition and to allow a new process or a previously excluded process in the underload condition. The crosses in Figure 3.21a show the effectiveness of the simple feedback system in simulated loads. Similar principles are used in many other situations, e.g., in fast, on-chip cache memory.

∇

Example 3.14 Consensus protocols in sensor networks

Sensor networks are used in a variety of applications where we want to collect and aggregate information over a region of space using multiple sensors that are connected together via a communications network. Examples include monitoring environmental conditions in a geographical area (or inside a building), monitoring the movement of animals or vehicles and monitoring the resource loading across a group of computers. In many sensor networks the computational resources are distributed along with the sensors, and it can be important for the set of distributed agents to reach a consensus about a certain property, such as the average temperature in a region or the average computational load among a set of computers.

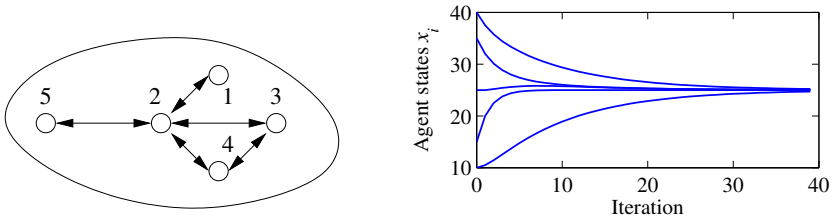
To illustrate how such a consensus might be achieved, we consider the problem of computing the average value of a set of numbers that are locally available to the individual agents. We wish to design a "protocol" (algorithm) such that all agents will agree on the average value. We consider the case in which all agents cannot necessarily communicate with each other directly, although we will assume that the communications network is connected (meaning that no two groups of agents are completely isolated from each other). Figure 3.22a shows a simple situation of this type.

We model the connectivity of the sensor network using a graph, with nodes corresponding to the sensors and edges corresponding to the existence of a direct communications link between two nodes. For any such graph, we can build an *adjacency matrix*, where each row and column of the matrix corresponds to a node and a 1 in the respective row and column indicates that the two nodes are connected. For the network shown in Figure 3.22a, the corresponding adjacency matrix is

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

We use the notation $\mathcal{N}_i$ to represent the set of neighbors of a node i. For example, in the network shown in Figure 3.22a $\mathcal{N}_2 = \{1,3,4,5\}$ and $\mathcal{N}_3 = \{2,4\}$.

To solve the consensus problem, let x_i be the state of the ith sensor, corresponding to that sensor's estimate of the average value that we are trying to compute. We initialize the state to the value of the quantity measured by the individual sensor.



(a) Sensor network (b) Consensus convergence

Figure 3.22: Consensus protocols for sensor networks. (a) A simple sensor network with five nodes. In this network, node 1 communicates with node 2 and node 2 communicates with nodes 1, 3, 4, 5, etc. (b) A simulation demonstrating the convergence of the consensus protocol (3.34) to the average value of the initial conditions.

The consensus protocol (algorithm) can now be realized as a local update law

$$x_i[k+1] = x_i[k] + \gamma \sum_{j \in \mathcal{N}_i} (x_j[k] - x_i[k]). \tag{3.34}$$

This protocol attempts to compute the average by updating the local state of each agent based on the value of its neighbors. The combined dynamics of all agents can be written in the form

$$x[k+1] = x[k] - \gamma(D-A)x[k], \tag{3.35}$$

where A is the adjacency matrix and D is a diagonal matrix with entries corresponding to the number of neighbors of each node. The constant γ describes the rate at which the estimate of the average is updated based on information from neighboring nodes. The matrix $L := D - A$ is called the *Laplacian* of the graph.

The equilibrium points of equation (3.35) are the set of states such that $x_e[k+1] = x_e[k]$. It can be shown that if the network is connected, $x_e = (\alpha, \alpha, \ldots, \alpha)$ is an equilibrium state for the system, corresponding to each sensor having an identical estimate α for the average. Furthermore, we can show that α is indeed the average value of the initial states. Since there can be cycles in the graph, it is possible that the state of the system could enter into an infinite loop and never converge to the desired consensus state. A formal analysis requires tools that will be introduced later in the text, but it can be shown that for any connected graph we can always find a γ such that the states of the individual agents converge to the average. A simulation demonstrating this property is shown in Figure 3.22b. Although we have focused here on consensus to the average value of a set of measurements, other consensus states can be achieved through choice of appropriate feedback laws. Examples include finding the maximum or minimum value in a network, counting the number of nodes in a network or computing higher-order statistical moments of a distributed quantity [OSFM07]. ∇

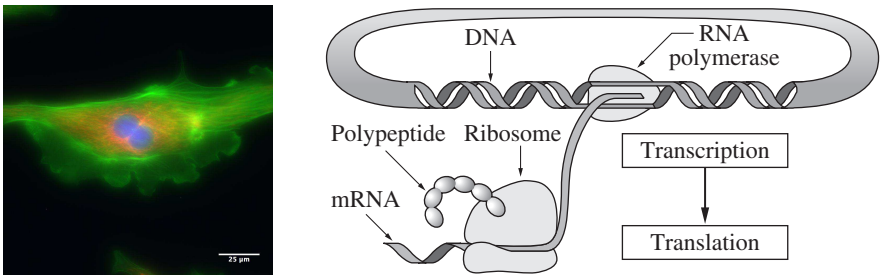


Figure 3.23: Biological circuitry. The cell on the left is a bovine pulmonary cell, stained so that the nucleus, actin and chromatin are visible. The figure on the right gives an overview of the process by which proteins in the cell are made. RNA is transcribed from DNA by an RNA polymerase enzyme. The RNA is then translated into a polypeptide chain by a molecular machine called a ribosome, and then the polypeptide chain folds into a protein molecule.

Biological Systems

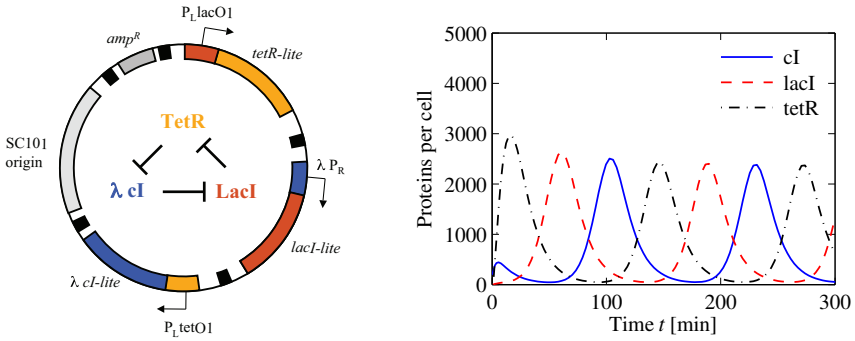
Biological systems provide perhaps the richest source of feedback and control examples. The basic problem of homeostasis, in which a quantity such as temperature or blood sugar level is regulated to a fixed value, is but one of the many types of complex feedback interactions that can occur in molecular machines, cells, organisms and ecosystems.

Example 3.15 Transcriptional regulation
Transcription is the process by which messenger RNA (mRNA) is generated from a segment of DNA. The promoter region of a gene allows transcription to be controlled by the presence of other proteins, called *transcription factors*, which bind to the promoter region and either repress or activate RNA polymerase, the enzyme that produces an mRNA transcript from DNA. The mRNA is then translated into a protein according to its nucleotide sequence. This process is illustrated in Figure 3.23.

A simple model of the transcriptional regulation process is through the use of a Hill function [dJ02, Mur04]. Consider the regulation of a protein A with a concentration given by p_a and a corresponding mRNA concentration m_a. Let B be a second protein with concentration p_b that represses the production of protein A through transcriptional regulation. The resulting dynamics of p_a and m_a can be written as

$$\frac{dm_a}{dt} = \frac{\alpha_{ab}}{1 + k_{ab} p_b^{n_{ab}}} + \alpha_{a0} - \gamma_a m_a, \qquad \frac{dp_a}{dt} = \beta_a m_a - \delta_a p_a, \tag{3.36}$$

where $\alpha_{ab} + \alpha_{a0}$ is the unregulated transcription rate, γ_a represents the rate of degradation of mRNA, α_{ab}, k_{ab} and n_{ab} are parameters that describe how B represses A, β_a represents the rate of production of the protein from its corresponding mRNA and δ_a represents the rate of degradation of the protein A. The parameter α_{a0} describes the "leakiness" of the promoter, and n_{ab} is called the Hill coefficient and relates to the cooperativity of the promoter.



(a) Repressilator plasmid (b) Repressilator simulation

Figure 3.24: The repressilator genetic regulatory network. (a) A schematic diagram of the repressilator, showing the layout of the genes in the plasmid that holds the circuit as well as the circuit diagram (center). (b) A simulation of a simple model for the repressilator, showing the oscillation of the individual protein concentrations. (Figure courtesy M. Elowitz.)

A similar model can be used when a protein activates the production of another protein rather than repressing it. In this case, the equations have the form

$$\frac{dm_a}{dt} = \frac{\alpha_{ab} k_{ab} p_b^{n_{ab}}}{1 + k_{ab} p_b^{n_{ab}}} + \alpha_{a0} - \gamma_a m_a, \qquad \frac{dp_a}{dt} = \beta_a m_a - \delta_a p_a, \tag{3.37}$$

where the variables are the same as described previously. Note that in the case of the activator, if p_b is zero, then the production rate is $\alpha_{a0} \ll \alpha_{ab}$ (versus $\alpha_{ab} + \alpha_{a0}$ for the repressor). As p_b gets large, the first term in the expression for $\dot{m}_a$ approaches 1 and the transcription rate becomes $\alpha_{ab} + \alpha_{a0}$ (versus α_{a0} for the repressor). Thus we see that the activator and repressor act in opposite fashion from each other.

As an example of how these models can be used, we consider the model of a "repressilator," originally due to Elowitz and Leibler [EL00]. The repressilator is a synthetic circuit in which three proteins each repress another in a cycle. This is shown schematically in Figure 3.24a, where the three proteins are TetR, λ cI and LacI. The basic idea of the repressilator is that if TetR is present, then it represses the production of λ cI. If λ cI is absent, then LacI is produced (at the unregulated transcription rate), which in turn represses TetR. Once TetR is repressed, then λ cI is no longer repressed, and so on. If the dynamics of the circuit are designed properly, the resulting protein concentrations will oscillate.

We can model this system using three copies of equation (3.36), with A and B replaced by the appropriate combination of TetR, cI and LacI. The state of the system is then given by $x = (m_{\text{TetR}}, p_{\text{TetR}}, m_{\text{cI}}, p_{\text{cI}}, m_{\text{LacI}}, p_{\text{LacI}})$. Figure 3.24b shows the traces of the three protein concentrations for parameters $n = 2$, $\alpha = 0.5$, $k = 6.25 \times 10^{-4}$, $\alpha_0 = 5 \times 10^{-4}$, $\gamma = 5.8 \times 10^{-3}$, $\beta = 0.12$ and $\delta = 1.2 \times 10^{-3}$ with initial conditions $x(0) = (1, 200, 0, 0, 0, 0)$ (following [EL00]). ∇

Example 3.16 Wave propagation in neuronal networks

The dynamics of the membrane potential in a cell are a fundamental mechanism in understanding signaling in cells, particularly in neurons and muscle cells. The Hodgkin–Huxley equations give a simple model for studying propagation waves in networks of neurons. The model for a single neuron has the form

$$C\frac{dV}{dt} = -I_{\text{Na}} - I_{\text{K}} - I_{\text{leak}} + I_{\text{input}},$$

where V is the membrane potential, C is the capacitance, I_{Na} and I_{K} are the current caused by the transport of sodium and potassium across the cell membrane, I_{leak} is a leakage current and I_{input} is the external stimulation of the cell. Each current obeys Ohm's law, i.e.,

$$I = g(V - E),$$

where g is the conductance, which is different for different ions, and E is the equilibrium voltage. The equilibrium voltage is given by Nernst's law,

$$E = \frac{RT}{nF}\log\frac{c_e}{c_i},$$

where R is Boltzmann's constant, T is the absolute temperature, F is Faraday's constant, n is the charge (or valence) of the ion and c_i and c_e are the ion concentrations inside the cell and in the external fluid. At 20 °C we have $RT/F = 20$ mV.

The Hodgkin–Huxley model was originally developed as a means to predict the quantitative behavior of the squid giant axon [HH52]. Hodgkin and Huxley shared the 1963 Nobel Prize in Physiology (along with J. C. Eccles) for analysis of the electrical and chemical events in nerve cell discharges. The voltage clamp described in Section 1.4 was a key element in Hodgkin and Huxley's experiments. ∇

3.5 Further Reading

Modeling is ubiquitous in engineering and science and has a long history in applied mathematics. For example, the Fourier series was introduced by Fourier when he modeled heat conduction in solids [Fou07]. Models of dynamics have been developed in many different fields, including mechanics [Arn78, Gol53], heat conduction [CJ59], fluids [BRS60], vehicles [Abk69, Bla91, Ell94], robotics [MLS94, SV89], circuits [Gui63], power systems [Kun93], acoustics [Ber54] and micromechanical systems [Sen01]. Control theory requires modeling from many different domains, and most control theory texts contain several chapters on modeling using ordinary differential equations and difference equations (see, for example, [FPEN05]). A classic book on the modeling of physical systems, especially mechanical, electrical and thermofluid systems, is Cannon [Can03]. The book by Aris [Ari94] is highly original and has a detailed discussion of the use of dimension-free variables. Two of the authors' favorite books on modeling of biological systems are J. D. Murray [Mur04] and Wilson [Wil99]. A good source for system identification in Ljung [Lju99].

Exercises

3.1 (Chain of integrators form) Consider the linear ordinary differential equation (3.8). Show that by choosing a state space representation with $x_1 = y$, the dynamics can be written as

$$A = \begin{pmatrix} 0 & 1 & & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & & -a_1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & \dots & 0 & 0 \end{pmatrix}.$$

This canonical form is called the *chain of integrators* form.

3.2 (Inverted pendulum) Use the equations of motion for a balance system to derive a dynamic model for the inverted pendulum described in Example 3.2 and verify that the dynamics are given by equation (3.11).

3.3 (Discrete-time dynamics) Consider the following discrete-time system

$$x[k+1] = Ax[k] + Bu[k], \qquad y[k] = Cx[k],$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \qquad A = \begin{pmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

In this problem, we will explore some of the properties of this discrete-time system as a function of the parameters, the initial conditions and the inputs.

(a) For the case when $a_{12} = 0$ and $u = 0$, give a closed form expression for the output of the system.

(b) A discrete system is in *equilibrium* when $x[k+1] = x[k]$ for all k. Let $u = r$ be a constant input and compute the resulting equilibrium point for the system. Show that if $|a_{ii}| < 1$ for all i, all initial conditions give solutions that converge to the equilibrium point.

(c) Write a computer program to plot the output of the system in response to a unit step input, $u[k] = 1$, $k \geq 0$. Plot the response of your system with $x[0] = 0$ and A given by $a_{11} = 0.5$, $a_{12} = 1$ and $a_{22} = 0.25$.

3.4 (Keynesian economics) Keynes' simple model for an economy is given by

$$Y[k] = C[k] + I[k] + G[k],$$

where Y, C, I and G are gross national product (GNP), consumption, investment and government expenditure for year k. Consumption and investment are modeled by difference equations of the form

$$C[k+1] = aY[k], \qquad I[k+1] = b(C[k+1] - C[k]),$$

where a and b are parameters. The first equation implies that consumption increases with GNP but that the effect is delayed. The second equation implies that investment is proportional to the rate of change of consumption.

Show that the equilibrium value of the GNP is given by

$$Y_e = \frac{1}{1-a} G_e,$$

where the parameter $1/(1-a)$ is the Keynes multiplier (the gain from G to Y). With $a = 0.75$ an increase of government expenditure will result in a fourfold increase of GNP. Also show that the model can be written as the following discrete-time state model:

$$\begin{pmatrix} C[k+1] \\ I[k+1] \end{pmatrix} = \begin{pmatrix} a & a \\ ab-b & ab \end{pmatrix} \begin{pmatrix} C[k] \\ I[k] \end{pmatrix} + \begin{pmatrix} a \\ ab \end{pmatrix} G[k],$$
$$Y[k] = C[k] + I[k] + G[k].$$

3.5 (Least squares system identification) Consider a nonlinear differential equation that can be written in the form

$$\frac{dx}{dt} = \sum_{i=1}^{M} \alpha_i f_i(x),$$

where $f_i(x)$ are known nonlinear functions and α_i are unknown, but constant, parameters. Suppose that we have measurements (or estimates) of the full state x at time instants $t_1, t_2, \ldots, t_N$, with $N > M$. Show that the parameters α_i can be estimated by finding the least squares solution to a linear equation of the form

$$H\alpha = b,$$

where $\alpha \in \mathbb{R}^M$ is the vector of all parameters and $H \in \mathbb{R}^{N \times M}$ and $b \in \mathbb{R}^N$ are appropriately defined.

3.6 (Normalized oscillator dynamics) Consider a damped spring–mass system with dynamics

$$m\ddot{q} + c\dot{q} + kq = F.$$

Let $\omega_0 = \sqrt{k/m}$ be the natural frequency and $\zeta = c/(2\sqrt{km})$ be the damping ratio.

(a) Show that by rescaling the equations, we can write the dynamics in the form

$$\ddot{q} + 2\zeta\omega_0\dot{q} + \omega_0^2 q = \omega_0^2 u, \tag{3.38}$$

where $u = F/k$. This form of the dynamics is that of a linear oscillator with natural frequency ω_0 and damping ratio ζ.

(b) Show that the system can be further normalized and written in the form

$$\frac{dz_1}{d\tau} = z_2, \qquad \frac{dz_2}{d\tau} = -z_1 - 2\zeta z_2 + v. \tag{3.39}$$

The essential dynamics of the system are governed by a single damping parameter ζ. The *Q-value* defined as $Q = 1/2\zeta$ is sometimes used instead of ζ.

3.7 (Electric generator) An electric generator connected to a strong power grid can be modeled by a momentum balance for the rotor of the generator:

$$J\frac{d^2\varphi}{dt^2} = P_m - P_e = P_m - \frac{EV}{X}\sin\varphi,$$

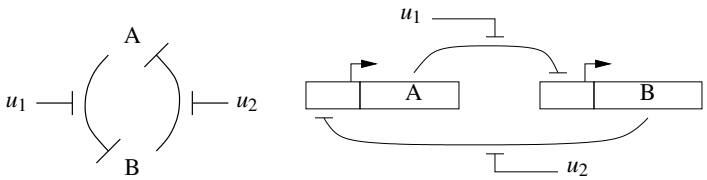
where J is the effective moment of inertia of the generator, φ the angle of rotation, P_m the mechanical power that drives the generator, P_e is the active electrical power, E the generator voltage, V the grid voltage and X the reactance of the line. Assuming that the line dynamics are much faster than the rotor dynamics, $P_e = VI = (EV/X)\sin\varphi$, where I is the current component in phase with the voltage E and φ is the phase angle between voltages E and V. Show that the dynamics of the electric generator has a normalized form that is similar to the dynamics of a pendulum with forcing at the pivot.

3.8 (Admission control for a queue) Consider the queuing system described in Example 3.12. The long delays created by temporary overloads can be reduced by rejecting requests when the queue gets large. This allows requests that are accepted to be serviced quickly and requests that cannot be accommodated to receive a rejection quickly so that they can try another server. Consider an admission control system described by

$$\frac{dx}{dt} = \lambda u - \mu_{\max}\frac{x}{x+1}, \qquad u = \mathrm{sat}_{(0,1)}(k(r-x)), \tag{3.40}$$

where the controller is a simple proportional control with saturation ($\mathrm{sat}_{(a,b)}$ is defined by equation (4.9)) and r is the desired (reference) queue length. Use a simulation to show that this controller reduces the rush-hour effect and explain how the choice of r affects the system dynamics.

3.9 (Biological switch) A genetic switch can be formed by connecting two repressors together in a cycle as shown below.

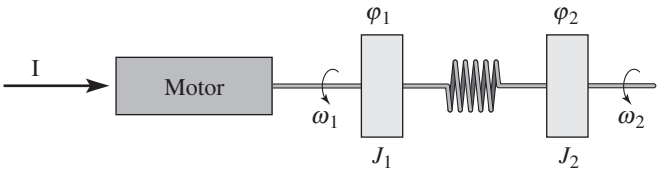


Using the models from Example 3.15—assuming that the parameters are the same for both genes and that the mRNA concentrations reach steady state quickly—show that the dynamics can be written in normalized coordinates as

$$\frac{dz_1}{d\tau} = \frac{\mu}{1+z_2^n} - z_1 - v_1, \qquad \frac{dz_2}{d\tau} = \frac{\mu}{1+z_1^n} - z_2 - v_2, \tag{3.41}$$

where z_1 and z_2 are scaled versions of the protein concentrations and the time scale has also been changed. Show that $\mu \approx 200$ using the parameters in Example 3.15, and use simulations to demonstrate the switch-like behavior of the system.

3.10 (Motor drive) Consider a system consisting of a motor driving two masses that are connected by a torsional spring, as shown in the diagram below.



This system can represent a motor with a flexible shaft that drives a load. Assuming that the motor delivers a torque that is proportional to the current I, the dynamics of the system can be described by the equations

$$\begin{aligned} J_1\frac{d^2\varphi_1}{dt^2} + c\left(\frac{d\varphi_1}{dt} - \frac{d\varphi_2}{dt}\right) + k(\varphi_1 - \varphi_2) &= k_I I, \\ J_2\frac{d^2\varphi_2}{dt^2} + c\left(\frac{d\varphi_2}{dt} - \frac{d\varphi_1}{dt}\right) + k(\varphi_2 - \varphi_1) &= T_d, \end{aligned} \tag{3.42}$$

where φ_1 and φ_2 are the angles of the two masses, $\omega_i = d\varphi_i/dt$ are their velocities, J_i represents moments of inertia, c is the damping coefficient, k represents the shaft stiffness, k_I is the torque constant for the motor, and T_d is the disturbance torque applied at the end of the shaft. Similar equations are obtained for a robot with flexible arms and for the arms of DVD and optical disk drives.

Derive a state space model for the system by introducing the (normalized) state variables $x_1 = \varphi_1, x_2 = \varphi_2, x_3 = \omega_1/\omega_0$, and $x_4 = \omega_2/\omega_0$, where $\omega_0 = \sqrt{k(J_1 + J_2)/(J_1 J_2)}$ is the undamped natural frequency of the system when the control signal is zero.

Chapter Four
Examples

> *... Don't apply any model until you understand the simplifying assumptions on which it is based, and you can test their validity. Catch phrase: use only as directed. Don't limit yourself to a single model: More than one model may be useful for understanding different aspects of the same phenomenon. Catch phrase: legalize polygamy."*
>
> Saul Golomb, "Mathematical Models—Uses and Limitations," 1970 [Gol70].

In this chapter we present a collection of examples spanning many different fields of science and engineering. These examples are used throughout the text and in exercises to illustrate different concepts. First-time readers may wish to focus on only a few examples with which they have had the most prior experience or insight to understand the concepts of state, input, output and dynamics in a familiar setting.

4.1 Cruise Control

The cruise control system of a car is a common feedback system encountered in everyday life. The system attempts to maintain a constant velocity in the presence of disturbances primarily caused by changes in the slope of a road. The controller compensates for these unknowns by measuring the speed of the car and adjusting the throttle appropriately.

To model the system we start with the block diagram in Figure 4.1. Let v be the speed of the car and v_r the desired (reference) speed. The controller, which typically is of the proportional-integral (PI) type described briefly in Chapter 1, receives the signals v and v_r and generates a control signal u that is sent to an actuator that controls the throttle position. The throttle in turn controls the torque T delivered by the engine, which is transmitted through the gears and the wheels, generating a force F that moves the car. There are disturbance forces F_d due to variations in the slope of the road, the rolling resistance and aerodynamic forces. The cruise controller also has a human–machine interface that allows the driver to set and modify the desired speed. There are also functions that disconnect the cruise control when the brake is touched.

The system has many individual components—actuator, engine, transmission, wheels and car body—and a detailed model can be very complicated. In spite of this, the model required to design the cruise controller can be quite simple.

To develop a mathematical model we start with a force balance for the car body. Let v be the speed of the car, m the total mass (including passengers), F the force generated by the contact of the wheels with the road, and F_d the disturbance force

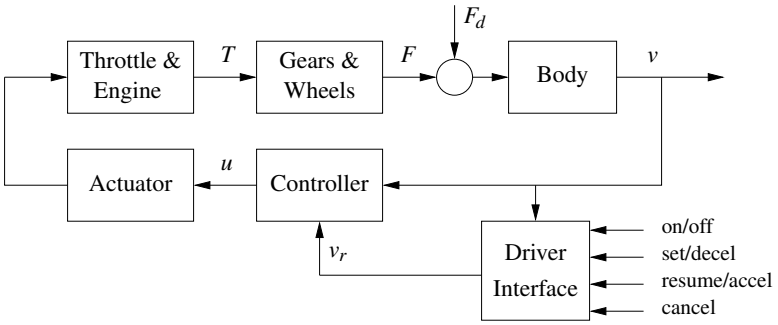


Figure 4.1: Block diagram of a cruise control system for an automobile. The throttle-controlled engine generates a torque T that is transmitted to the ground through the gearbox and wheels. Combined with the external forces from the environment, such as aerodynamic drag and gravitational forces on hills, the net force causes the car to move. The velocity of the car v is measured by a control system that adjusts the throttle through an actuation mechanism. A driver interface allows the system to be turned on and off and the reference speed v_r to be established.

due to gravity, friction and aerodynamic drag. The equation of motion of the car is simply

$$m\frac{dv}{dt} = F - F_d. \tag{4.1}$$

The force F is generated by the engine, whose torque is proportional to the rate of fuel injection, which is itself proportional to a control signal $0 \leq u \leq 1$ that controls the throttle position. The torque also depends on engine speed ω. A simple representation of the torque at full throttle is given by the torque curve

$$T(\omega) = T_m \left(1 - \beta\left(\frac{\omega}{\omega_m} - 1\right)^2\right), \tag{4.2}$$

where the maximum torque T_m is obtained at engine speed ω_m. Typical parameters are $T_m = 190$ Nm, $\omega_m = 420$ rad/s (about 4000 RPM) and $\beta = 0.4$. Let n be the gear ratio and r the wheel radius. The engine speed is related to the velocity through the expression

$$\omega = \frac{n}{r}v =: \alpha_n v,$$

and the driving force can be written as

$$F = \frac{nu}{r}T(\omega) = \alpha_n u T(\alpha_n v).$$

Typical values of α_n for gears 1 through 5 are $\alpha_1 = 40$, $\alpha_2 = 25$, $\alpha_3 = 16$, $\alpha_4 = 12$ and $\alpha_5 = 10$. The inverse of α_n has a physical interpretation as the *effective wheel radius*. Figure 4.2 shows the torque as a function of engine speed and vehicle speed. The figure shows that the effect of the gear is to "flatten" the torque curve so that an almost full torque can be obtained almost over the whole speed range.

The disturbance force F_d has three major components: F_g, the forces due to

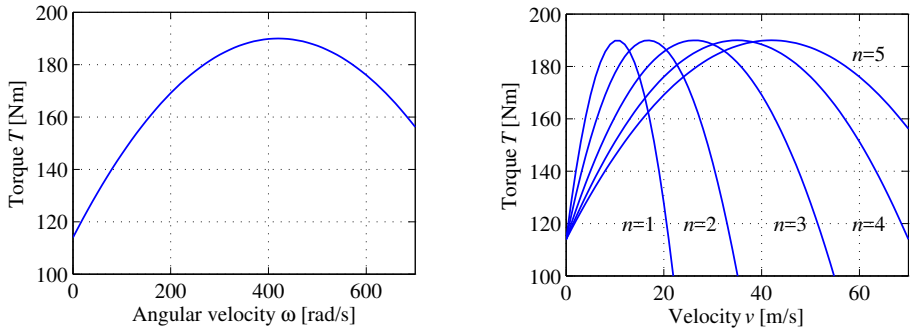


Figure 4.2: Torque curves for typical car engine. The graph on the left shows the torque generated by the engine as a function of the angular velocity of the engine, while the curve on the right shows torque as a function of car speed for different gears.

gravity; F_r, the forces due to rolling friction; and F_a, the aerodynamic drag. Letting the slope of the road be θ, gravity gives the force $F_g = mg\sin\theta$, as illustrated in Figure 4.3a, where $g = 9.8$ m/s^2 is the gravitational constant. A simple model of rolling friction is

$$F_r = mgC_r\,\mathrm{sgn}(v),$$

where C_r is the coefficient of rolling friction and $\mathrm{sgn}(v)$ is the sign of v (± 1) or zero if $v = 0$. A typical value for the coefficient of rolling friction is $C_r = 0.01$. Finally, the aerodynamic drag is proportional to the square of the speed:

$$F_a = \frac{1}{2}\rho C_d A|v|v,$$

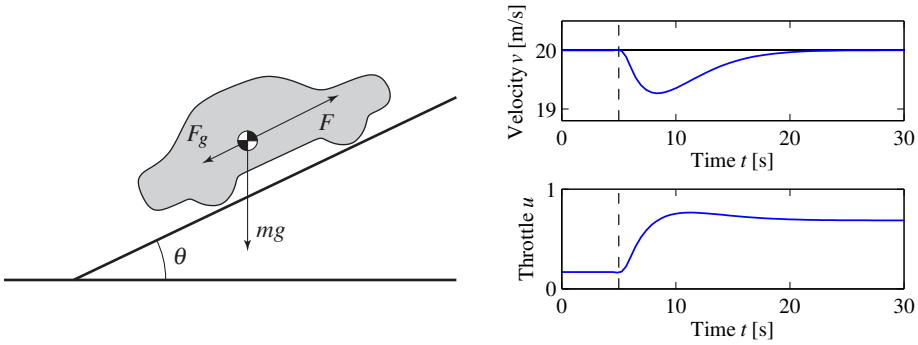
where ρ is the density of air, C_d is the shape-dependent aerodynamic drag coefficient and A is the frontal area of the car. Typical parameters are $\rho = 1.3$ kg/m^3, $C_d = 0.32$ and $A = 2.4$ m^2.

Summarizing, we find that the car can be modeled by

$$m\frac{dv}{dt} = \alpha_n u T(\alpha_n v) - mgC_r\,\mathrm{sgn}(v) - \frac{1}{2}\rho C_d A v^2 - mg\sin\theta, \tag{4.3}$$

where the function T is given by equation (4.2). The model (4.3) is a dynamical system of first order. The state is the car velocity v, which is also the output. The input is the signal u that controls the throttle position, and the disturbance is the force F_d, which depends on the slope of the road. The system is nonlinear because of the torque curve, the gravity term and the nonlinear character of rolling friction and aerodynamic drag. There can also be variations in the parameters; e.g., the mass of the car depends on the number of passengers and the load being carried in the car.

We add to this model a feedback controller that attempts to regulate the speed of the car in the presence of disturbances. We shall use a proportional-integral



(a) Effect of gravitational forces (b) Closed loop response

Figure 4.3: Car with cruise control encountering a sloping road. A schematic diagram is shown in (a), and (b) shows the response in speed and throttle when a slope of 4° is encountered. The hill is modeled as a net change of 4° in hill angle θ, with a linear change in the angle between $t = 5$ and $t = 6$. The PI controller has proportional gain is $k_p = 0.5$, and the integral gain is $k_i = 0.1$.

controller, which has the form

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau.$$

This controller can itself be realized as an input/output dynamical system by defining a controller state z and implementing the differential equation

$$\frac{dz}{dt} = v_r - v, \qquad u = k_p(v_r - v) + k_i z, \tag{4.4}$$

where v_r is the desired (reference) speed. As discussed briefly in Section 1.6, the integrator (represented by the state z) ensures that in steady state the error will be driven to zero, even when there are disturbances or modeling errors. (The design of PI controllers is the subject of Chapter 11.) Figure 4.3b shows the response of the closed loop system, consisting of equations (4.3) and (4.4), when it encounters a hill. The figure shows that even if the hill is so steep that the throttle changes from 0.17 to almost full throttle, the largest speed error is less than 1 m/s, and the desired velocity is recovered after 20 s.

Many approximations were made when deriving the model (4.3). It may seem surprising that such a seemingly complicated system can be described by the simple model (4.3). It is important to make sure that we restrict our use of the model to the uncertainty lemon conceptualized in Figure 3.5b. The model is not valid for very rapid changes of the throttle because we have ignored the details of the engine dynamics, neither is it valid for very slow changes because the properties of the engine will change over the years. Nevertheless the model is very useful for the design of a cruise control system. As we shall see in later chapters, the reason for this is the inherent robustness of feedback systems: even if the model is not perfectly accurate, we can use it to design a controller and make use of the feedback

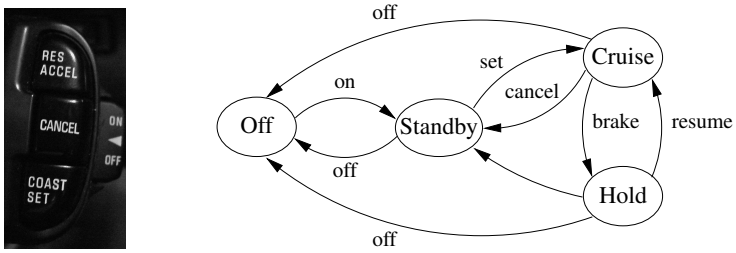


Figure 4.4: Finite state machine for cruise control system. The figure on the left shows some typical buttons used to control the system. The controller can be in one of four modes, corresponding to the nodes in the diagram on the right. Transition between the modes is controlled by pressing one of the five buttons on the cruise control interface: on, off, set, resume or cancel.

in the controller to manage the uncertainty in the system.

The cruise control system also has a human–machine interface that allows the driver to communicate with the system. There are many different ways to implement this system; one version is illustrated in Figure 4.4. The system has four buttons: on-off, set/decelerate, resume/accelerate and cancel. The operation of the system is governed by a finite state machine that controls the modes of the PI controller and the reference generator. Implementation of controllers and reference generators will be discussed more fully in Chapter 11.

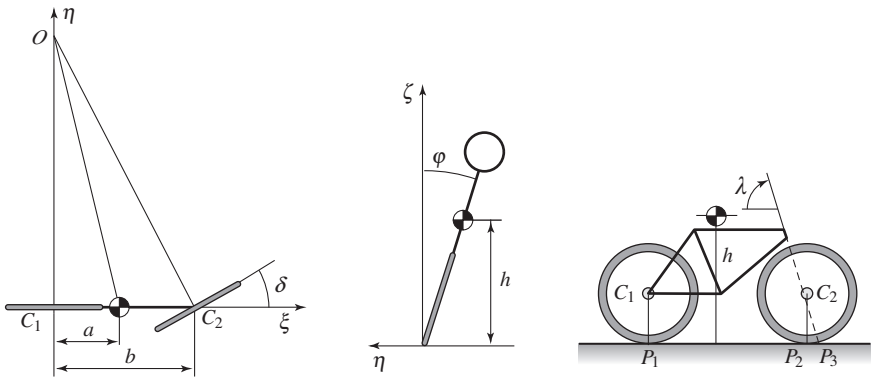
The use of control in automotive systems goes well beyond the simple cruise control system described here. Applications include emissions control, traction control, power control (especially in hybrid vehicles) and adaptive cruise control. Many automotive applications are discussed in detail in the book by Kiencke and Nielsen [KN00] and in the survey papers by Powers et al. [BP96, PN00].

4.2 Bicycle Dynamics

The bicycle is an interesting dynamical system with the feature that one of its key properties is due to a feedback mechanism that is created by the design of the front fork. A detailed model of a bicycle is complex because the system has many degrees of freedom and the geometry is complicated. However, a great deal of insight can be obtained from simple models.

To derive the equations of motion we assume that the bicycle rolls on the horizontal xy plane. Introduce a coordinate system that is fixed to the bicycle with the ξ-axis through the contact points of the wheels with the ground, the η-axis horizontal and the ζ-axis vertical, as shown in Figure 4.5. Let v_0 be the velocity of the bicycle at the rear wheel, b the wheel base, φ the tilt angle and δ the steering angle. The coordinate system rotates around the point O with the angular velocity $\omega = v_0\delta/b$, and an observer fixed to the bicycle experiences forces due to the motion of the coordinate system.

The tilting motion of the bicycle is similar to an inverted pendulum, as shown



(a) top view (b) rear view (c) side view

Figure 4.5: Schematic views of a bicycle. The steering angle is δ, and the roll angle is φ. The center of mass has height h and distance a from a vertical through the contact point P_1 of the rear wheel. The wheel base b is the distance between P_1 and P_2, and the trail c is the distance between P_2 and P_3.

in the rear view in Figure 4.5b. To model the tilt, consider the rigid body obtained when the wheels, the rider and the front fork assembly are fixed to the bicycle frame. Let m be the total mass of the system, J the moment of inertia of this body with respect to the ξ-axis and D the product of inertia with respect to the $\xi\zeta$ axes. Furthermore, let the ξ and ζ coordinates of the center of mass with respect to the rear wheel contact point, P_1, be a and h, respectively. We have $J \approx mh^2$ and $D = mah$. The torques acting on the system are due to gravity and centripetal action. Assuming that the steering angle δ is small, the equation of motion becomes

$$J\frac{d^2\varphi}{dt^2} - \frac{Dv_0}{b}\frac{d\delta}{dt} = mgh\sin\varphi + \frac{mv_0^2h}{b}\delta. \tag{4.5}$$

The term $mgh\sin\varphi$ is the torque generated by gravity. The terms containing δ and its derivative are the torques generated by steering, with the term $(Dv_0/b)\,d\delta/dt$ due to inertial forces and the term $(mv_0^2h/b)\,\delta$ due to centripetal forces.

The steering angle is influenced by the torque the rider applies to the handle bar. Because of the tilt of the steering axis and the shape of the front fork, the contact point of the front wheel with the road P_2 is behind the axis of rotation of the front wheel assembly, as shown in Figure 4.5c. The distance c between the contact point of the front wheel P_2 and the projection of the axis of rotation of the front fork assembly P_3 is called the *trail*. The steering properties of a bicycle depend critically on the trail. A large trail increases stability but makes the steering less agile.

A consequence of the design of the front fork is that the steering angle δ is influenced both by steering torque T and by the tilt of the frame φ. This means that a bicycle with a front fork is a *feedback system* as illustrated by the block diagram in Figure 4.6. The steering angle δ influences the tilt angle φ, and the tilt angle influences the steering angle, giving rise to the circular causality that is

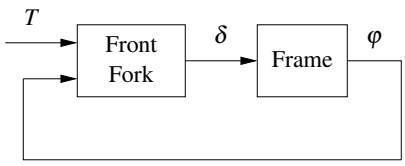


Figure 4.6: Block diagram of a bicycle with a front fork. The steering torque applied to the handlebars is T, the roll angle is φ and the steering angle is δ. Notice that the front fork creates a feedback from the roll angle φ to the steering angle δ that under certain conditions can stabilize the system.

characteristic of reasoning about feedback. For a front fork with a positive trail, the bicycle will steer into the lean, creating a centrifugal force that attempts to diminish the lean. Under certain conditions, the feedback can actually stabilize the bicycle. A crude empirical model is obtained by assuming that the block B can be modeled as the static system

$$\delta = k_1 T - k_2 \varphi. \tag{4.6}$$

This model neglects the dynamics of the front fork, the tire–road interaction and the fact that the parameters depend on the velocity. A more accurate model, called the *Whipple model*, is obtained using the rigid-body dynamics of the front fork and the frame. Assuming small angles, this model becomes

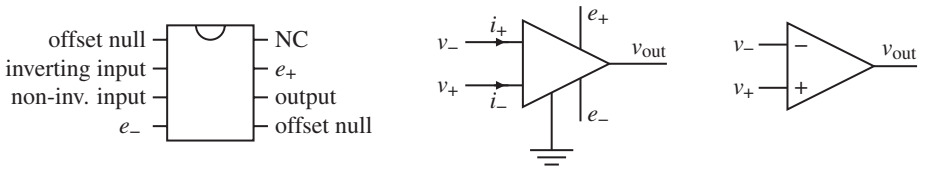
$$M \begin{pmatrix} \ddot{\varphi} \\ \ddot{\delta} \end{pmatrix} + C v_0 \begin{pmatrix} \dot{\varphi} \\ \dot{\delta} \end{pmatrix} + (K_0 + K_2 v_0^2) \begin{pmatrix} \varphi \\ \delta \end{pmatrix} = \begin{pmatrix} 0 \\ T \end{pmatrix}, \tag{4.7}$$

where the elements of the 2×2 matrices M, C, K_0 and K_2 depend on the geometry and the mass distribution of the bicycle. Note that this has a form somewhat similar to that of the spring–mass system introduced in Chapter 3 and the balance system in Example 3.1. Even this more complex model is inaccurate because the interaction between the tire and the road is neglected; taking this into account requires two additional state variables. Again, the uncertainty lemon in Figure 3.5b provides a framework for understanding the validity of the model under these assumptions.

Interesting presentations on the development of the bicycle are given in the books by D. Wilson [Wil04] and Herlihy [Her04]. The model (4.7) was presented in a paper by Whipple in 1899 [Whi99]. More details on bicycle modeling are given in the paper [ÅKL05], which has many references.

4.3 Operational Amplifier Circuits

An operational amplifier (op amp) is a modern implementation of Black's feedback amplifier. It is a universal component that is widely used for instrumentation, control and communication. It is also a key element in analog computing. Schematic diagrams of the operational amplifier are shown in Figure 4.7. The amplifier has one inverting input (v_-), one noninverting input (v_+) and one output (v_{out}). There are also connections for the supply voltages, e_- and e_+, and a zero adjustment



(a) Chip pinout (b) Full schematic (c) Simple view

Figure 4.7: An operational amplifier and two schematic diagrams. (a) The amplifier pin connections on an integrated circuit chip. (b) A schematic with all connections. (c) Only the signal connections.

(offset null). A simple model is obtained by assuming that the input currents i_- and i_+ are zero and that the output is given by the static relation

$$v_{\text{out}} = \text{sat}_{(v_{\min}, v_{\max})}\big(k(v_+ - v_-)\big), \tag{4.8}$$

where sat denotes the saturation function

$$\text{sat}_{(a,b)}(x) = \begin{cases} a & \text{if } x < a \\ x & \text{if } a \le x \le b \\ b & \text{if } x > b. \end{cases} \tag{4.9}$$

We assume that the gain k is large, in the range of 10^6–10^8, and the voltages $v_{\min}$ and $v_{\max}$ satisfy

$$e_- \le v_{\min} < v_{\max} \le e_+$$

and hence are in the range of the supply voltages. More accurate models are obtained by replacing the saturation function with a smooth function as shown in Figure 4.8. For small input signals the amplifier characteristic (4.8) is linear:

$$v_{\text{out}} = k(v_+ - v_-) =: -kv. \tag{4.10}$$

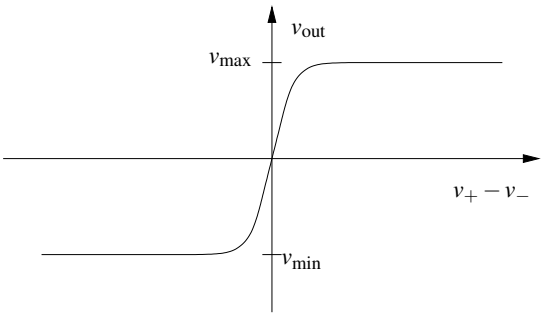
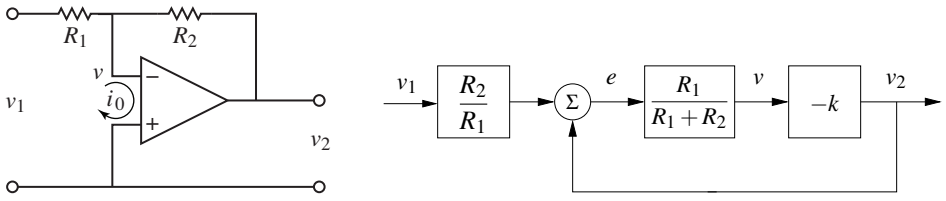


Figure 4.8: Input/output characteristics of an operational amplifier. The differential input is given by $v_+ - v_-$. The output voltage is a linear function of the input in a small range around 0, with saturation at $v_{\min}$ and $v_{\max}$. In the linear regime the op amp has high gain.



(a) Amplifier circuit (b) Block diagram

Figure 4.9: Stable amplifier using an op amp. The circuit (a) uses negative feedback around an operational amplifier and has a corresponding block diagram (b). The resistors R_1 and R_2 determine the gain of the amplifier.

Since the open loop gain k is very large, the range of input signals where the system is linear is very small.

A simple amplifier is obtained by arranging feedback around the basic operational amplifier as shown in Figure 4.9a. To model the feedback amplifier in the linear range, we assume that the current $i_0 = i_- + i_+$ is zero and that the gain of the amplifier is so large that the voltage $v = v_- - v_+$ is also zero. It follows from Ohm's law that the currents through resistors R_1 and R_2 are given by

$$\frac{v_1}{R_1} = -\frac{v_2}{R_2},$$

and hence the closed loop gain of the amplifier is

$$\frac{v_2}{v_1} = -k_{\text{cl}}, \qquad \text{where} \quad k_{\text{cl}} = \frac{R_2}{R_1}. \tag{4.11}$$

A more accurate model is obtained by continuing to neglect the current i_0 but assuming that the voltage v is small but not negligible. The current balance is then

$$\frac{v_1 - v}{R_1} = \frac{v - v_2}{R_2}. \tag{4.12}$$

Assuming that the amplifier operates in the linear range and using equation (4.10), the gain of the closed loop system becomes

$$k_{\text{cl}} = -\frac{v_2}{v_1} = \frac{R_2}{R_1}\,\frac{kR_1}{R_1 + R_2 + kR_1} \tag{4.13}$$

If the open loop gain k of the operational amplifier is large, the closed loop gain k_{cl} is the same as in the simple model given by equation (4.11). Notice that the closed loop gain depends only on the passive components and that variations in k have only a marginal effect on the closed loop gain. For example if $k = 10^6$ and $R_2/R_1 = 100$, a variation of k by 100% gives only a variation of 0.01% in the closed loop gain. The drastic reduction in sensitivity is a nice illustration of how feedback can be used to make precise systems from uncertain components. In this particular case, feedback is used to trade high gain and low robustness for low gain and high robustness. Equation (4.13) was the formula that inspired Black when he invented the feedback amplifier [Bla34] (see the quote at the beginning of Chapter 13).

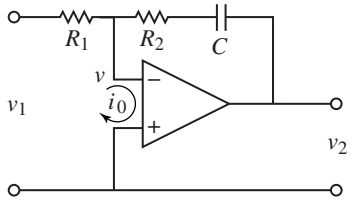


Figure 4.10: Circuit diagram of a PI controller obtained by feedback around an operational amplifier. The capacitor C is used to store charge and represents the integral of the input.

It is instructive to develop a block diagram for the feedback amplifier in Figure 4.9a. To do this we will represent the pure amplifier with input v and output v_2 as one block. To complete the block diagram, we must describe how v depends on v_1 and v_2. Solving equation (4.12) for v gives

$$v = \frac{R_2}{R_1+R_2}v_1 + \frac{R_1}{R_1+R_2}v_2 = \frac{R_1}{R_1+R_2}\Big(\frac{R_2}{R_1}v_1 + v_2\Big),$$

and we obtain the block diagram shown in Figure 4.9b. The diagram clearly shows that the system has feedback and that the gain from v_2 to v is $R_1/(R_1+R_2)$, which can also be read from the circuit diagram in Figure 4.9a. If the loop is stable and the gain of the amplifier is large, it follows that the error e is small, and we find that $v_2 = -(R_2/R_1)v_1$. Notice that the resistor R_1 appears in two blocks in the block diagram. This situation is typical in electrical circuits, and it is one reason why block diagrams are not always well suited for some types of physical modeling.

The simple model of the amplifier given by equation (4.10) provides qualitative insight, but it neglects the fact that the amplifier is a dynamical system. A more realistic model is

$$\frac{dv_{\text{out}}}{dt} = -av_{\text{out}} - bv. \tag{4.14}$$

The parameter b that has dimensions of frequency and is called the *gain-bandwidth product* of the amplifier. Whether a more complicated model is used depends on the questions to be answered and the required size of the uncertainty lemon. The model (4.14) is still not valid for very high or very low frequencies since drift causes deviations at low frequencies and there are additional dynamics that appear at frequencies close to b. The model is also not valid for large signals—an upper limit is given by the voltage of the power supply, typically in the range of 5–10 V—neither is it valid for very low signals because of electrical noise. These effects can be added, if needed, but increase the complexity of the analysis.

The operational amplifier is very versatile, and many different systems can be built by combining it with resistors and capacitors. In fact, any linear system can be implemented by combining operational amplifiers with resistors and capacitors. Exercise 4.5 shows how a second-order oscillator is implemented, and Figure 4.10 shows the circuit diagram for an analog proportional-integral controller. To develop a simple model for the circuit we assume that the current i_0 is zero and that the open loop gain k is so large that the input voltage v is negligible. The current i

through the capacitor is $i = C dv_c/dt$, where v_c is the voltage across the capacitor. Since the same current goes through the resistor R_1, we get

$$i = \frac{v_1}{R_1} = C\frac{dv_c}{dt},$$

which implies that

$$v_c(t) = \frac{1}{C}\int i(t)\,dt = \frac{1}{R_1 C}\int_0^t v_1(\tau)d\tau.$$

The output voltage is thus given by

$$v_2(t) = -R_2 i - v_c = -\frac{R_2}{R_1}v_1(t) - \frac{1}{R_1 C}\int_0^t v_1(\tau)d\tau,$$

which is the input/output relation for a PI controller.

The development of operational amplifiers was pioneered by Philbrick [Lun05, Phi48], and their usage is described in many textbooks (e.g., [CD75]). Good information is also available from suppliers [Jun02, Man02].

4.4 Computing Systems and Networks

The application of feedback to computing systems follows the same principles as the control of physical systems, but the types of measurements and control inputs that can be used are somewhat different. Measurements (sensors) are typically related to resource utilization in the computing system or network and can include quantities such as the processor load, memory usage or network bandwidth. Control variables (actuators) typically involve setting limits on the resources available to a process. This might be done by controlling the amount of memory, disk space or time that a process can consume, turning on or off processing, delaying availability of a resource or rejecting incoming requests to a server process. Process modeling for networked computing systems is also challenging, and empirical models based on measurements are often used when a first-principles model is not available.

Web Server Control

Web servers respond to requests from the Internet and provide information in the form of web pages. Modern web servers start multiple processes to respond to requests, with each process assigned to a single source until no further requests are received from that source for a predefined period of time. Processes that are idle become part of a pool that can be used to respond to new requests. To provide a fast response to web requests, it is important that the web server processes do not overload the server's computational capabilities or exhaust its memory. Since other processes may be running on the server, the amount of available processing power and memory is uncertain, and feedback can be used to provide good performance in the presence of this uncertainty.

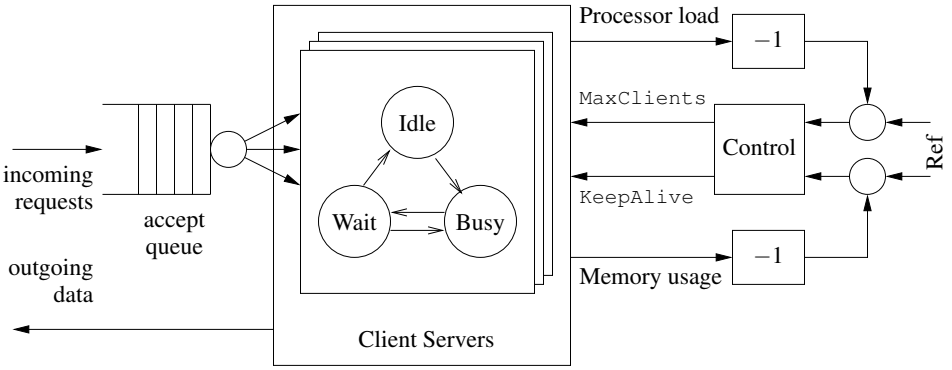


Figure 4.11: Feedback control of a web server. Connection requests arrive on an input queue, where they are sent to a server process. A finite state machine keeps track of the state of the individual server processes and responds to requests. A control algorithm can modify the server's operation by controlling parameters that affect its behavior, such as the maximum number of requests that can be serviced at a single time (`MaxClients`) or the amount of time that a connection can remain idle before it is dropped (`KeepAlive`).

Figure 4.11 illustrates the use of feedback to modulate the operation of an Apache web server. The web server operates by placing incoming connection requests on a queue and then starting a subprocess to handle requests for each accepted connection. This subprocess responds to requests from a given connection as they come in, alternating between a `Busy` state and a `Wait` state. (Keeping the subprocess active between requests is known as the *persistence* of the connection and provides a substantial reduction in latency to requests for multiple pieces of information from a single site.) If no requests are received for a sufficiently long period of time, controlled by the `KeepAlive` parameter, then the connection is dropped and the subprocess enters an `Idle` state, where it can be assigned another connection. A maximum of `MaxClients` simultaneous requests will be served, with the remainder remaining on the incoming request queue.

The parameters that control the server represent a trade-off between performance (how quickly requests receive a response) and resource usage (the amount of processing power and memory used by the server). Increasing the `MaxClients` parameter allows connection requests to be pulled off of the queue more quickly but increases the amount of processing power and memory usage that is required. Increasing the `KeepAlive` timeout means that individual connections can remain idle for a longer period of time, which decreases the processing load on the machine but increases the length of the queue (and hence the amount of time required for a user to initiate a connection). Successful operation of a busy server requires a proper choice of these parameters, often based on trial and error.

To model the dynamics of this system in more detail, we create a discrete-time model with states given by the average processor load x_{cpu} and the percentage memory usage x_{mem}. The inputs to the system are taken as the maximum number of clients u_{mc} and the keep-alive time u_{ka}. If we assume a linear model around the

equilibrium point, the dynamics can be written as

$$\begin{pmatrix} x_{\text{cpu}}[k+1] \\ x_{\text{mem}}[k+1] \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_{\text{cpu}}[k] \\ x_{\text{mem}}[k] \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \begin{pmatrix} u_{\text{ka}}[k] \\ u_{\text{mc}}[k] \end{pmatrix}, \quad (4.15)$$

where the coefficients of the A and B matrices can be determined based on empirical measurements or detailed modeling of the web server's processing and memory usage. Using system identification, Diao et al. [DGH+02, HDPT04] identified the linearized dynamics as

$$A = \begin{pmatrix} 0.54 & -0.11 \\ -0.026 & 0.63 \end{pmatrix}, \qquad B = \begin{pmatrix} -85 & 4.4 \\ -2.5 & 2.8 \end{pmatrix} \times 10^{-4},$$

where the system was linearized about the equilibrium point

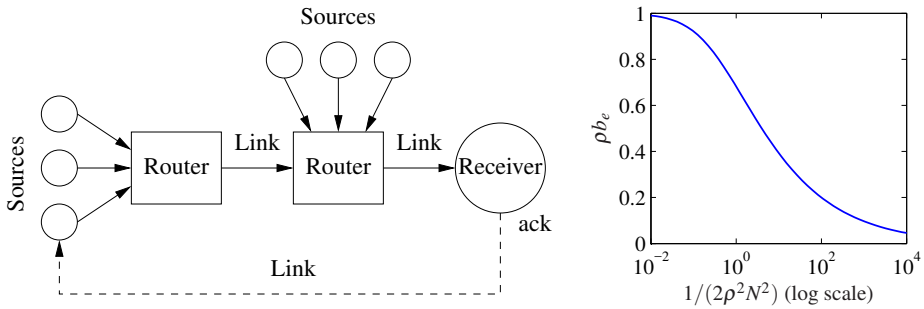
$$x_{\text{cpu}} = 0.58, \qquad u_{\text{ka}} = 11 \text{ s}, \qquad x_{\text{mem}} = 0.55, \qquad u_{\text{mc}} = 600.$$

This model shows the basic characteristics that were described above. Looking first at the B matrix, we see that increasing the `KeepAlive` timeout (first column of the B matrix) decreases both the processor usage and the memory usage since there is more persistence in connections and hence the server spends a longer time waiting for a connection to close rather than taking on a new active connection. The `MaxClients` connection increases both the processing and memory requirements. Note that the largest effect on the processor load is the `KeepAlive` timeout. The A matrix tells us how the processor and memory usage evolve in a region of the state space near the equilibrium point. The diagonal terms describe how the individual resources return to equilibrium after a transient increase or decrease. The off-diagonal terms show that there is coupling between the two resources, so that a change in one could cause a later change in the other.

Although this model is very simple, we will see in later examples that it can be used to modify the parameters controlling the server in real time and provide robustness with respect to uncertainties in the load on the machine. Similar types of mechanisms have been used for other types of servers. It is important to remember the assumptions on the model and their role in determining when the model is valid. In particular, since we have chosen to use average quantities over a given sample time, the model will not provide an accurate representation for high-frequency phenomena.

Congestion Control

The Internet was created to obtain a large, highly decentralized, efficient and expandable communication system. The system consists of a large number of interconnected gateways. A message is split into several packets which are transmitted over different paths in the network, and the packages are rejoined to recover the message at the receiver. An acknowledgment ("ack") message is sent back to the sender when a packet is received. The operation of the system is governed by a simple but powerful decentralized control structure that has evolved over time.



(a) Block diagram (b) Operating point

Figure 4.12: Internet congestion control. (a) Source computers send information to routers, which forward the information to other routers that eventually connect to the receiving computer. When a packet is received, an acknowledgment packet is sent back through the routers (not shown). The routers buffer information received from the sources and send the data across the outgoing link. (b) The equilibrium buffer size b_e for a set of N identical computers sending packets through a single router with drop probability ρ.

The system has two control mechanisms called *protocols*: the Transmission Control Protocol (TCP) for end-to-end network communication and the Internet Protocol (IP) for routing packets and for host-to-gateway or gateway-to-gateway communication. The current protocols evolved after some spectacular congestion collapses occurred in the mid 1980s, when throughput unexpectedly could drop by a factor of 1000 [Jac95]. The control mechanism in TCP is based on conserving the number of packets in the loop from the sender to the receiver and back to the sender. The sending rate is increased exponentially when there is no congestion, and it is dropped to a low level when there is congestion.

To derive an overall model for congestion control, we model three separate elements of the system: the rate at which packets are sent by individual sources (computers), the dynamics of the queues in the links (routers) and the admission control mechanism for the queues. Figure 4.12a is a block diagram of the system.

The current source control mechanism on the Internet is a protocol known as TCP/Reno [LPD02]. This protocol operates by sending packets to a receiver and waiting to receive an acknowledgment from the receiver that the packet has arrived. If no acknowledgment is sent within a certain timeout period, the packet is retransmitted. To avoid waiting for the acknowledgment before sending the next packet, Reno transmits multiple packets up to a fixed *window* around the latest packet that has been acknowledged. If the window length is chosen properly, packets at the beginning of the window will be acknowledged before the source transmits packets at the end of the window, allowing the computer to continuously stream packets at a high rate.

To determine the size of the window to use, TCP/Reno uses a feedback mechanism in which (roughly speaking) the window size is increased at a fixed rate as long as packets are acknowledged and the window size is cut in half when packets are lost. This mechanism allows a dynamic adjustment of the window size in which

each computer acts in a greedy fashion as long as packets are being delivered but backs off quickly when congestion occurs.

A model for the behavior of the source can be developed by describing the dynamics of the window size. Suppose we have N computers and let w_i be the current window size (measured in number of packets) for the ith computer. Let q_i represent the end-to-end probability that a packet will be dropped someplace between the source and the receiver. We can model the dynamics of the window size by the differential equation

$$\frac{dw_i}{dt} = (1-q_i)\frac{r_i(t-\tau_i)}{w_i} + q_i(-\frac{w_i}{2}r_i(t-\tau_i)), \qquad r_i = \frac{w_i}{\tau_i}, \tag{4.16}$$

where τ_i is the end-to-end transmission time for a packet to reach is destination and the acknowledgment to be sent back and r_i is the resulting rate at which packets are cleared from the list of packets that have been received. The first term in the dynamics represents the increase in window size when a packet is received, and the second term represents the decrease in window size when a packet is lost. Notice that r_i is evaluated at time $t-\tau_i$, representing the time required to receive additional acknowledgments.

The link dynamics are controlled by the dynamics of the router queue and the admission control mechanism for the queue. Assume that we have L links in the network and use l to index the individual links. We model the queue in terms of the current number of packets in the router's buffer b_l and assume that the router can contain a maximum of $b_{l,\max}$ packets and transmits packets at a rate c_l, equal to the capacity of the link. The buffer dynamics can then be written as

$$\frac{db_l}{dt} = s_l - c_l, \qquad s_l = \sum_{\{i:\ l\in L_i\}} r_i(t-\tau_{li}^f), \tag{4.17}$$

where L_i is the set of links that are being used by source i, τ_{li}^f is the time it takes a packet from source i to reach link l and s_l is the total rate at which packets arrive at link l.

The admission control mechanism determines whether a given packet is accepted by a router. Since our model is based on the average quantities in the network and not the individual packets, one simple model is to assume that the probability that a packet is dropped depends on how full the buffer is: $p_l = m_l(b_l, b_{\max})$. For simplicity, we will assume for now that $p_l = \rho_l b_l$ (see Exercise 4.6 for a more detailed model). The probability that a packet is dropped at a given link can be used to determine the end-to-end probability that a packet is lost in transmission:

$$q_i = 1 - \prod_{l\in L_i}(1-p_l) \approx \sum_{l\in L_i} p_l(t-\tau_{li}^b), \tag{4.18}$$

where τ_{li}^b is the backward delay from link l to source i and the approximation is valid as long as the individual drop probabilities are small. We use the backward delay since this represents the time required for the acknowledgment packet to be received by the source.

Together, equations (4.16), (4.17) and (4.18) represent a model of congestion control dynamics. We can obtain substantial insight by considering a special case in which we have N identical sources and 1 link. In addition, we assume for the moment that the forward and backward time delays can be ignored, in which case the dynamics can be reduced to the form

$$\frac{dw_i}{dt} = \frac{1}{\tau} - \frac{\rho c(2+w_i^2)}{2}, \qquad \frac{db}{dt} = \sum_{i=1}^{N} \frac{w_i}{\tau} - c, \qquad \tau = \frac{b}{c}, \tag{4.19}$$

where $w_i \in \mathbb{R}$, $i = 1,\ldots,N$, are the window sizes for the sources of data, $b \in \mathbb{R}$ is the current buffer size of the router, ρ controls the rate at which packets are dropped and c is the capacity of the link connecting the router to the computers. The variable τ represents the amount of time required for a packet to be processed by a router, based on the size of the buffer and the capacity of the link. Substituting τ into the equations, we write the state space dynamics as

$$\frac{dw_i}{dt} = \frac{c}{b} - \rho c\left(1 + \frac{w_i^2}{2}\right), \qquad \frac{db}{dt} = \sum_{i=1}^{N} \frac{cw_i}{b} - c. \tag{4.20}$$

More sophisticated models can be found in [HMTG00, LPD02].

The nominal operating point for the system can be found by setting $\dot{w}_i = \dot{b} = 0$:

$$0 = \frac{c}{b} - \rho c\left(1 + \frac{w_i^2}{2}\right), \qquad 0 = \sum_{i=1}^{N} \frac{cw_i}{b} - c.$$

Exploiting the fact that all of the source dynamics are identical, it follows that all of the w_i should be the same, and it can be shown that there is a unique equilibrium satisfying the equations

$$w_{i,e} = \frac{b_e}{N} = \frac{c\tau_e}{N}, \qquad \frac{1}{2\rho^2 N^2}(\rho b_e)^3 + (\rho b_e) - 1 = 0. \tag{4.21}$$

The solution for the second equation is a bit messy but can easily be determined numerically. A plot of its solution as a function of $1/(2\rho^2N^2)$ is shown in Figure 4.12b. We also note that at equilibrium we have the following additional equalities:

$$\tau_e = \frac{b_e}{c} = \frac{Nw_e}{c}, \qquad q_e = Np_e = N\rho b_e, \qquad r_e = \frac{w_e}{\tau_e}. \tag{4.22}$$

Figure 4.13 shows a simulation of 60 sources communicating across a single link, with 20 sources dropping out at $t = 500$ ms and the remaining sources increasing their rates (window sizes) to compensate. Note that the buffer size and window sizes automatically adjust to match the capacity of the link.

A comprehensive treatment of computer networks is given in the textbook by Tannenbaum [Tan96]. A good presentation of the ideas behind the control principles for the Internet is given by one of its designers, Van Jacobson, in [Jac95]. F. Kelly [Kel85] presents an early effort on the analysis of the system. The books by

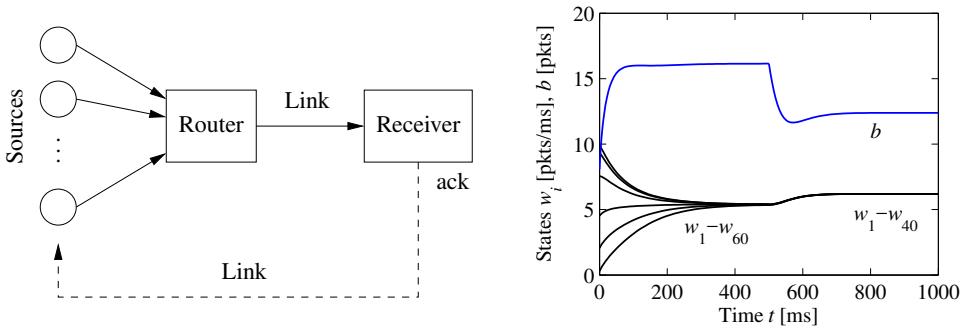


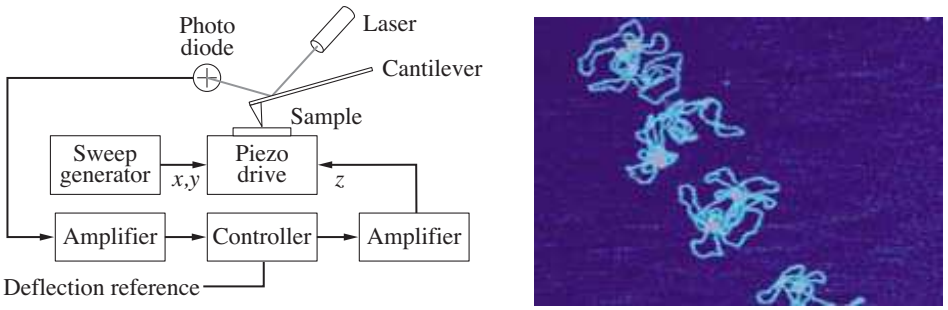
Figure 4.13: Internet congestion control for N identical sources across a single link. As shown on the left, multiple sources attempt to communicate through a router across a single link. An "ack" packet sent by the receiver acknowledges that the message was received; otherwise the message packet is resent and the sending rate is slowed down at the source. The simulation on the right is for 60 sources starting random rates, with 20 sources dropping out at $t = 500$ ms. The buffer size is shown at the top, and the individual source rates for 6 of the sources are shown at the bottom.

Hellerstein et al. [HDPT04] and Janert [Jan14] give many examples of the use of feedback in computer systems.

4.5 Atomic Force Microscopy

The 1986 Nobel Prize in Physics was shared by Gerd Binnig and Heinrich Rohrer for their design of the *scanning tunneling microscope*. The idea of the instrument is to bring an atomically sharp tip so close to a conducting surface that tunneling occurs. An image is obtained by traversing the tip across the sample and measuring the tunneling current as a function of tip position. This invention has stimulated the development of a family of instruments that permit visualization of surface structure at the nanometer scale, including the *atomic force microscope* (AFM), where a sample is probed by a tip on a cantilever. An AFM can operate in two modes. In *tapping mode* the cantilever is vibrated, and the amplitude of vibration is controlled by feedback. In *contact mode* the cantilever is in contact with the sample, and its bending is controlled by feedback. In both cases control is actuated by a piezo element that controls the vertical position of the cantilever base (or the sample). The control system has a direct influence on picture quality and scanning rate.

A schematic picture of an atomic force microscope is shown in Figure 4.14a. A microcantilever with a tip having a radius of the order of 10 nm is placed close to the sample. The tip can be moved vertically and horizontally using a piezoelectric scanner. It is clamped to the sample surface by attractive van der Waals forces and repulsive Pauli forces. The cantilever tilt depends on the topography of the surface and the position of the cantilever base, which is controlled by the piezo element.



(a) Schematic diagram (b) AFM image of DNA

Figure 4.14: Atomic force microscope. (a) A schematic diagram of an atomic force microscope, consisting of a piezo drive that scans the sample under the AFM tip. A laser reflects off of the cantilever and is used to measure the detection of the tip through a feedback controller. (b) An AFM image of strands of DNA. (Image courtesy Veeco Instruments.)

The tilt is measured by sensing the deflection of the laser beam using a photodiode. The signal from the photodiode is amplified and sent to a controller that drives the amplifier for the vertical position of the cantilever. By controlling the piezo element so that the deflection of the cantilever is constant, the signal driving the vertical deflection of the piezo element is a measure of the atomic forces between the cantilever tip and the atoms of the sample. An image of the surface is obtained by scanning the cantilever along the sample. The resolution makes it possible to see the structure of the sample on the atomic scale, as illustrated in Figure 4.14b, which shows an AFM image of DNA.

The horizontal motion of an AFM is typically modeled as a spring–mass system with low damping. The vertical motion is more complicated. To model the system, we start with the block diagram shown in Figure 4.15. Signals that are easily accessible are the input voltage u to the power amplifier that drives the piezo element, the voltage v applied to the piezo element and the output voltage y of the signal amplifier for the photodiode. The controller is a PI controller implemented by a computer, which is connected to the system by analog-to-digital (A/D) and digital-to-analog (D/A) converters. The deflection of the cantilever φ is also shown in the figure. The desired reference value for the deflection is an input to the computer.

There are several different configurations that have different dynamics. Here we will discuss a high-performance system from [SÅD+07] where the cantilever base is positioned vertically using a piezo stack. We begin the modeling with a simple experiment on the system. Figure 4.16a shows a step response of a scanner from the input voltage u to the power amplifier to the output voltage y of the signal amplifier for the photodiode. This experiment captures the dynamics of the chain of blocks from u to y in the block diagram in Figure 4.15. Figure 4.16a shows that the system responds quickly but that there is a poorly damped oscillatory mode with a period of about 35 μs. A primary task of the modeling is to understand the origin of the oscillatory behavior. To do so we will explore the system in more

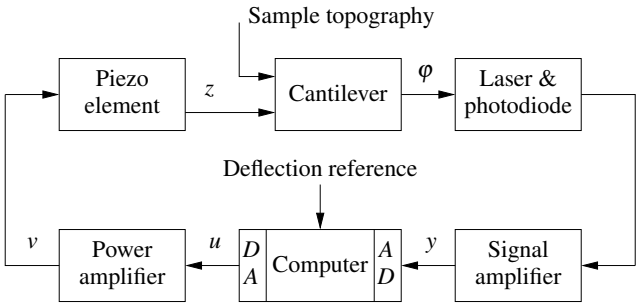


Figure 4.15: Block diagram of the system for vertical positioning of the cantilever for an atomic force microscope in contact mode. The control system attempts to keep the cantilever deflection equal to its reference value. Cantilever deflection is measured, amplified and converted to a digital signal, then compared with its reference value. A correcting signal is generated by the computer, converted to analog form, amplified and sent to the piezo element.

detail.

The natural frequency of the clamped cantilever is typically several hundred kilohertz, which is much higher than the observed oscillation of about 30 kHz. As a first approximation we will model it as a static system. Since the deflections are small, we can assume that the bending φ of the cantilever is proportional to the difference in height between the cantilever tip at the probe and the piezo scanner. A more accurate model can be obtained by modeling the cantilever as a spring–mass system of the type discussed in Chapter 3.

Figure 4.16a also shows that the response of the power amplifier is fast. The photodiode and the signal amplifier also have fast responses and can thus be modeled as static systems. The remaining block is a piezo system with suspension. A schematic mechanical representation of the vertical motion of the scanner is shown in Figure 4.16b. We will model the system as two masses separated by an ideal piezo element. The mass m_1 is half of the piezo system, and the mass m_2 is the other half of the piezo system plus the mass of the support.

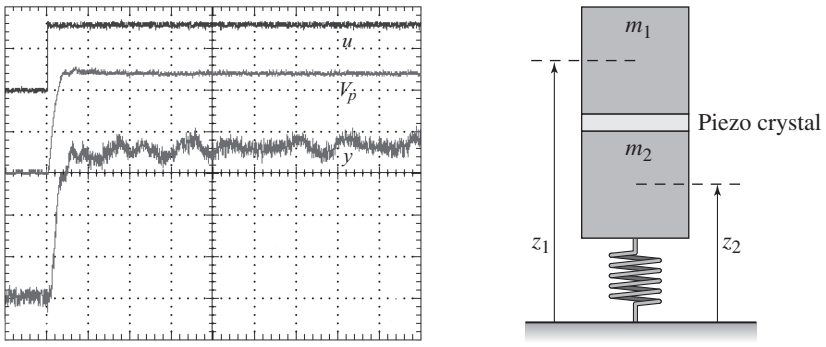
A simple model is obtained by assuming that the piezo crystal generates a force F between the masses and that there is a damping c in the spring. Let the positions of the center of the masses be z_1 and z_2. A momentum balance gives the following model for the system:

$$m_1 \frac{d^2 z_1}{dt^2} = F, \qquad m_2 \frac{d^2 z_2}{dt^2} = -c_2 \frac{dz_2}{dt} - k_2 z_2 - F.$$

Let the elongation of the piezo element $l = z_1 - z_2$ be the control variable and the height z_1 of the cantilever base be the output. Eliminating the variable F in the equations above and substituting $z_1 - l$ for z_2 gives the model

$$(m_1 + m_2)\frac{d^2 z_1}{dt^2} + c_2 \frac{dz_1}{dt} + k_2 z_1 = m_2 \frac{d^2 l}{dt^2} + c_2 \frac{dl}{dt} + k_2 l. \tag{4.23}$$

Summarizing, we find that a simple model of the system is obtained by mod-



(a) Step response (b) Mechanical model

Figure 4.16: Modeling of an atomic force microscope. (a) A measured step response. The top curve shows the voltage u applied to the drive amplifier (50 mV/div), the middle curve is the output V_p of the power amplifier (500 mV/div) and the bottom curve is the output y of the signal amplifier (500 mV/div). The time scale is 25 μs/div. Data have been supplied by Georg Schitter. (b) A simple mechanical model for the vertical positioner and the piezo crystal.

eling the piezo by (4.23) and all the other blocks by static models. Introducing the linear equations $l = k_3 u$ and $y = k_4 z_1$, we now have a complete model relating the output y to the control signal u. A more accurate model can be obtained by introducing the dynamics of the cantilever and the power amplifier. As in the previous examples, the concept of the uncertainty lemon in Figure 3.5b provides a framework for describing the uncertainty: the model will be accurate up to the frequencies of the fastest modeled modes and over a range of motion in which linearized stiffness models can be used.

The experimental results in Figure 4.16a can be explained qualitatively as follows. When a voltage is applied to the piezo, it expands by l_0, the mass m_1 moves up and the mass m_2 moves down instantaneously. The system settles after a poorly damped oscillation.

It is highly desirable to design a control system for the vertical motion so that it responds quickly with little oscillation. The instrument designer has several choices: to accept the oscillation and have a slow response time, to design a control system that can damp the oscillations or to redesign the mechanics to give resonances of higher frequency. The last two alternatives give a faster response and faster imaging.

Since the dynamic behavior of the system changes with the properties of the sample, it is necessary to tune the feedback loop. In simple systems this is currently done manually by adjusting parameters of a PI controller. There are interesting possibilities for making AFM systems easier to use by introducing automatic tuning and adaptation.

The book by Sarid [Sar91] gives a broad coverage of atomic force microscopes. The interaction of atoms close to surfaces is fundamental to solid state physics, see Kittel [Kit95]. The model discussed in this section is based on Schitter [Sch01].

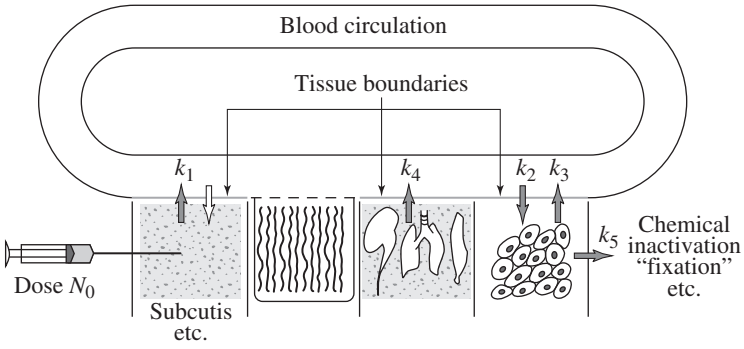


Figure 4.17: Abstraction used to compartmentalize the body for the purpose of describing drug distribution (based on Teorell [Teo37]). The body is abstracted by a number of compartments with perfect mixing, and the complex transport processes are approximated by assuming that the flow is proportional to the concentration differences in the compartments. The constants k_i parameterize the rates of flow between different compartments.

4.6 Drug Administration

The phrase "Take two pills three times a day" is a recommendation with which we are all familiar. Behind this recommendation is a solution of an open loop control problem. The key issue is to make sure that the concentration of a medicine in a part of the body is sufficiently high to be effective but not so high that it will cause undesirable side effects. The control action is quantized, *take two pills*, and sampled, *every 8 hours*. The prescriptions are based on simple models captured in empirical tables, and the dose is based on the age and weight of the patient.

Drug administration is a control problem. To solve it we must understand how a drug spreads in the body after it is administered. This topic, called *pharmacokinetics*, is now a discipline of its own, and the models used are called *compartment models*. They go back to the 1920s when Widmark modeled the propagation of alcohol in the body [WT24]. Compartment models are now important for the screening of all drugs used by humans. The schematic diagram in Figure 4.17 illustrates the idea of a compartment model. The body is viewed as a number of compartments like blood plasma, kidney, liver and tissues that are separated by membranes. It is assumed that there is perfect mixing so that the drug concentration is constant in each compartment. The complex transport processes are approximated by assuming that the flow rates between the compartments are proportional to the concentration differences in the compartments.

To describe the effect of a drug it is necessary to know both its concentration and how it influences the body. The relation between concentration c and its effect e is typically nonlinear. A simple model is

$$e = \frac{c}{c_0 + c} e_{\max}. \tag{4.24}$$

The effect is linear for low concentrations, and it saturates at high concentrations. The relation can also be dynamic, and it is then called *pharmacodynamics*.

Compartment Models

The simplest dynamic model for drug administration is obtained by assuming that the drug is evenly distributed in a single compartment after it has been administered and that the drug is removed at a rate proportional to the concentration. The compartments behave like stirred tanks with perfect mixing. Let c be the concentration, V the volume and q the outflow rate. Converting the description of the system into differential equations gives the model

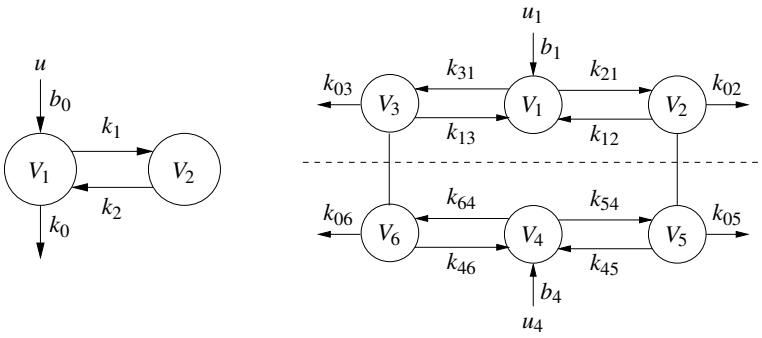
$$V\frac{dc}{dt} = -qc, \quad c \geq 0. \tag{4.25}$$

This equation has the solution $c(t) = c_0 e^{-qt/V} = c_0 e^{-kt}$, which shows that the concentration decays exponentially with the time constant $T = V/q$ after an injection. The input is introduced implicitly as an initial condition in the model (4.25). More generally, the way the input enters the model depends on how the drug is administered. For example, the input can be represented as a mass flow into the compartment where the drug is injected. A pill that is dissolved can also be interpreted as an input in terms of a mass flow rate.

The model (4.25) is called a a *one-compartment model* or a *single-pool model*. The parameter q/V is called the elimination rate constant. This simple model is often used to model the concentration in the blood plasma. By measuring the concentration at a few times, the initial concentration can be obtained by extrapolation. If the total amount of injected substance is known, the volume V can then be determined as $V = m/c_0$; this volume is called the *apparent volume of distribution*. This volume is larger than the real volume if the concentration in the plasma is lower than in other parts of the body. The model (4.25) is very simple, and there are large individual variations in the parameters. The parameters V and q are often normalized by dividing by the weight of the person. Typical parameters for aspirin are $V = 0.2$ L/kg and $q = 0.01$ (L/h)/kg. These numbers can be compared with a blood volume of 0.07 L/kg, a plasma volume of 0.05 L/kg, an intracellular fluid volume of 0.4 L/kg and an outflow of 0.0015 L/min/kg.

The simple one-compartment model captures the gross behavior of drug distribution, but it is based on many simplifications. Improved models can be obtained by considering the body as composed of several compartments. Examples of such systems are shown in Figure 4.18, where the compartments are represented as circles and the flows by arrows.

Modeling will be illustrated using the two-compartment model in Figure 4.18a. We assume that there is perfect mixing in each compartment and that the transport between the compartments is driven by concentration differences. We further assume that a drug with concentration c_0 is injected in compartment 1 at a volume flow rate of u and that the concentration in compartment 2 is the output. Let c_1 and c_2 be the concentrations of the drug in the compartments, and let V_1 and V_2 be the



(a) Two compartment model (b) Thyroid hormone model

Figure 4.18: Schematic diagrams of compartment models. (a) A simple two-compartment model. Each compartment is labeled by its volume, and arrows indicate the flow of chemical into, out of and between compartments. (b) A system with six compartments used to study the metabolism of thyroid hormone [God83]. The notation k_{ij} denotes the transport from compartment j to compartment i.

volumes of the compartments. The mass balances for the compartments are

$$\begin{aligned} V_1\frac{dc_1}{dt} &= q(c_2-c_1)-q_0c_1+c_0u, \qquad c_1\geq 0,\\ V_2\frac{dc_2}{dt} &= q(c_1-c_2), \qquad c_2\geq 0,\\ y &= c_2, \end{aligned} \tag{4.26}$$

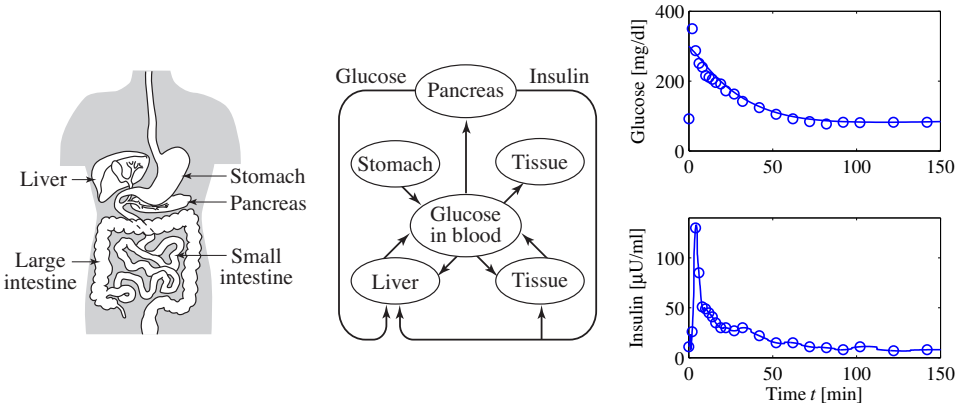
where q represents flow rate between the compartments and q_0 represents the flow rate out of compartment 1 that is not going to compartment 2. Introducing the variables $k_0 = q_0/V_1$, $k_1 = q/V_1$, $k_2 = q/V_2$ and $b_0 = c_0/V_1$ and using matrix notation, the model can be written as

$$\frac{dc}{dt} = \begin{pmatrix} -k_0-k_1 & k_1 \\ k_2 & -k_2 \end{pmatrix} c + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 0 & 1 \end{pmatrix} c. \tag{4.27}$$

Comparing this model with its graphical representation in Figure 4.18a, we find that the mathematical representation (4.27) can be written by inspection.

It should also be emphasized that simple compartment models such as the one in equation (4.27) have a limited range of validity. Low-frequency limits exist because the human body changes with time, and since the compartment model uses average concentrations, they will not accurately represent rapid changes. There are also nonlinear effects that influence transportation between the compartments.

Compartment models are widely used in medicine, engineering and environmental science. An interesting property of these systems is that variables like concentration and mass are always positive. An essential difficulty in compartment modeling is deciding how to divide a complex system into compartments. Compartment models can also be nonlinear, as illustrated in the next section.



(a) Relevant body organs (b) Schematic diagram (c) Intravenous injection

Figure 4.19: Insulin–glucose dynamics. (a) Sketch of body parts involved in the control of glucose. (b) Schematic diagram of the system. (c) Responses of insulin and glucose when glucose in injected intravenously. From [PB86].

Insulin–Glucose Dynamics

It is essential that the blood glucose concentration in the body is kept within a narrow range (0.7–1.1 g/L). Glucose concentration is influenced by many factors like food intake, digestion and exercise. A schematic picture of the relevant parts of the body is shown in Figures 4.19a and b.

There is a sophisticated mechanism that regulates glucose concentration. Glucose concentration is maintained by the pancreas, which secretes the hormones insulin and glucagon. Glucagon is released into the bloodstream when the glucose level is low. It acts on cells in the liver that release glucose. Insulin is secreted when the glucose level is high, and the glucose level is lowered by causing the liver and other cells to take up more glucose. In diseases like juvenile diabetes the pancreas is unable to produce insulin and the patient must inject insulin into the body to maintain a proper glucose level.

The mechanisms that regulate glucose and insulin are complicated; dynamics with time scales that range from seconds to hours have been observed. Models of different complexity have been developed. The models are typically tested with data from experiments where glucose is injected intravenously and insulin and glucose concentrations are measured at regular time intervals.

A relatively simple model called the *minimal model* was developed by Bergman and coworkers [Ber89]. This models uses two compartments, one representing the concentration of glucose in the bloodstream and the other representing the concentration of insulin in the interstitial fluid. Insulin in the bloodstream is considered an input. The reaction of glucose to insulin can be modeled by the equations

$$\frac{dx_1}{dt} = -(p_1 + x_2)x_1 + p_1 g_e, \qquad \frac{dx_2}{dt} = -p_2 x_2 + p_3(u - i_e), \tag{4.28}$$

where g_e and i_e represent the equilibrium values of glucose and insulin, x_1 is the concentration of glucose and x_2 is proportional to the concentration of interstitial insulin. Notice the presence of the term x_2x_1 in the first equation. Also notice that the model does not capture the complete feedback loop because it does not describe how the pancreas reacts to the glucose. Figure 4.19c shows a fit of the model to a test on a normal person where glucose was injected intravenously at time $t = 0$. The glucose concentration rises rapidly, and the pancreas responds with a rapid spikelike injection of insulin. The glucose and insulin levels then gradually approach the equilibrium values.

Models of the type in equation (4.28) and more complicated models having many compartments have been developed and fitted to experimental data. A difficulty in modeling is that there are significant variations in model parameters over time and for different patients. For example, the parameter p_1 in equation (4.28) has been reported to vary with an order of magnitude for healthy individuals. The models have been used for diagnosis and to develop schemes for the treatment of persons with diseases. Attempts to develop a fully automatic artificial pancreas have been hampered by the lack of reliable sensors.

The papers by Widmark and Tandberg [WT24] and Teorell [Teo37] are classics in pharmacokinetics, which is now an established discipline with many textbooks [Dos68, Jac72, GP82]. Because of its medical importance, pharmacokinetics is now an essential component of drug development. The book by Riggs [Rig63] is a good source for the modeling of physiological systems, and a more mathematical treatment is given in [KS01]. Compartment models are discussed in [God83]. The problem of determining rate coefficients from experimental data is discussed in [BÅ70] and [God83]. There are many publications on the insulin–glucose model. The minimal model is discussed in [CT84, Ber89] and more recent references are [MLK06, FCF+06].

4.7 Population Dynamics

Population growth is a complex dynamic process that involves the interaction of one or more species with their environment and the larger ecosystem. The dynamics of population groups are interesting and important in many different areas of social and environmental policy. There are examples where new species have been introduced into new habitats, sometimes with disastrous results. There have also been attempts to control population growth both through incentives and through legislation. In this section we describe some of the models that can be used to understand how populations evolve with time and as a function of their environments.

Logistic Growth Model

Let x be the population of a species at time t. A simple model is to assume that the birth rates and mortality rates are proportional to the total population. This gives

the linear model

$$\frac{dx}{dt} = bx - dx = (b-d)x = rx, \qquad x \geq 0, \tag{4.29}$$

where birth rate b and mortality rate d are parameters. The model gives an exponential increase if $b > d$ or an exponential decrease if $b < d$. A more realistic model is to assume that the birth rate decreases when the population is large. The following modification of the model (4.29) has this property:

$$\frac{dx}{dt} = rx(1 - \frac{x}{k}), \qquad x \geq 0, \tag{4.30}$$

where k is the *carrying capacity* of the environment. The model (4.30) is called the *logistic growth model*.

Predator–Prey Models

A more sophisticated model of population dynamics includes the effects of competing populations, where one species may feed on another. This situation, referred to as the *predator–prey problem*, was introduced in Example 3.3, where we developed a discrete-time model that captured some of the features of historical records of lynx and hare populations.

In this section, we replace the difference equation model used there with a more sophisticated differential equation model. Let $H(t)$ represent the number of hares (prey) and let $L(t)$ represent the number of lynxes (predator). The dynamics of the system are modeled as

$$\begin{aligned} \frac{dH}{dt} &= rH\left(1 - \frac{H}{k}\right) - \frac{aHL}{c+H}, & H \geq 0, \\ \frac{dL}{dt} &= b\frac{aHL}{c+H} - dL, & L \geq 0. \end{aligned} \tag{4.31}$$

In the first equation, r represents the growth rate of the hares, k represents the maximum population of the hares (in the absence of lynxes), a represents the interaction term that describes how the hares are diminished as a function of the lynx population and c controls the prey consumption rate for low hare population. In the second equation, b represents the growth coefficient of the lynxes and d represents the mortality rate of the lynxes. Note that the hare dynamics include a term that resembles the logistic growth model (4.30).

Of particular interest are the values at which the population values remain constant, called *equilibrium points*. The equilibrium points for this system can be determined by setting the right-hand side of the above equations to zero. Letting H_e and L_e represent the equilibrium state, from the second equation we have

$$L_e = 0 \quad \text{or} \quad H_e^* = \frac{cd}{ab-d}. \tag{4.32}$$

Substituting this into the first equation, we have that for $L_e = 0$ either $H_e = 0$ or

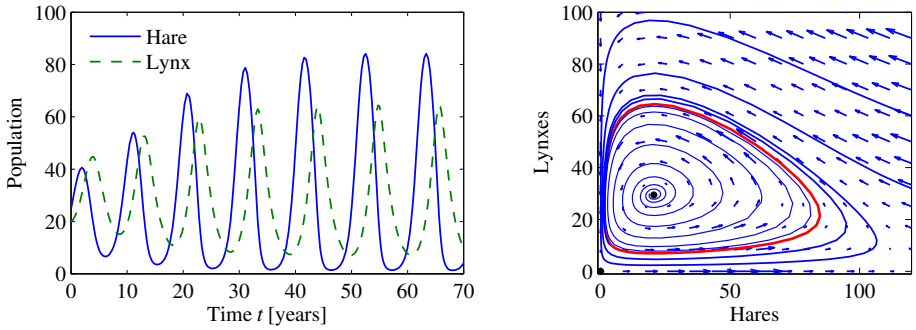


Figure 4.20: Simulation of the predator–prey system. The figure on the left shows a simulation of the two populations as a function of time. The figure on the right shows the populations plotted against each other, starting from different values of the population. The oscillation seen in both figures is an example of a *limit cycle*. The parameter values used for the simulations are $a = 3.2$, $b = 0.6$, $c = 50$, $d = 0.56$, $k = 125$ and $r = 1.6$.

$H_e = k$. For $L_e \neq 0$, we obtain

$$L_e^* = \frac{rH_e(c+H_e)}{aH_e}\left(1 - \frac{H_e}{k}\right) = \frac{bcr(abk - cd - dk)}{(ab-d)^2 k}. \tag{4.33}$$

Thus, we have three possible equilibrium points $x_e = (L_e, H_e)$:

$$x_e = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad x_e = \begin{pmatrix} k \\ 0 \end{pmatrix}, \qquad x_e = \begin{pmatrix} H_e^* \\ L_e^* \end{pmatrix},$$

where H_e^* and L_e^* are given in equations (4.32) and (4.33). Note that the equilibrium populations may be negative for some parameter values, corresponding to a nonachievable equilibrium point.

Figure 4.20 shows a simulation of the dynamics starting from a set of population values near the nonzero equilibrium values. We see that for this choice of parameters, the simulation predicts an oscillatory population count for each species, reminiscent of the data shown in Figure 3.7.

Volume I of the two-volume set by J. D. Murray [Mur04] give a broad coverage of population dynamics.

Exercises

4.1 (Cruise control) Consider the cruise control example described in Section 4.1. Build a simulation that re-creates the response to a hill shown in Figure 4.3b and show the effects of increasing and decreasing the mass of the car by 25%. Redesign the controller (using trial and error is fine) so that it returns to within 1% of the desired speed within 3 s of encountering the beginning of the hill.

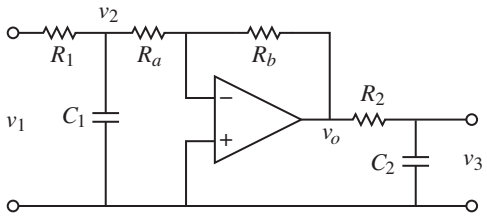
4.2 (Bicycle dynamics) Show that the dynamics of a bicycle frame given by equation (4.5) can be approximated in state space form as

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ mgh/J & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} Dv_0/(bJ) \\ mv_0^2 h/(bJ) \end{pmatrix} u,$$
$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} x,$$

where the input u is the steering angle δ and the output y is the tilt angle φ. What do the states x_1 and x_2 represent?

4.3 (Bicycle steering) Combine the bicycle model given by equation (4.5) and the model for steering kinematics in Example 3.10 to obtain a model that describes the path of the center of mass of the bicycle.

4.4 (Operational amplifier circuit) Consider the op amp circuit shown below.

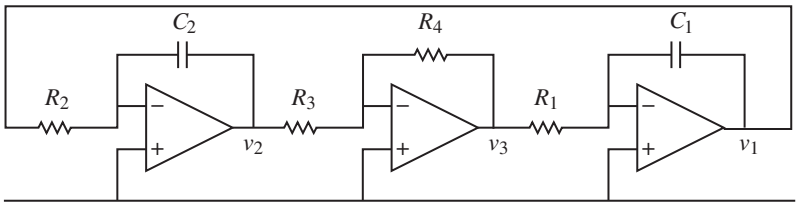


Show that the dynamics can be written in state space form as

$$\frac{dx}{dt} = \begin{pmatrix} -\dfrac{1}{R_1C_1} - \dfrac{1}{R_aC_1} & 0 \\ \dfrac{R_b}{R_a}\dfrac{1}{R_2C_2} & -\dfrac{1}{R_2C_2} \end{pmatrix} x + \begin{pmatrix} \dfrac{1}{R_1C_1} \\ 0 \end{pmatrix} u, \quad y = \begin{pmatrix} 0 & 1 \end{pmatrix} x,$$

where $u = v_1$ and $y = v_3$. (Hint: Use v_2 and v_3 as your state variables.)

4.5 (Operational amplifier oscillator) The op amp circuit shown below is an implementation of an oscillator.



Show that the dynamics can be written in state space form as

$$\frac{dx}{dt} = \begin{pmatrix} 0 & \dfrac{R_4}{R_1R_3C_1} \\ -\dfrac{1}{R_2C_2} & 0 \end{pmatrix} x,$$

where the state variables represent the voltages across the capacitors $x_1 = v_1$ and $x_2 = v_2$.

4.6 (Congestion control using RED [LPW+02]) A number of improvements can be made to the model for Internet congestion control presented in Section 4.4. To ensure that the router's buffer size remains positive, we can modify the buffer dynamics to satisfy

$$\frac{db_l}{dt} = \begin{cases} s_l - c_l & b_l > 0 \\ \text{sat}_{(0,\infty)}(s_l - c_l) & b_l = 0. \end{cases}$$

In addition, we can model the drop probability of a packet based on how close we are to the buffer limits, a mechanism known as random early detection (RED):

$$p_l = m_l(a_l) = \begin{cases} 0 & a_l(t) \le b_l^{\text{lower}} \\ \rho_l r_i(t) - \rho_l b_l^{\text{lower}} & b_l^{\text{lower}} < a_l(t) < b_l^{\text{upper}} \\ \eta_l r_i(t) - (1 - 2b_l^{\text{upper}}) & b_l^{\text{upper}} \le a_l(t) < 2b_l^{\text{upper}} \\ 1 & a_l(t) \ge 2b_l^{\text{upper}}, \end{cases}$$

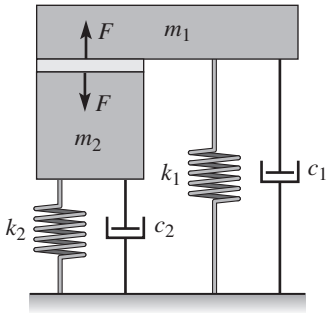
$$\frac{da_l}{dt} = -\alpha_l c_l (a_l - b_l),$$

where α_l, b_l^{upper}, b_l^{lower} and p_l^{upper} are parameters for the RED protocol.

Using the model above, write a simulation for the system and find a set of parameter values for which there is a stable equilibrium point and a set for which the system exhibits oscillatory solutions. The following sets of parameters should be explored:

$$\begin{aligned} &N = 20, 30, \ldots, 60, && b_l^{\text{lower}} = 40 \text{ pkts}, && \rho_l = 0.1, \\ &c = 8, 9, \ldots, 15 \text{ pkts/ms}, && b_l^{\text{upper}} = 540 \text{ pkts}, && \alpha_l = 10^{-4}, \\ &\tau = 55, 60, \ldots, 100 \text{ ms}. \end{aligned}$$

4.7 (Atomic force microscope with piezo tube) A schematic diagram of an AFM where the vertical scanner is a piezo tube with preloading is shown below.



Show that the dynamics can be written as

$$(m_1 + m_2)\frac{d^2 z_1}{dt^2} + (c_1 + c_2)\frac{dz_1}{dt} + (k_1 + k_2) z_1 = m_2 \frac{d^2 l}{dt^2} + c_2 \frac{dl}{dt} + k_2 l.$$

Are there parameter values that make the dynamics particularly simple?

4.8 (Drug administration) The metabolism of alcohol in the body can be modeled by the nonlinear compartment model

$$V_b \frac{dc_b}{dt} = q(c_l - c_b) + q_{iv}, \qquad V_l \frac{dc_l}{dt} = q(c_b - c_l) - q_{\max} \frac{c_l}{c_0 + c_l} + q_{gi},$$

where $V_b = 48$ L and $V_l = 0.6$ L are the apparent volumes of distribution of body water and liver water, c_b and c_l are the concentrations of alcohol in the compartments, q_{iv} and q_{gi} are the injection rates for intravenous and gastrointestinal intake, $q = 1.5$ L/min is the total hepatic blood flow, $q_{\max} = 2.75$ mmol/min and $c_0 = 0.1$ mmol/L. Simulate the system and compute the concentration in the blood for oral and intravenous doses of 12 g and 40 g of alcohol.

4.9 (Population dynamics) Consider the model for logistic growth given by equation (4.30). Show that the maximum growth rate occurs when the size of the population is half of the steady-state value.

4.10 (Fisheries management) The dynamics of a commercial fishery can be described by the following simple model:

$$\frac{dx}{dt} = f(x) - h(x,u), \qquad y = bh(x,u) - cu,$$

where x is the total biomass, $f(x) = rx(1 - x/k)$ is the growth rate and $h(x,u) = axu$ is the harvesting rate. The output y is the rate of revenue, and the parameters a, b and c are constants representing the price of fish and the cost of fishing. Show that there is an equilibrium where the steady-state biomass is $x_e = c/(ab)$. Compare with the situation when the biomass is regulated to a constant value and find the maximum sustainable return in that case.

Chapter Five
Dynamic Behavior

It Don't Mean a Thing If It Ain't Got That Swing.

Duke Ellington (1899–1974)

In this chapter we present a broad discussion of the behavior of dynamical systems focused on systems modeled by nonlinear differential equations. This allows us to consider equilibrium points, stability, limit cycles and other key concepts in understanding dynamic behavior. We also introduce some methods for analyzing the global behavior of solutions.

5.1 Solving Differential Equations

In the previous two chapters we saw that one of the methods of modeling dynamical systems is through the use of ordinary differential equations (ODEs). A state space, input/output system has the form

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x,u), \tag{5.1}$$

where $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ is the input and $y \in \mathbb{R}^q$ is the output. The smooth maps $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ and $h : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^q$ represent the dynamics and measurements for the system. In general, they can be nonlinear functions of their arguments. We will sometimes focus on single-input, single-output (SISO) systems, for which $p = q = 1$.

We begin by investigating systems in which the input has been set to a function of the state, $u = \alpha(x)$. This is one of the simplest types of feedback, in which the system regulates its own behavior. The differential equations in this case become

$$\frac{dx}{dt} = f(x, \alpha(x)) =: F(x). \tag{5.2}$$

To understand the dynamic behavior of this system, we need to analyze the features of the solutions of equation (5.2). While in some simple situations we can write down the solutions in analytical form, often we must rely on computational approaches. We begin by describing the class of solutions for this problem.

We say that $x(t)$ is a *solution* of the differential equation (5.2) on the time interval $t_0 \in \mathbb{R}$ to $t_f \in \mathbb{R}$ if

$$\frac{dx(t)}{dt} = F(x(t)) \quad \text{for all } t_0 < t < t_f.$$

A given differential equation may have many solutions. We will most often be interested in the *initial value problem*, where $x(t)$ is prescribed at a given time $t_0 \in \mathbb{R}$ and we wish to find a solution valid for all *future* time $t > t_0$.

We say that $x(t)$ is a solution of the differential equation (5.2) with initial value $x_0 \in \mathbb{R}^n$ at $t_0 \in \mathbb{R}$ if

$$x(t_0) = x_0 \quad \text{and} \quad \frac{dx(t)}{dt} = F(x(t)) \quad \text{for all } t_0 < t < t_f.$$

For most differential equations we will encounter, there is a *unique* solution that is defined for $t_0 < t < t_f$. The solution may be defined for all time $t > t_0$, in which case we take $t_f = \infty$. Because we will primarily be interested in solutions of the initial value problem for ODEs, we will usually refer to this simply as the solution of an ODE.

We will typically assume that t_0 is equal to 0. In the case when F is independent of time (as in equation (5.2)), we can do so without loss of generality by choosing a new independent (time) variable, $\tau = t - t_0$ (Exercise 5.1).

Example 5.1 Damped oscillator
Consider a damped linear oscillator with dynamics of the form

$$\ddot{q} + 2\zeta\omega_0\dot{q} + \omega_0^2 q = 0,$$

where q is the displacement of the oscillator from its rest position. These dynamics are equivalent to those of a spring–mass system, as shown in Exercise 3.6. We assume that $\zeta < 1$, corresponding to a lightly damped system (the reason for this particular choice will become clear later). We can rewrite this in state space form by setting $x_1 = q$ and $x_2 = \dot{q}/\omega_0$, giving

$$\frac{dx_1}{dt} = \omega_0 x_2, \qquad \frac{dx_2}{dt} = -\omega_0 x_1 - 2\zeta\omega_0 x_2.$$

In vector form, the right-hand side can be written as

$$F(x) = \begin{pmatrix} \omega_0 x_2 \\ -\omega_0 x_1 - 2\zeta\omega_0 x_2 \end{pmatrix}.$$

The solution to the initial value problem can be written in a number of different ways and will be explored in more detail in Chapter 6. Here we simply assert that the solution can be written as

$$x_1(t) = e^{-\zeta\omega_0 t}\left(x_{10}\cos\omega_d t + \frac{1}{\omega_d}(\omega_0\zeta x_{10} + x_{20})\sin\omega_d t\right),$$
$$x_2(t) = e^{-\zeta\omega_0 t}\left(x_{20}\cos\omega_d t - \frac{1}{\omega_d}(\omega_0^2 x_{10} + \omega_0\zeta x_{20})\sin\omega_d t\right),$$

where $x_0 = (x_{10}, x_{20})$ is the initial condition and $\omega_d = \omega_0\sqrt{1-\zeta^2}$. This solution can be verified by substituting it into the differential equation. We see that the solution is explicitly dependent on the initial condition, and it can be shown that this solution is unique. A plot of the initial condition response is shown in Figure 5.1.

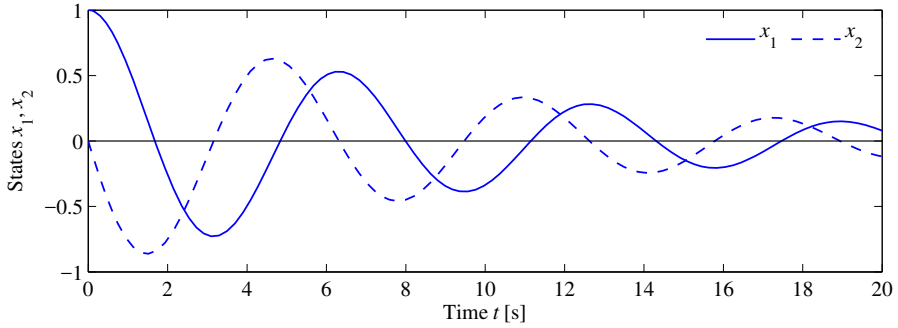


Figure 5.1: Response of the damped oscillator to the initial condition $x_0 = (1,0)$. The solution is unique for the given initial conditions and consists of an oscillatory solution for each state, with an exponentially decaying magnitude.

We note that this form of the solution holds only for $0 < \zeta < 1$, corresponding to an "underdamped" oscillator. ∇

Without imposing some mathematical conditions on the function F, the differential equation (5.2) may not have a solution for all t, and there is no guarantee that the solution is unique. We illustrate these possibilities with two examples.

Example 5.2 Finite escape time
Let $x \in \mathbb{R}$ and consider the differential equation

$$\frac{dx}{dt} = x^2 \tag{5.3}$$

with the initial condition $x(0) = 1$. By differentiation we can verify that the function

$$x(t) = \frac{1}{1-t}$$

satisfies the differential equation and that it also satisfies the initial condition. A graph of the solution is given in Figure 5.2a; notice that the solution goes to infinity as t goes to 1. We say that this system has *finite escape time*. Thus the solution exists only in the time interval $0 \le t < 1$. ∇

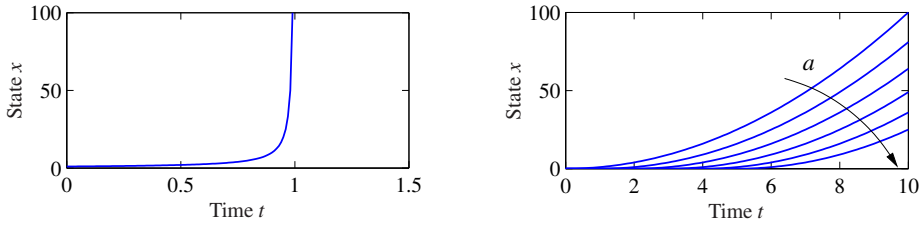
Example 5.3 Nonunique solution
Let $x \in \mathbb{R}$ and consider the differential equation

$$\frac{dx}{dt} = 2\sqrt{x} \tag{5.4}$$

with initial condition $x(0) = 0$. We can show that the function

$$x(t) = \begin{cases} 0 & \text{if } 0 \le t \le a \\ (t-a)^2 & \text{if } t > a \end{cases}$$

satisfies the differential equation for all values of the parameter $a \ge 0$. To see this,



(a) Finite escape time (b) Nonunique solutions

Figure 5.2: Existence and uniqueness of solutions. Equation (5.3) has a solution only for time $t < 1$, at which point the solution goes to ∞, as shown in (a). Equation (5.4) is an example of a system with many solutions, as shown in (b). For each value of a, we get a different solution starting from the same initial condition.

we differentiate $x(t)$ to obtain

$$\frac{dx}{dt} = \begin{cases} 0 & \text{if } 0 \le t \le a \\ 2(t-a) & \text{if } t > a, \end{cases}$$

and hence $\dot{x} = 2\sqrt{x}$ for all $t \ge 0$ with $x(0) = 0$. A graph of some of the possible solutions is given in Figure 5.2b. Notice that in this case there are many solutions to the differential equation. ∇

These simple examples show that there may be difficulties even with simple differential equations. Existence and uniqueness can be guaranteed by requiring that the function F have the property that for some fixed $c \in \mathbb{R}$,

$$\|F(x) - F(y)\| < c\|x - y\| \quad \text{for all } x, y,$$

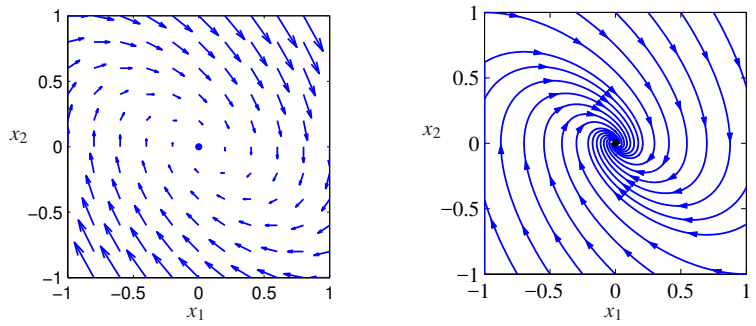
which is called *Lipschitz continuity*. A sufficient condition for a function to be Lipschitz is that the Jacobian $\partial F/\partial x$ is uniformly bounded for all x. The difficulty in Example 5.2 is that the derivative $\partial F/\partial x$ becomes large for large x, and the difficulty in Example 5.3 is that the derivative $\partial F/\partial x$ is infinite at the origin.

5.2 Qualitative Analysis

The qualitative behavior of nonlinear systems is important in understanding some of the key concepts of stability in nonlinear dynamics. We will focus on an important class of systems known as planar dynamical systems. These systems have two state variables $x \in \mathbb{R}^2$, allowing their solutions to be plotted in the (x_1, x_2) plane. The basic concepts that we describe hold more generally and can be used to understand dynamical behavior in higher dimensions.

Phase Portraits

A convenient way to understand the behavior of dynamical systems with state $x \in \mathbb{R}^2$ is to plot the phase portrait of the system, briefly introduced in Chapter 3.



(a) Vector field (b) Phase portrait

Figure 5.3: Phase portraits. (a) This plot shows the vector field for a planar dynamical system. Each arrow shows the velocity at that point in the state space. (b) This plot includes the solutions (sometimes called streamlines) from different initial conditions, with the vector field superimposed.

We start by introducing the concept of a *vector field*. For a system of ordinary differential equations

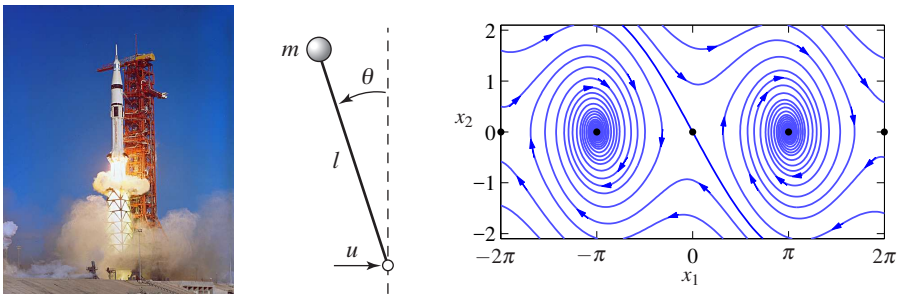
$$\frac{dx}{dt} = F(x),$$

the right-hand side of the differential equation defines at every $x \in \mathbb{R}^n$ a velocity $F(x) \in \mathbb{R}^n$. This velocity tells us how x changes and can be represented as a vector $F(x) \in \mathbb{R}^n$.

For planar dynamical systems, each state corresponds to a point in the plane and $F(x)$ is a vector representing the velocity of that state. We can plot these vectors on a grid of points in the plane and obtain a visual image of the dynamics of the system, as shown in Figure 5.3a. The points where the velocities are zero are of particular interest since they define stationary points of the flow: if we start at such a state, we stay at that state.

A *phase portrait* is constructed by plotting the flow of the vector field corresponding to the planar dynamical system. That is, for a set of initial conditions, we plot the solution of the differential equation in the plane $\mathbb{R}^2$. This corresponds to following the arrows at each point in the phase plane and drawing the resulting trajectory. By plotting the solutions for several different initial conditions, we obtain a phase portrait, as show in Figure 5.3b. Phase portraits are also sometimes called *phase plane diagrams*.

Phase portraits give insight into the dynamics of the system by showing the solutions plotted in the (two-dimensional) state space of the system. For example, we can see whether all trajectories tend to a single point as time increases or whether there are more complicated behaviors. In the example in Figure 5.3, corresponding to a damped oscillator, the solutions approach the origin for all initial conditions. This is consistent with our simulation in Figure 5.1, but it allows us to infer the behavior for all initial conditions rather than a single initial condition. However, the phase portrait does not readily tell us the rate of change of the states (although



(a) (b) (c)

Figure 5.4: Equilibrium points for an inverted pendulum. An inverted pendulum is a model for a class of balance systems in which we wish to keep a system upright, such as a rocket (a). Using a simplified model of an inverted pendulum (b), we can develop a phase portrait that shows the dynamics of the system (c). The system has multiple equilibrium points, marked by the solid dots along the $x_2 = 0$ line.

this can be inferred from the lengths of the arrows in the vector field plot).

Equilibrium Points and Limit Cycles

An *equilibrium point* of a dynamical system represents a stationary condition for the dynamics. We say that a state x_e is an equilibrium point for a dynamical system

$$\frac{dx}{dt} = F(x)$$

if $F(x_e) = 0$. If a dynamical system has an initial condition $x(0) = x_e$, then it will stay at the equilibrium point: $x(t) = x_e$ for all $t \geq 0$, where we have taken $t_0 = 0$.

Equilibrium points are one of the most important features of a dynamical system since they define the states corresponding to constant operating conditions. A dynamical system can have zero, one or more equilibrium points.

Example 5.4 Inverted pendulum

Consider the inverted pendulum in Figure 5.4, which is a part of the balance system we considered in Chapter 3. The inverted pendulum is a simplified version of the problem of stabilizing a rocket: by applying forces at the base of the rocket, we seek to keep the rocket stabilized in the upright position. The state variables are the angle $\theta = x_1$ and the angular velocity $d\theta/dt = x_2$, the control variable is the acceleration u of the pivot and the output is the angle θ.

For simplicity we assume that $mgl/J_t = 1$, $l/J_t = 1$ and set $c = \gamma/J_t$, so that the dynamics (equation (3.11)) become

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ \sin x_1 - cx_2 + u\cos x_1 \end{pmatrix}. \tag{5.5}$$

This is a nonlinear time-invariant system of second order. This same set of equations can also be obtained by appropriate normalization of the system dynamics as illustrated in Example 3.9.

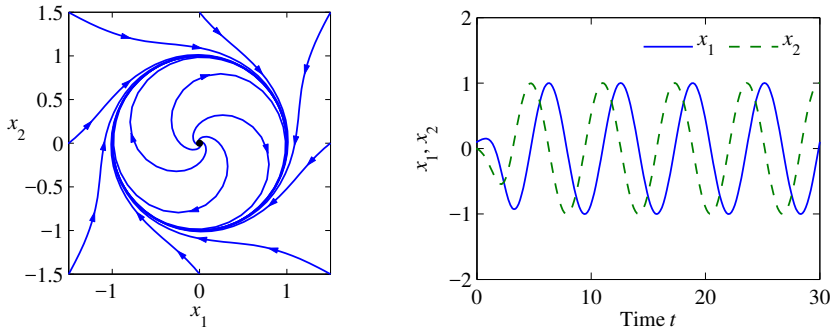


Figure 5.5: Phase portrait and time domain simulation for a system with a limit cycle. The phase portrait (a) shows the states of the solution plotted for different initial conditions. The limit cycle corresponds to a closed loop trajectory. The simulation (b) shows a single solution plotted as a function of time, with the limit cycle corresponding to a steady oscillation of fixed amplitude.

We consider the open loop dynamics by setting $u = 0$. The equilibrium points for the system are given by

$$x_e = \begin{pmatrix} \pm n\pi \\ 0 \end{pmatrix},$$

where $n = 0, 1, 2, \ldots$. The equilibrium points for n even correspond to the pendulum pointing up and those for n odd correspond to the pendulum hanging down. A phase portrait for this system (without corrective inputs) is shown in Figure 5.4c. The phase portrait shows $-2\pi \leq x_1 \leq 2\pi$, so five of the equilibrium points are shown. ∇

Nonlinear systems can exhibit rich behavior. Apart from equilibria they can also exhibit stationary periodic solutions. This is of great practical value in generating sinusoidally varying voltages in power systems or in generating periodic signals for animal locomotion. A simple example is given in Exercise 5.12, which shows the circuit diagram for an electronic oscillator. A normalized model of the oscillator is given by the equation

$$\frac{dx_1}{dt} = x_2 + x_1(1 - x_1^2 - x_2^2), \qquad \frac{dx_2}{dt} = -x_1 + x_2(1 - x_1^2 - x_2^2). \tag{5.6}$$

The phase portrait and time domain solutions are given in Figure 5.5. The figure shows that the solutions in the phase plane converge to a circular trajectory. In the time domain this corresponds to an oscillatory solution. Mathematically the circle is called a *limit cycle*. More formally, we call an nonconstant solution $x(t)$ a limit cycle of period $T > 0$ if $x(t+T) = x(t)$ for all $t \in \mathbb{R}$.

There are methods for determining limit cycles for second-order systems, but for general higher-order systems we have to resort to computational analysis. Computer algorithms find limit cycles by searching for periodic trajectories in state

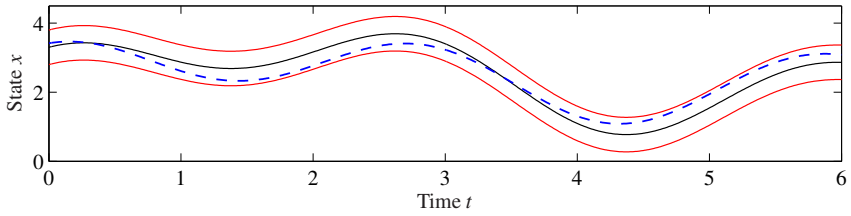


Figure 5.6: Illustration of Lyapunov's concept of a stable solution. The solution represented by the solid line is stable if we can guarantee that all solutions remain within a tube of diameter ε by choosing initial conditions sufficiently close the solution.

space that satisfy the dynamics of the system. In many situations, stable limit cycles can be found by simulating the system with different initial conditions.

5.3 Stability

The stability of a solution determines whether or not solutions nearby the solution remain close, get closer or move further away. We now give a formal definition of stability and describe tests for determining whether a solution is stable.

Definitions

Let $x(t;a)$ be a solution to the differential equation with initial condition a. A solution is *stable* if other solutions that start near a stay close to $x(t;a)$. Formally, we say that the solution $x(t;a)$ is stable if for all $\varepsilon > 0$, there exists a $\delta > 0$ such that

$$\|b-a\| < \delta \quad \Longrightarrow \quad \|x(t;b)-x(t;a)\| < \varepsilon \quad \text{for all } t > 0.$$

Note that this definition does not imply that $x(t;b)$ approaches $x(t;a)$ as time increases but just that it stays nearby. Furthermore, the value of δ may depend on ε, so that if we wish to stay very close to the solution, we may have to start very, very close ($\delta \ll \varepsilon$). This type of stability, which is illustrated in Figure 5.6, is also called *stability in the sense of Lyapunov*. If a solution is stable in this sense and the trajectories do not converge, we say that the solution is *neutrally stable*.

An important special case is when the solution $x(t;a) = x_e$ is an equilibrium solution. Instead of saying that the solution is stable, we simply say that the equilibrium point is stable. An example of a neutrally stable equilibrium point is shown in Figure 5.7. From the phase portrait, we see that if we start near the equilibrium point, then we stay near the equilibrium point. Indeed, for this example, given any ε that defines the range of possible initial conditions, we can simply choose $\delta = \varepsilon$ to satisfy the definition of stability since the trajectories are perfect circles.

A solution $x(t;a)$ is *asymptotically stable* if it is stable in the sense of Lyapunov and also $x(t;b)$ approaches $x(t;a)$ as t approaches ∞ for b sufficiently close to a. Hence, the solution $x(t;a)$ is asymptotically stable if for every $\varepsilon > 0$ there exists a

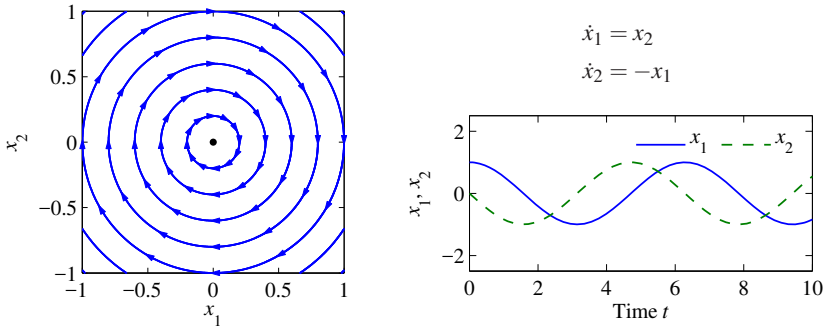


Figure 5.7: Phase portrait and time domain simulation for a system with a single stable equilibrium point. The equilibrium point x_e at the origin is stable since all trajectories that start near x_e stay near x_e.

$\delta > 0$ such that

$$\|b-a\| < \delta \quad \Longrightarrow \quad \|x(t;b)-x(t;a)\| < \varepsilon \text{ and } \lim_{t\to\infty} \|x(t;b)-x(t;a)\| = 0.$$

This corresponds to the case where all nearby trajectories converge to the stable solution for large time. Figure 5.8 shows an example of an asymptotically stable equilibrium point. Indeed, as seen in the phase portrait, not only do all trajectories stay near the equilibrium point at the origin, but they also all approach the origin as t gets large (the directions of the arrows on the phase portrait show the direction in which the trajectories move).

A solution $x(t;a)$ is *unstable* if it is not stable. More specifically, we say that a solution $x(t;a)$ is unstable if given some $\varepsilon > 0$, there does *not* exist a $\delta > 0$ such that if $\|b-a\| < \delta$, then $\|x(t;b)-x(t;a)\| < \varepsilon$ for all t. An example of an unstable equilibrium point is shown in Figure 5.9.

The definitions above are given without careful description of their domain of applicability. More formally, we define a solution to be *locally stable* (or *locally*

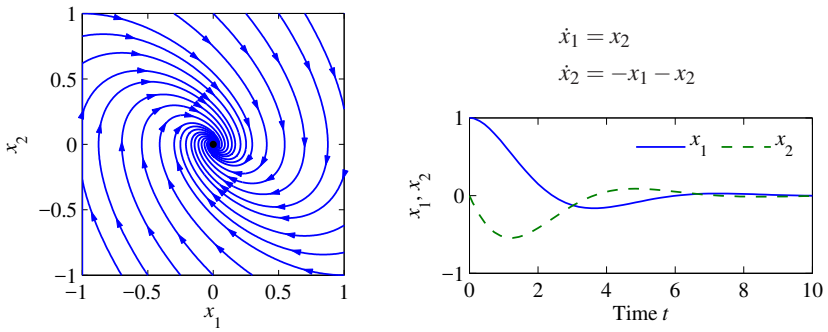


Figure 5.8: Phase portrait and time domain simulation for a system with a single asymptotically stable equilibrium point. The equilibrium point x_e at the origin is asymptotically stable since the trajectories converge to this point as $t \to \infty$.

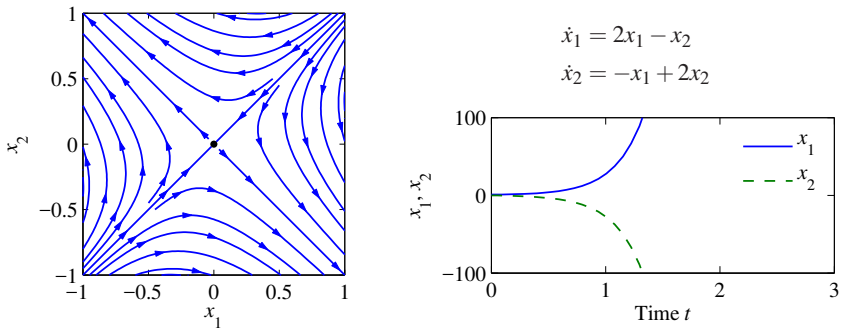


Figure 5.9: Phase portrait and time domain simulation for a system with a single unstable equilibrium point. The equilibrium point x_e at the origin is unstable since not all trajectories that start near x_e stay near x_e. The sample trajectory on the right shows that the trajectories very quickly depart from zero.

asymptotically stable) if it is stable for all initial conditions $x \in B_r(a)$, where

$$B_r(a) = \{x : \|x - a\| < r\}$$

is a ball of radius r around a and $r > 0$. A solution is *globally asymptotically stable* if it is asymptotically stable for all $r > 0$.

For planar dynamical systems, equilibrium points have been assigned names based on their stability type. An asymptotically stable equilibrium point is called a *sink* or sometimes an *attractor*. An unstable equilibrium point can be either a *source*, if all trajectories lead away from the equilibrium point, or a *saddle*, if some trajectories lead to the equilibrium point and others move away (this is the situation pictured in Figure 5.9). Finally, an equilibrium point that is stable but not asymptotically stable (i.e., neutrally stable, such as the one in Figure 5.7) is called a *center*.

Example 5.5 Congestion control

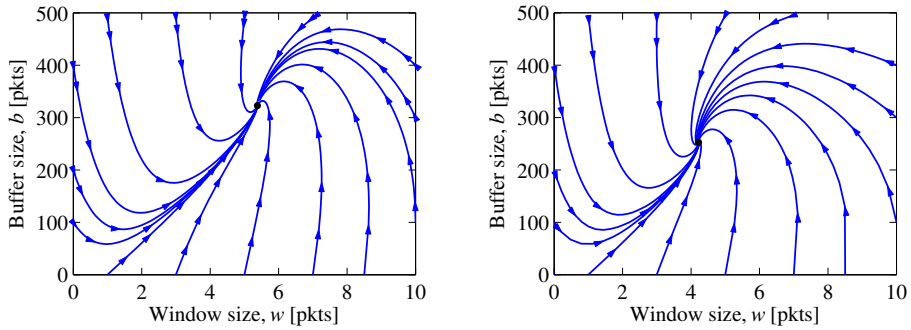
The model for congestion control in a network consisting of N identical computers connected to a single router, described in more detail in Section 4.4, is given by

$$\frac{dw}{dt} = \frac{c}{b} - \rho c\left(1 + \frac{w^2}{2}\right), \qquad \frac{db}{dt} = N\frac{wc}{b} - c,$$

where w is the window size and b is the buffer size of the router. The equilibrium points are given by

$$b = Nw \quad \text{with} \quad w\left(1 + \frac{w^2}{2}\right) = \frac{\rho}{N}.$$

Phase portraits are shown in Figure 5.10 for two different sets of parameter values. In each case we see that the system converges to an equilibrium point in which the buffer is below its full capacity of 500 packets. The equilibrium size of the buffer represents a balance between the transmission rates for the sources and the



(a) $\rho = 2 \times 10^{-4}$, $c = 10$ pkts/ms (b) $\rho = 4 \times 10^{-4}$, $c = 20$ pkts/ms

Figure 5.10: Phase portraits for a congestion control protocol running with $N = 60$ identical source computers. The equilibrium values correspond to a fixed window at the source, which results in a steady-state buffer size and corresponding transmission rate. A faster link (b) uses a smaller buffer size since it can handle packets at a higher rate.

capacity of the link. We see from the phase portraits that the equilibrium points are asymptotically stable since all initial conditions result in trajectories that converge to these points. ∇

Stability of Linear Systems

A linear dynamical system has the form

$$\frac{dx}{dt} = Ax, \quad x(0) = x_0, \tag{5.7}$$

where $A \in \mathbb{R}^{n \times n}$ is a square matrix, corresponding to the dynamics matrix of a linear control system (3.7). For a linear system, the stability of the equilibrium at the origin can be determined from the eigenvalues of the matrix A:

$$\lambda(A) := \{s \in \mathbb{C} : \det(sI - A) = 0\}.$$

The polynomial $\det(sI - A)$ is the *characteristic polynomial* and the eigenvalues are its roots. We use the notation λ_j for the jth eigenvalue of A, so that $\lambda_j \in \lambda(A)$. In general λ can be complex-valued, although if A is real-valued, then for any eigenvalue λ, its complex conjugate λ^* will also be an eigenvalue. The origin is always an equilibrium for a linear system. Since the stability of a linear system depends only on the matrix A, we find that stability is a property of the system. For a linear system we can therefore talk about the stability of the system rather than the stability of a particular solution or equilibrium point.

The easiest class of linear systems to analyze are those whose system matrices

are in diagonal form. In this case, the dynamics have the form

$$\frac{dx}{dt} = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix} x. \tag{5.8}$$

It is easy to see that the state trajectories for this system are independent of each other, so that we can write the solution in terms of n individual systems $\dot{x}_j = \lambda_j x_j$. Each of these scalar solutions is of the form

$$x_j(t) = e^{\lambda_j t} x_j(0).$$

We see that the equilibrium point $x_e = 0$ is stable if $\lambda_j \leq 0$ and asymptotically stable if $\lambda_j < 0$.

Another simple case is when the dynamics are in the block diagonal form

$$\frac{dx}{dt} = \begin{pmatrix} \sigma_1 & \omega_1 & & 0 & 0 \\ -\omega_1 & \sigma_1 & & 0 & 0 \\ 0 & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & & \sigma_m & \omega_m \\ 0 & 0 & & -\omega_m & \sigma_m \end{pmatrix} x.$$

In this case, the eigenvalues can be shown to be $\lambda_j = \sigma_j \pm i\omega_j$. We once again can separate the state trajectories into independent solutions for each pair of states, and the solutions are of the form

$$\begin{aligned} x_{2j-1}(t) &= e^{\sigma_j t}\left(x_{2j-1}(0)\cos\omega_j t + x_{2j}(0)\sin\omega_j t\right), \\ x_{2j}(t) &= e^{\sigma_j t}\left(-x_{2j-1}(0)\sin\omega_j t + x_{2j}(0)\cos\omega_j t\right), \end{aligned}$$

where $j = 1, 2, \ldots, m$. We see that this system is asymptotically stable if and only if $\sigma_j = \operatorname{Re}\lambda_j < 0$. It is also possible to combine real and complex eigenvalues in (block) diagonal form, resulting in a mixture of solutions of the two types.

Very few systems are in one of the diagonal forms above, but many systems can be transformed into these forms via coordinate transformations. One such class of systems is those for which the dynamics matrix has distinct (nonrepeating) eigenvalues. In this case there is a matrix $T \in \mathbb{R}^{n\times n}$ such that the matrix TAT^{-1} is in (block) diagonal form, with the block diagonal elements corresponding to the eigenvalues of the original matrix A (see Exercise 5.14). If we choose new coordinates $z = Tx$, then

$$\frac{dz}{dt} = T\dot{x} = TAx = TAT^{-1}z$$

and the linear system has a (block) diagonal dynamics matrix. Furthermore, the eigenvalues of the transformed system are the same as those of the original system since if v is an eigenvector of A, then $w = Tv$ can be shown to be an eigenvector of TAT^{-1}. We can reason about the stability of the original system by noting that $x(t) = T^{-1}z(t)$, and so if the transformed system is stable (or asymptotically

stable), then the original system has the same type of stability.

This analysis shows that for linear systems with distinct eigenvalues, the stability of the system can be completely determined by examining the real part of the eigenvalues of the dynamics matrix. For more general systems, we make use of the following theorem, proved in the next chapter:

Theorem 5.1 (Stability of a linear system)**.** *The system*

$$\frac{dx}{dt} = Ax$$

is asymptotically stable if and only if all eigenvalues of A all have a strictly negative real part and is unstable if any eigenvalue of A has a strictly positive real part.

Note that it is not enough to have eigenvalues with $\mathrm{Re}(\lambda) \leq 0$. As a simple example, consider the system $\ddot{q} = 0$, which can be written in state space form as

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

The system has eigenvalues $\lambda = 0$ but the solutions are not bounded since we have

$$x_1(t) = x_{1,0} + x_{2,0}t, \quad x_2(t) = x_{2,0}.$$

Example 5.6 Compartment model
Consider the two-compartment module for drug delivery described in Section 4.6. Using concentrations as state variables and denoting the state vector by x, the system dynamics are given by

$$\frac{dx}{dt} = \begin{pmatrix} -k_0 - k_1 & k_1 \\ k_2 & -k_2 \end{pmatrix} x + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 0 & 1 \end{pmatrix} x,$$

where the input u is the rate of injection of a drug into compartment 1 and the concentration of the drug in compartment 2 is the measured output y. We wish to design a feedback control law that maintains a constant output given by $y = y_d$.

We choose an output feedback control law of the form

$$u = -k(y - y_d) + u_d,$$

where u_d is the rate of injection required to maintain the desired concentration and k is a feedback gain that should be chosen such that the closed loop system is stable. Substituting the control law into the system, we obtain

$$\begin{aligned} \frac{dx}{dt} &= \begin{pmatrix} -k_0 - k_1 & k_1 - b_0 k \\ k_2 & -k_2 \end{pmatrix} x + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} (u_d + k y_d) =: Ax + Bu_e, \\ y &= \begin{pmatrix} 0 & 1 \end{pmatrix} x =: Cx. \end{aligned}$$

The equilibrium concentration $x_e \in \mathbb{R}^2$ is given by $x_e = -A^{-1}Bu_e$ and

$$y_e = -CA^{-1}Bu_e = \frac{b_0 k_2}{k_0 k_2 + b_0 k_2 k}(u_d + k y_d).$$

Choosing u_d such that $y_e = y_d$ provides the constant rate of injection required to maintain the desired output. We can now shift coordinates to place the equilibrium point at the origin, which yields (after some algebra)

$$\frac{dz}{dt} = \begin{pmatrix} -k_0 - k_1 & k_1 - b_0 k \\ k_2 & -k_2 \end{pmatrix} z,$$

where $z = x - x_e$. We can now apply the results of Theorem 5.1 to determine the stability of the system. The eigenvalues of the system are given by the roots of the characteristic polynomial

$$\lambda(s) = s^2 + (k_0 + k_1 + k_2)s + (k_0 k_2 + b_0 k_2 k).$$

While the specific form of the roots is messy, it can be shown that the roots have negative real part as long as the linear term and the constant term are both positive (see Section 2.1, page 2-5). Hence the system is stable for any $k > 0$. ∇

Stability Analysis via Linear Approximation

An important feature of differential equations is that it is often possible to determine the local stability of an equilibrium point by approximating the system by a linear system. The following example illustrates the basic idea.

Example 5.7 Inverted pendulum
Consider again an inverted pendulum whose open loop dynamics are given by

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ \sin x_1 - \gamma x_2 \end{pmatrix},$$

where we have defined the state as $x = (\theta, \dot{\theta})$. We first consider the equilibrium point at $x = (0,0)$, corresponding to the straight-up position. If we assume that the angle $\theta = x_1$ remains small, then we can replace $\sin x_1$ with x_1 and $\cos x_1$ with 1, which gives the approximate system

$$\frac{dx}{dt} = \begin{pmatrix} x_2 \\ x_1 - \gamma x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -\gamma \end{pmatrix} x. \tag{5.9}$$

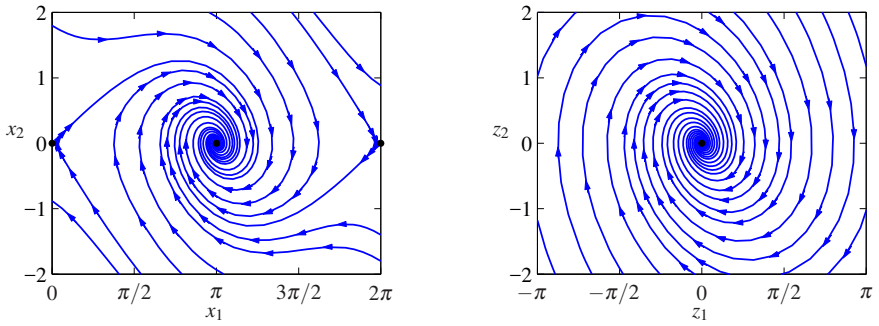
Intuitively, this system should behave similarly to the more complicated model as long as x_1 is small. In particular, it can be verified that the equilibrium point $(0,0)$ is unstable by plotting the phase portrait or computing the eigenvalues of the dynamics matrix in equation (5.9)

We can also approximate the system around the stable equilibrium point at $x = (\pi, 0)$. In this case we have to expand $\sin x_1$ and $\cos x_1$ around $x_1 = \pi$, according to the expansions

$$\sin(\pi + \theta) = -\sin\theta \approx -\theta, \qquad \cos(\pi + \theta) = -\cos(\theta) \approx -1.$$

If we define $z_1 = x_1 - \pi$ and $z_2 = x_2$, the resulting approximate dynamics are given by

$$\frac{dz}{dt} = \begin{pmatrix} z_2 \\ -z_1 - \gamma z_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & -\gamma \end{pmatrix} z. \tag{5.10}$$



(a) Nonlinear model (b) Linear approximation

Figure 5.11: Comparison between the phase portraits for the full nonlinear systems (a) and its linear approximation around the origin (b). Notice that near the equilibrium point at the center of the plots, the phase portraits (and hence the dynamics) are almost identical.

Figure 5.11 shows the phase portraits for the original system and the approximate system around the corresponding equilibrium points. Note that $z = (0,0)$ is the equilibrium point for this system and that it has the same basic form as the dynamics shown in Figure 5.8. The solutions for the original system and the approximate are very similar, although not exactly the same. It can be shown that if a linear approximation has either asymptotically stable or unstable equilibrium points, then the local stability of the original system must be the same (see Theorem 5.3 on page 5-23 for the case of asymptotic stability). ∇

More generally, suppose that we have a nonlinear system

$$\frac{dx}{dt} = F(x)$$

that has an equilibrium point at x_e. Computing the Taylor series expansion of the vector field, we can write

$$\frac{dx}{dt} = F(x_e) + \left.\frac{\partial F}{\partial x}\right|_{x_e} (x - x_e) + \text{higher-order terms in } (x - x_e).$$

Since $F(x_e) = 0$, we can approximate the system by choosing a new state variable $z = x - x_e$ and writing

$$\frac{dz}{dt} = Az, \qquad \text{where} \quad A = \left.\frac{\partial F}{\partial x}\right|_{x_e}. \tag{5.11}$$

We call the system (5.11) the *linear approximation* of the original nonlinear system or the *linearization* at x_e.

The fact that a linear model can be used to study the behavior of a nonlinear system near an equilibrium point is a powerful one. Indeed, we can take this even further and use a local linear approximation of a nonlinear system to design a feedback law that keeps the system near its equilibrium point (design of dynamics). Thus, feedback can be used to make sure that solutions remain close to

the equilibrium point, which in turn ensures that the linear approximation used to stabilize it is valid.

Linear approximations can also be used to understand the stability of nonequilibrium solutions, as illustrated by the following example.

Example 5.8 Stability of limit cycles
Consider the system given by equation (5.6),

$$\frac{dx_1}{dt} = x_2 + x_1(1 - x_1^2 - x_2^2), \qquad \frac{dx_2}{dt} = -x_1 + x_2(1 - x_1^2 - x_2^2),$$

whose phase portrait is shown in Figure 5.5. The differential equation has a periodic solution

$$x_1(t) = x_1(0)\cos t + x_2(0)\sin t, \tag{5.12}$$

with $x_1^2(0) + x_2^2(0) = 1$.

To explore the stability of this solution, we introduce polar coordinates r and φ, which are related to the state variables x_1 and x_2 by

$$x_1 = r\cos\varphi, \qquad x_2 = r\sin\varphi.$$

Differentiation gives the following linear equations for $\dot{r}$ and $\dot{\varphi}$:

$$\dot{x}_1 = \dot{r}\cos\varphi - r\dot{\varphi}\sin\varphi, \qquad \dot{x}_2 = \dot{r}\sin\varphi + r\dot{\varphi}\cos\varphi.$$

Solving this linear system for $\dot{r}$ and $\dot{\varphi}$ gives, after some calculation,

$$\frac{dr}{dt} = r(1 - r^2), \qquad \frac{d\varphi}{dt} = -1. \tag{5.13}$$

Notice that the equations are decoupled; hence we can analyze the stability of each state separately.

The equation for r has three equilibria: $r = 0$, $r = 1$ and $r = -1$ (not realizable since r must be positive). We can analyze the stability of these equilibria by linearizing the radial dynamics with $F(r) = r(1 - r^2)$. The corresponding linear dynamics are given by

$$\frac{d\tilde{r}}{dt} = \left.\frac{\partial F}{\partial \tilde{r}}\right|_{r_e} \tilde{r} = (1 - 3r_e^2)\tilde{r}, \quad r_e = 0, 1,$$

where $\tilde{r} = r - r_e$ represents the deviation from the equilibrium point. It follows from the sign of $(1 - 3r_e^2)$ that the equilibrium $r = 0$ is unstable and the equilibrium $r = 1$ is asymptotically stable for equation (5.13). Thus for any initial condition $r > 0$ the solution goes to $r = 1$ as time goes to infinity, but if the system starts with $r = 0$, it will remain at the equilibrium for all times. This implies that all solutions to the original system that do not start at $x_1 = x_2 = 0$ will approach the circle $x_1^2 + x_2^2 = 1$ as time increases.

To show the stability of the full solution (5.12), we must investigate the behavior of neighboring solutions with different initial conditions. We have already shown that the radius r will approach that of the solution (5.12) as long as $r(0) > 0$. The equation for the angle φ can be integrated analytically to give $\varphi(t) = -t +$

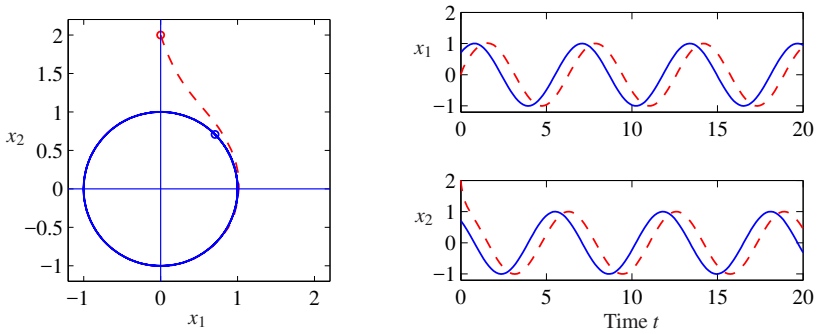


Figure 5.12: Solution curves for a stable limit cycle. The phase portrait on the left shows that the trajectory for the system rapidly converges to the stable limit cycle. The starting points for the trajectories are marked by circles in the phase portrait. The time domain plots on the right show that the states do not converge to the solution but instead maintain a constant phase error.

$\varphi(0)$, which shows that solutions starting at different angles φ will neither converge nor diverge. Thus, the unit circle is *attracting*, but the solution (5.12) is only stable, not asymptotically stable. The behavior of the system is illustrated by the simulation in Figure 5.12. Notice that the solutions approach the circle rapidly, but that there is a constant phase shift between the solutions. ∇

5.4 Lyapunov Stability Analysis

We now return to the study of the full nonlinear system

$$\frac{dx}{dt} = F(x), \quad x \in \mathbb{R}^n. \tag{5.14}$$

Having defined when a solution for a nonlinear dynamical system is stable, we can now ask how to prove that a given solution is stable, asymptotically stable or unstable. For physical systems, one can often argue about stability based on dissipation of energy. The generalization of that technique to arbitrary dynamical systems is based on the use of Lyapunov functions in place of energy.

In this section we will describe techniques for determining the stability of solutions for a nonlinear system (5.14). We will generally be interested in stability of equilibrium points, and it will be convenient to assume that $x_e = 0$ is the equilibrium point of interest. (If not, rewrite the equations in a new set of coordinates $z = x - x_e$.)

Lyapunov Functions

A *Lyapunov function* $V : \mathbb{R}^n \to \mathbb{R}$ is an energy-like function that can be used to determine the stability of a system. Roughly speaking, if we can find a nonnegative function that always decreases along trajectories of the system, we can conclude that the minimum of the function is a stable equilibrium point (locally).

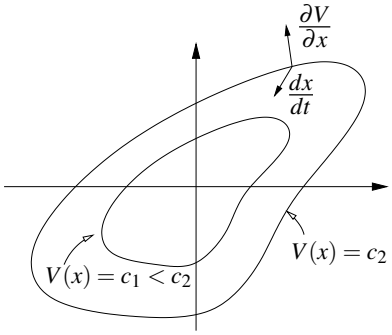


Figure 5.13: Geometric illustration of Lyapunov's stability theorem. The closed contours represent the level sets of the Lyapunov function $V(x) = c$. If dx/dt points inward to these sets at all points along the contour, then the trajectories of the system will always cause $V(x)$ to decrease along the trajectory.

To describe this more formally, we start with a few definitions. Let $B_r = B_r(0)$ be a ball of radius r around the origin. We say that a continuous function V is *positive definite* on B_r if $V(x) > 0$ for all $x \in B_r$, $x \neq 0$ and $V(0) = 0$. Similarly, a function is *negative definite* on B_r if $V(x) < 0$ for all $x \in B_r$, $x \neq 0$ and $V(0) = 0$. We say that a function V is *positive semidefinite* if $V(x) \geq 0$ for all $x \in B_r$, but $V(x)$ can be zero at points other than just $x = 0$.

To illustrate the difference between a positive definite function and a positive semidefinite function, suppose that $x \in \mathbb{R}^2$ and let

$$V_1(x) = x_1^2, \qquad V_2(x) = x_1^2 + x_2^2.$$

Both V_1 and V_2 are always nonnegative. However, it is possible for V_1 to be zero even if $x \neq 0$. Specifically, if we set $x = (0, c)$, where $c \in \mathbb{R}$ is any nonzero number, then $V_1(x) = 0$. On the other hand, $V_2(x) = 0$ if and only if $x = (0,0)$. Thus V_1 is positive semidefinite and V_2 is positive definite.

We can now characterize the stability of an equilibrium point $x_e = 0$ for the system (5.14).

Theorem 5.2 (Lyapunov stability theorem)**.** *Let V be a function on $\mathbb{R}^n$ and let $\dot{V}$ represent the time derivative of V along trajectories of the system dynamics* (5.14)*:*

$$\dot{V} = \frac{\partial V}{\partial x}\frac{dx}{dt} = \frac{\partial V}{\partial x}F(x).$$

If there exists $r > 0$ such that V is positive definite and $\dot{V}$ is negative semidefinite on B_r, then $x = 0$ is (locally) stable in the sense of Lyapunov. If V is positive definite and $\dot{V}$ is negative definite in B_r, then $x = 0$ is (locally) asymptotically stable.

If V satisfies one of the conditions above, we say that V is a (local) *Lyapunov function* for the system. These results have a nice geometric interpretation. The level curves for a positive definite function are the curves defined by $V(x) = c$, $c > 0$, and for each c this gives a closed contour, as shown in Figure 5.13. The condition that $\dot{V}(x)$ is negative simply means that the vector field points toward

lower-level contours. This means that the trajectories move to smaller and smaller values of V and if $\dot{V}$ is negative definite then x must approach 0.

Finding Lyapunov functions is not always easy. For example, consider the linear system

$$\frac{dx_1}{dt} = x_2, \qquad \frac{dx_2}{dt} = -\alpha x_2.$$

Example 5.9 Scalar nonlinear system
Consider the scalar nonlinear system

$$\frac{dx}{dt} = \frac{2}{1+x} - x.$$

This system has equilibrium points at $x = 1$ and $x = -2$. We consider the equilibrium point at $x = 1$ and rewrite the dynamics using $z = x - 1$:

$$\frac{dz}{dt} = \frac{2}{2+z} - z - 1,$$

which has an equilibrium point at $z = 0$. Now consider the candidate Lyapunov function

$$V(z) = \frac{1}{2}z^2,$$

which is globally positive definite. The derivative of V along trajectories of the system is given by

$$\dot{V}(z) = z\dot{z} = \frac{2z}{2+z} - z^2 - z.$$

If we restrict our analysis to an interval B_r, where $r < 2$, then $2 + z > 0$ and we can multiply through by $2 + z$ to obtain

$$2z - (z^2 + z)(2 + z) = -z^3 - 3z^2 = -z^2(z + 3) < 0, \qquad z \in B_r,\ r < 2.$$

It follows that $\dot{V}(z) < 0$ for all $z \in B_r$, $z \neq 0$, and hence the equilibrium point $x = 1$ is locally asymptotically stable. ∇

A slightly more complicated situation occurs if $\dot{V}$ is negative semidefinite. In this case it is possible that $\dot{V}(x) = 0$ when $x \neq 0$, and hence x could stop decreasing in value. The following example illustrates this case.

Example 5.10 Hanging pendulum
A normalized model for a hanging pendulum is

$$\frac{dx_1}{dt} = x_2, \qquad \frac{dx_2}{dt} = -\sin x_1,$$

where x_1 is the angle between the pendulum and the vertical, with positive x_1 corresponding to counterclockwise rotation. The equation has an equilibrium $x_1 = x_2 = 0$, which corresponds to the pendulum hanging straight down. To explore the stability of this equilibrium we choose the total energy as a Lyapunov function:

$$V(x) = 1 - \cos x_1 + \frac{1}{2}x_2^2 \approx \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2.$$

The Taylor series approximation shows that the function is positive definite for small x. The time derivative of $V(x)$ is

$$\dot{V} = \dot{x}_1 \sin x_1 + \dot{x}_2 x_2 = x_2 \sin x_1 - x_2 \sin x_1 = 0.$$

Since this function is negative semidefinite, it follows from Lyapunov's theorem that the equilibrium is stable but not necessarily asymptotically stable. When perturbed, the pendulum actually moves in a trajectory that corresponds to constant energy. ∇

Lyapunov functions are not always easy to find, and they are not unique. In many cases energy functions can be used as a starting point, as was done in Example 5.10. It turns out that Lyapunov functions can always be found for any stable system (under certain conditions), and hence one knows that if a system is stable, a Lyapunov function exists (and vice versa). Recent results using sum-of-squares methods have provided systematic approaches for finding Lyapunov systems [PPP02]. Sum-of-squares techniques can be applied to a broad variety of systems, including systems whose dynamics are described by polynomial equations, as well as hybrid systems, which can have different models for different regions of state space.

For a linear dynamical system of the form

$$\frac{dx}{dt} = Ax,$$

it is possible to construct Lyapunov functions in a systematic manner. To do so, we consider quadratic functions of the form

$$V(x) = x^T P x,$$

where $P \in \mathbb{R}^{n \times n}$ is a symmetric matrix ($P = P^T$). The condition that V be positive definite on B_r for some $r > 0$ is equivalent to the condition that P be a *positive definite matrix*:

$$x^T P x > 0, \quad \text{for all } x \neq 0,$$

which we write as $P \succ 0$. It can be shown that if P is symmetric, then P is positive definite if and only if all of its eigenvalues are real and positive.

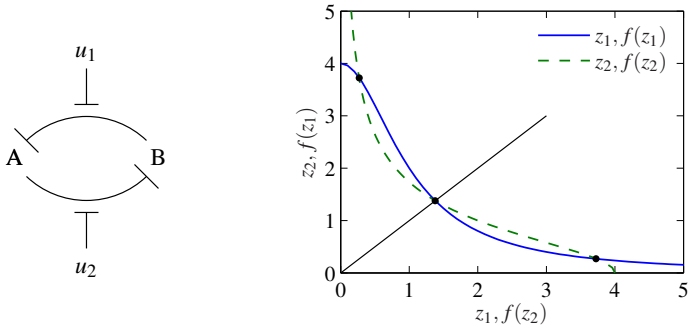
Given a candidate Lyapunov function $V(x) = x^T P x$, we can now compute its derivative along flows of the system:

$$\dot{V} = \frac{\partial V}{\partial x}\frac{dx}{dt} = x^T (A^T P + PA) x =: -x^T Q x.$$

The requirement that $\dot{V}$ be negative definite on B_r (for asymptotic stability) becomes a condition that the matrix Q be positive definite. Thus, to find a Lyapunov function for a linear system it is sufficient to choose a $Q \succ 0$ and solve the *Lyapunov equation*:

$$A^T P + PA = -Q. \tag{5.15}$$

This is a linear equation in the entries of P, and hence it can be solved using linear algebra. It can be shown that the equation always has a solution if all of



(a) Circuit diagram (b) Equilibrium points

Figure 5.14: Stability of a genetic switch. The circuit diagram in (a) represents two proteins that are each repressing the production of the other. The inputs u_1 and u_2 interfere with this repression, allowing the circuit dynamics to be modified. The equilibrium points for this circuit can be determined by the intersection of the two curves shown in (b).

the eigenvalues of the matrix A are in the left half-plane. Moreover, the solution P is positive definite if Q is positive definite. It is thus always possible to find a quadratic Lyapunov function for a stable linear system. We will defer a proof of this until Chapter 6, where more tools for analysis of linear systems will be developed.

Knowing that we have a direct method to find Lyapunov functions for linear systems, we can now investigate the stability of nonlinear systems. Consider the system

$$\frac{dx}{dt} = F(x) =: Ax + \tilde{F}(x), \tag{5.16}$$

where $F(0) = 0$ and $\tilde{F}(x)$ contains terms that are second order and higher in the elements of x. The function Ax is an approximation of $F(x)$ near the origin, and we can determine the Lyapunov function for the linear approximation and investigate if it is also a Lyapunov function for the full nonlinear system. The following example illustrates the approach.

Example 5.11 Genetic switch

Consider the dynamics of a set of repressors connected together in a cycle, as shown in Figure 5.14a. The normalized dynamics for this system were given in Exercise 3.9:

$$\frac{dz_1}{d\tau} = \frac{\mu}{1+z_2^n} - z_1, \qquad \frac{dz_2}{d\tau} = \frac{\mu}{1+z_1^n} - z_2, \tag{5.17}$$

where z_1 and z_2 are scaled versions of the protein concentrations, $n > 0$ and $\mu > 0$ are parameters that describe the interconnection between the genes and we have set the external inputs u_1 and u_2 to zero.

The equilibrium points for the system are found by equating the time deriva-

tives to zero. We define

$$f(u) = \frac{\mu}{1+u^n}, \qquad f'(u) = \frac{df}{du} = \frac{-\mu n u^{n-1}}{(1+u^n)^2},$$

so that our dynamics become

$$\frac{dz_1}{d\tau} = f(z_2) - z_1, \qquad \frac{dz_2}{d\tau} = f(z_1) - z_2,$$

and the equilibrium points are defined as the solutions of the equations

$$z_1 = f(z_2), \qquad z_2 = f(z_1).$$

If we plot the curves $(z_1, f(z_1))$ and $(f(z_2), z_2)$ on a graph, then these equations will have a solution when the curves intersect, as shown in Figure 5.14b. Because of the shape of the curves, it can be shown that there will always be three solutions: one at $z_{1e} = z_{2e}$, one with $z_{1e} < z_{2e}$ and one with $z_{1e} > z_{2e}$. If $\mu \gg 1$, then we can show that the solutions are given approximately by

$$z_{1e} \approx \mu, \quad z_{2e} \approx \frac{1}{\mu^{n-1}}; \qquad z_{1e} = z_{2e}; \qquad z_{1e} \approx \frac{1}{\mu^{n-1}}, \quad z_{2e} \approx \mu. \tag{5.18}$$

To check the stability of the system, we write $f(u)$ in terms of its Taylor series expansion about u_e:

$$f(u) = f(u_e) + f'(u_e)\cdot(u-u_e) + \frac{1}{2}f''(u_e)\cdot(u-u_e)^2 + \text{higher-order terms},$$

where f' represents the first derivative of the function, and f'' the second. Using these approximations, the dynamics can then be written as

$$\frac{dw}{dt} = \begin{pmatrix} -1 & f'(z_{2e}) \\ f'(z_{1e}) & -1 \end{pmatrix} w + \tilde{F}(w),$$

where $w = z - z_e$ is the shifted state and $\tilde{F}(w)$ represents quadratic and higher-order terms.

We now use equation (5.15) to search for a Lyapunov function. Choosing $Q = I$ and letting $P \in \mathbb{R}^{2\times 2}$ have elements p_{ij}, we search for a solution of the equation

$$\begin{pmatrix} -1 & f_1' \\ f_2' & -1 \end{pmatrix} \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} + \begin{pmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{pmatrix} \begin{pmatrix} -1 & f_2' \\ f_1' & -1 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},$$

where $f_1' = f'(z_{1e})$ and $f_2' = f'(z_{2e})$. Note that we have set $p_{21} = p_{12}$ to force P to be symmetric. Multiplying out the matrices, we obtain

$$\begin{pmatrix} -2p_{11} + 2f_1'p_{12} & p_{11}f_2' - 2p_{12} + p_{22}f_1' \\ p_{11}f_2' - 2p_{12} + p_{22}f_1' & -2p_{22} + 2f_2'p_{12} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},$$

which is a set of *linear* equations for the unknowns p_{ij}. We can solve these linear equations to obtain

$$p_{11} = -\frac{f_1'^2 - f_2'f_1' + 2}{4(f_1'f_2' - 1)}, \qquad p_{12} = -\frac{f_1' + f_2'}{4(f_1'f_2' - 1)}, \qquad p_{22} = -\frac{f_2'^2 - f_1'f_2' + 2}{4(f_1'f_2' - 1)}.$$

To check that $V(w) = w^T P w$ is a Lyapunov function, we must verify that $V(w)$ is positive definite function or equivalently that $P \succ 0$. Since P is a 2×2 symmetric matrix, it has two real eigenvalues λ_1 and λ_2 that satisfy

$$\lambda_1 + \lambda_2 = \text{trace}(P), \qquad \lambda_1 \cdot \lambda_2 = \det(P).$$

In order for P to be positive definite λ_1 and λ_2 must be positive, and we thus require that

$$\text{trace}(P) = \frac{f_1'^2 - 2f_2'f_1' + f_2'^2 + 4}{4 - 4f_1'f_2'} > 0, \quad \det(P) = \frac{f_1'^2 - 2f_2'f_1' + f_2'^2 + 4}{16 - 16f_1'f_2'} > 0.$$

We see that $\text{trace}(P) = 4\det(P)$ and the numerator of the expressions is just $(f_1 - f_2)^2 + 4 > 0$, so it suffices to check the sign of $1 - f_1'f_2'$. In particular, for P to be positive definite, we require that

$$f'(z_{1e})f'(z_{2e}) < 1.$$

We can now make use of the expressions for f' defined earlier and evaluate at the approximate locations of the equilibrium points derived in equation (5.18). For the equilibrium points where $z_{1e} \neq z_{2e}$, we can show that

$$f'(z_{1e})f'(z_{2e}) \approx f'(\mu)f'(\frac{1}{\mu^{n-1}}) = \frac{-\mu n \mu^{n-1}}{(1+\mu^n)^2} \cdot \frac{-\mu n \mu^{-(n-1)^2}}{(1+\mu^{-n(n-1)})^2} \approx n^2 \mu^{-n^2+n}.$$

Using $n = 2$ and $\mu \approx 200$ from Exercise 3.9, we see that $f'(z_{1e})f'(z_{2e}) \ll 1$ and hence P is a positive definite. This implies that V is a positive definite function and hence a potential Lyapunov function for the system.

To determine if the equilibria $z_{1e} \neq z_{2e}$ are stable for the system (5.17), we now compute $\dot{V}$ at the equilibrium point. By construction,

$$\begin{aligned} \dot{V} &= w^T(PA + A^T P)w + \tilde{F}^T(w)Pw + w^T P\tilde{F}(w) \\ &= -w^T w + \tilde{F}^T(w)Pw + w^T P\tilde{F}(w). \end{aligned}$$

Since all terms in $\tilde{F}$ are quadratic or higher order in w, it follows that $\tilde{F}^T(w)Pw$ and $w^T P\tilde{F}(w)$ consist of terms that are at least third order in w. Therefore if w is sufficiently close to zero, then the cubic and higher-order terms will be smaller than the quadratic terms. Hence, sufficiently close to $w = 0$, $\dot{V}$ is negative definite, allowing us to conclude that these equilibrium points are both stable.

Figure 5.15 shows the phase portrait and time traces for a system with $\mu = 4$, illustrating the bistable nature of the system. When the initial condition starts with a concentration of protein B greater than that of A, the solution converges to the equilibrium point at (approximately) $(1/\mu^{n-1}, \mu)$. If A is greater than B, then it goes to $(\mu, 1/\mu^{n-1})$. The equilibrium point with $z_{1e} = z_{2e}$ is unstable. ∇

More generally, we can investigate what the linear approximation tells about the stability of a solution to a nonlinear equation. The following theorem gives a partial answer for the case of stability of an equilibrium point.

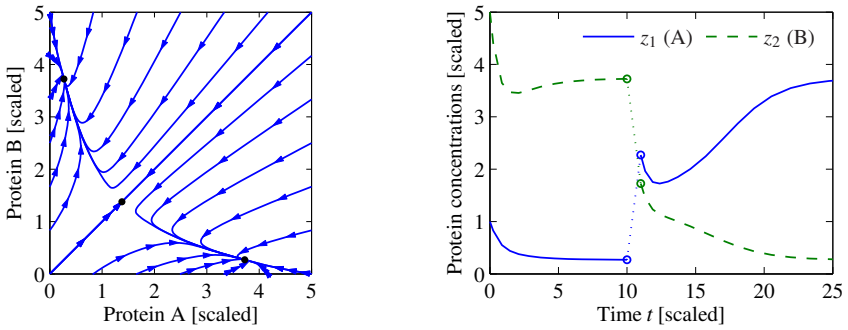


Figure 5.15: Dynamics of a genetic switch. The phase portrait on the left shows that the switch has three equilibrium points, corresponding to protein A having a concentration greater than, equal to or less than protein B. The equilibrium point with equal protein concentrations is unstable, but the other equilibrium points are stable. The simulation on the right shows the time response of the system starting from two different initial conditions. The initial portion of the curve corresponds to initial concentrations $z(0) = (1,5)$ and converges to the equilibrium where $z_{1e} < z_{2e}$. At time $t = 10$, the concentrations are perturbed by $+2$ in z_1 and -2 in z_2, moving the state into the region of the state space whose solutions converge to the equilibrium point where $z_{2e} < z_{1e}$.

Theorem 5.3. *Consider the dynamical system* (5.16) *with* $F(0) = 0$ *and* $\tilde{F}$ *such that* $\lim \|\tilde{F}(x)\|/\|x\| \to 0$ *as* $\|x\| \to 0$. *If the real parts of all eigenvalues of A are strictly less than zero, then* $x_e = 0$ *is a locally asymptotically stable equilibrium point of equation* (5.16).

This theorem implies that asymptotic stability of the linear approximation implies *local* asymptotic stability of the original nonlinear system. The theorem is very important for control because it implies that stabilization of a linear approximation of a nonlinear system results in a stable equilibrium for the nonlinear system. The proof of this theorem follows the technique used in Example 5.11. A formal proof can be found in [Kha01].

It can also be shown that if A has one or more eigenvalues with strictly positive real part, then $x_e = 0$ is an unstable equilibrium for the nonlinear system.

Krasovski–Lasalle Invariance Principle

For general nonlinear systems, especially those in symbolic form, it can be difficult to find a positive definite function V whose derivative is strictly negative definite. The Krasovski–Lasalle theorem enables us to conclude the asymptotic stability of an equilibrium point under less restrictive conditions, namely, in the case where $\dot{V}$ is negative semidefinite, which is often easier to construct. It only applies only to time-invariant or periodic systems, which are the cases we consider here. This section makes use of some additional concepts from dynamical systems; see Hahn [Hah67] or Khalil [Kha01] for a more detailed description.

We will deal with the time-invariant case and begin by introducing a few more

definitions. We denote the solution trajectories of the time-invariant system

$$\frac{dx}{dt} = F(x) \tag{5.19}$$

as $x(t;a)$, which is the solution of equation (5.19) at time t starting from a at $t_0 = 0$. The ω *limit set* of a trajectory $x(t;a)$ is the set of all points $z \in \mathbb{R}^n$ such that there exists a strictly increasing sequence of times t_n such that $x(t_n;a) \to z$ as $n \to \infty$. A set $M \subset \mathbb{R}^n$ is said to be an *invariant set* if for all $b \in M$, we have $x(t;b) \in M$ for all $t \geq 0$. It can be proved that the ω limit set of every trajectory is closed and invariant. We may now state the Krasovski–Lasalle principle.

Theorem 5.4 (Krasovski–Lasalle principle)**.** *Let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally positive definite function such that on the compact set $\Omega_r = \{x \in \mathbb{R}^n : V(x) \leq r\}$ we have $\dot{V}(x) \leq 0$. Define*

$$S = \{x \in \Omega_r : \dot{V}(x) = 0\}.$$

As $t \to \infty$, the trajectory tends to the largest invariant set inside S; i.e., its ω limit set is contained inside the largest invariant set in S. In particular, if S contains no invariant sets other than $x = 0$, then 0 is asymptotically stable.

Proofs are given in [Kra63] and [LaS60].

Lyapunov functions can often be used to design stabilizing controllers, as is illustrated by the following example, which also illustrates how the Krasovski–Lasalle principle can be applied.

Example 5.12 Inverted pendulum

Following the analysis in Example 3.9, an inverted pendulum can be described by the following normalized model:

$$\frac{dx_1}{dt} = x_2, \qquad \frac{dx_2}{dt} = \sin x_1 + u\cos x_1, \tag{5.20}$$

where x_1 is the angular deviation from the upright position and u is the (scaled) acceleration of the pivot, as shown in Figure 5.16a. The system has an equilibrium at $x_1 = x_2 = 0$, which corresponds to the pendulum standing upright. This equilibrium is unstable.

To find a stabilizing controller we consider the following candidate for a Lyapunov function:

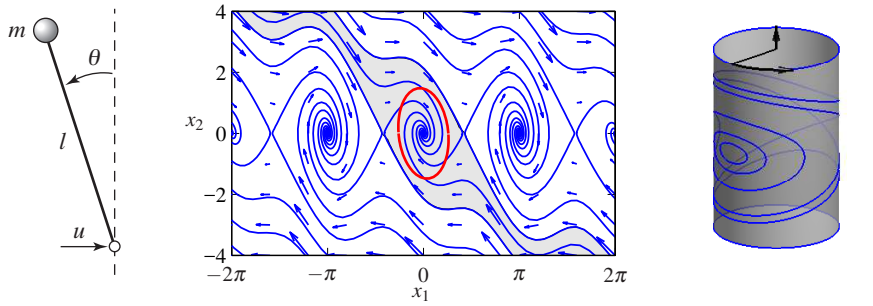
$$V(x) = (\cos x_1 - 1) + a(1 - \cos^2 x_1) + \frac{1}{2}x_2^2 \approx \big(a - \frac{1}{2}\big)x_1^2 + \frac{1}{2}x_2^2.$$

The Taylor series expansion shows that the function is positive definite near the origin if $a > 0.5$. The time derivative of $V(x)$ is

$$\dot{V} = -\dot{x}_1 \sin x_1 + 2a\dot{x}_1 \sin x_1 \cos x_1 + \dot{x}_2 x_2 = x_2(u + 2a\sin x_1)\cos x_1.$$

Choosing the feedback law

$$u = -2a\sin x_1 - x_2 \cos x_1$$



(a) Physical system (b) Phase portrait (c) Manifold view

Figure 5.16: Stabilized inverted pendulum. A control law applies a force u at the bottom of the pendulum to stabilize the inverted position (a). The phase portrait (b) shows that the equilibrium point corresponding to the vertical position is stabilized. The shaded region indicates the set of initial conditions that converge to the origin. The ellipse corresponds to a level set of a Lyapunov function $V(x)$ for which $V(x) > 0$ and $\dot{V}(x) < 0$ for all points inside the ellipse. This can be used as an estimate of the region of attraction of the equilibrium point. The actual dynamics of the system evolve on a manifold (c).

gives

$$\dot{V} = -x_2^2 \cos^2 x_1.$$

It follows from Lyapunov's theorem that the equilibrium is (locally) stable. However, since the function is only negative semidefinite, we cannot conclude asymptotic stability using Theorem 5.2. However, note that $\dot{V} = 0$ implies that $x_2 = 0$ or $x_1 = \pi/2 \pm n\pi$.

If we restrict our analysis to a small neighborhood of the origin Ω_r, $r \ll \pi/2$, then we can define

$$S = \{(x_1, x_2) \in \Omega_r : x_2 = 0\}$$

and we can compute the largest invariant set inside S. For a trajectory to remain in this set we must have $x_2 = 0$ for all t and hence $\dot{x}_2(t) = 0$ as well. Using the dynamics of the system (5.20), we see that $x_2(t) = 0$ and $\dot{x}_2(t) = 0$ implies $x_1(t) = 0$ as well. Hence the largest invariant set inside S is $(x_1, x_2) = 0$, and we can use the Krasovski–Lasalle principle to conclude that the origin is locally asymptotically stable. A phase portrait of the closed loop system is shown in Figure 5.16b.

In the analysis and the phase portrait, we have treated the angle of the pendulum $\theta = x_1$ as a real number. In fact, θ is an angle with $\theta = 2\pi$ equivalent to $\theta = 0$. Hence the dynamics of the system actually evolves on a *manifold* (smooth surface) as shown in Figure 5.16c. Analysis of nonlinear dynamical systems on manifolds is more complicated, but uses many of the same basic ideas presented here. ∇

5.5 Parametric and Nonlocal Behavior

Most of the tools that we have explored are focused on the local behavior of a fixed system near an equilibrium point. In this section we briefly introduce some concepts regarding the global behavior of nonlinear systems and the dependence of a system's behavior on parameters in the system model.

Regions of Attraction

To get some insight into the behavior of a nonlinear system we can start by finding the equilibrium points. We can then proceed to analyze the local behavior around the equilibria. The behavior of a system near an equilibrium point is called the *local* behavior of the system.

The solutions of the system can be very different far away from an equilibrium point. This is seen, for example, in the stabilized pendulum in Example 5.12. The inverted equilibrium point is stable, with small oscillations that eventually converge to the origin. But far away from this equilibrium point there are trajectories that converge to other equilibrium points or even cases in which the pendulum swings around the top multiple times, giving very long oscillations that are topologically different from those near the origin.

To better understand the dynamics of the system, we can examine the set of all initial conditions that converge to a given asymptotically stable equilibrium point. This set is called the *region of attraction* for the equilibrium point. An example is shown by the shaded region of the phase portrait in Figure 5.16b. In general, computing regions of attraction is difficult. However, even if we cannot determine the region of attraction, we can often obtain patches around the stable equilibria that are attracting. This gives partial information about the behavior of the system.

One method for approximating the region of attraction is through the use of Lyapunov functions. Suppose that V is a local Lyapunov function for a system around an equilibrium point x_0. Let Ω_r be a set on which $V(x)$ has a value less than r,

$$\Omega_r = \{x \in \mathbb{R}^n : V(x) \leq r\},$$

and suppose that $\dot{V}(x) \leq 0$ for all $x \in \Omega_r$, with equality only at the equilibrium point x_0. Then Ω_r is inside the region of attraction of the equilibrium point. Since this approximation depends on the Lyapunov function and the choice of Lyapunov function is not unique, it can sometimes be a very conservative estimate.

It is sometimes the case that we can find a Lyapunov function V such that V is positive definite and $\dot{V}$ is negative (semi-) definite for all $x \in \mathbb{R}^n$. In many instances it can then be shown that the region of attraction for the equilibrium point is the entire state space, and the equilibrium point is *globally* asymptotically stable. More detailed conditions for global stability can be found in [Kha01].

Example 5.13 Stabilized inverted pendulum
Consider again the stabilized inverted pendulum from Example 5.12. The Lya-

punov function for the system was

$$V(x) = (\cos x_1 - 1) + a(1 - \cos^2 x_1) + \frac{1}{2}x_2^2.$$

With $a > 0.5$, $\dot{V}$ was negative semidefinite for all x and nonzero when $x_1 \neq \pm\pi/2$. Hence any x such that $|x_1| < \pi/2$ and $V(x) > 0$ will be inside the invariant set defined by the level curves of $V(x)$. One of these level sets is shown in Figure 5.16b. ∇

Bifurcations

Another important property of nonlinear systems is how their behavior changes as the parameters governing the dynamics change. We can study this in the context of models by exploring how the location of equilibrium points, their stability, their regions of attraction and other dynamic phenomena, such as limit cycles, vary based on the values of the parameters in the model.

Consider a differential equation of the form

$$\frac{dx}{dt} = F(x, \mu), \quad x \in \mathbb{R}^n, \mu \in \mathbb{R}^k, \tag{5.21}$$

where x is the state and μ is a set of parameters that describe the family of equations. The equilibrium solutions satisfy

$$F(x, \mu) = 0,$$

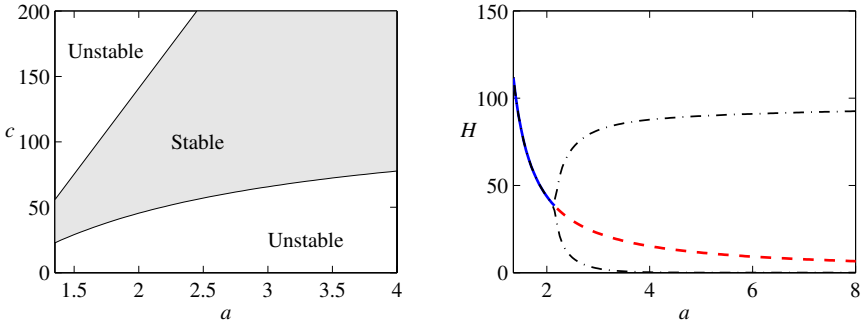
and as μ is varied, the corresponding solutions $x_e(\mu)$ can also vary. We say that the system (5.21) has a *bifurcation* at $\mu = \mu^*$ if the behavior of the system changes qualitatively at μ^*. This can occur either because of a change in stability type or a change in the number of solutions at a given value of μ.

Example 5.14 Predator–prey
Consider the predator–prey system described in Example 3.3 and modeled as a continuous time system as described in Section 4.7. The dynamics of the system are given by

$$\frac{dH}{dt} = rH\left(1 - \frac{H}{k}\right) - \frac{aHL}{c+H}, \qquad \frac{dL}{dt} = b\frac{aHL}{c+H} - dL, \tag{5.22}$$

where H and L are the numbers of hares (prey) and lynxes (predators) and a, b, c, d, k and r are parameters that model a given predator–prey system (described in more detail in Section 4.7). The system has an equilibrium point at $H_e > 0$ and $L_e > 0$ that can be found numerically.

To explore how the parameters of the model affect the behavior of the system, we choose to focus on two specific parameters of interest: a, the interaction coefficient between the populations and c, a parameter affecting the prey consumption rate. Figure 5.17a is a numerically computed *parametric stability diagram* showing the regions in the chosen parameter space for which the equilibrium point is



(a) Stability diagram (b) Bifurcation diagram

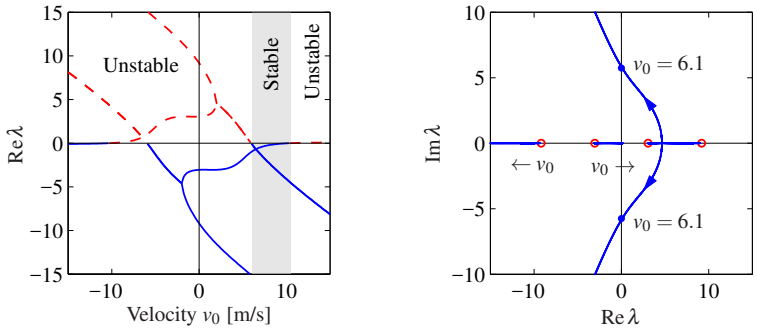
Figure 5.17: Bifurcation analysis of the predator–prey system. (a) Parametric stability diagram showing the regions in parameter space for which the system is stable. (b) Bifurcation diagram showing the location and stability of the equilibrium point as a function of a. The solid line represents a stable equilibrium point, and the dashed line represents an unstable equilibrium point. The dashed-dotted lines indicate the upper and lower bounds for the limit cycle at that parameter value (computed via simulation). The nominal values of the parameters in the model are $a = 3.2$, $b = 0.6$, $c = 50$, $d = 0.56$, $k = 125$ and $r = 1.6$.

stable (leaving the other parameters at their nominal values). We see from this figure that for certain combinations of a and c we get a stable equilibrium point, while at other values this equilibrium point is unstable.

Figure 5.17b is a numerically computed *bifurcation diagram* for the system. In this plot, we choose one parameter to vary (a) and then plot the equilibrium value of one of the states (H) on the vertical axis. The remaining parameters are set to their nominal values. A solid line indicates that the equilibrium point is stable; a dashed line indicates that the equilibrium point is unstable. Note that the stability in the bifurcation diagram matches that in the parametric stability diagram for $c = 50$ (the nominal value) and a varying from 1.35 to 4. For the predator–prey system, when the equilibrium point is unstable, the solution converges to a stable limit cycle. The amplitude of this limit cycle is shown by the dashed-dotted line in Figure 5.17b. ∇

A particular form of bifurcation that is very common when controlling linear systems is that the equilibrium remains fixed but the stability of the equilibrium changes as the parameters are varied. In such a case it is revealing to plot the eigenvalues of the system as a function of the parameters. Such plots are called *root locus diagrams* because they give the locus of the eigenvalues when parameters change. Bifurcations occur when parameter values are such that there are eigenvalues with zero real part. Computing environments such LabVIEW, MATLAB and Mathematica have tools for plotting root loci. A more detailed discussion of the root locus is given in Section 12.5.

Example 5.15 Root locus diagram for a bicycle model
Consider the linear bicycle model given by equation (4.7) in Section 4.2. Introduc-



(a) Stability diagram (b) Root locus diagram

Figure 5.18: Stability plots for a bicycle moving at constant velocity. The plot in (a) shows the real part of the system eigenvalues as a function of the bicycle velocity v_0. The system is stable when all eigenvalues have negative real part (shaded region). The plot in (b) shows the locus of eigenvalues on the complex plane as the velocity v is varied and gives a different view of the stability of the system. This type of plot is called a *root locus diagram*.

ing the state variables $x_1 = \varphi$, $x_2 = \delta$, $x_3 = \dot{\varphi}$ and $x_4 = \dot{\delta}$ and setting the steering torque $T = 0$, the equations can be written as

$$\frac{dx}{dt} = \begin{pmatrix} 0 & I \\ -M^{-1}(K_0 + K_2 v_0^2) & -M^{-1} C v_0 \end{pmatrix} x =: Ax,$$

where I is a 2×2 identity matrix and v_0 is the velocity of the bicycle. Figure 5.18a shows the real parts of the eigenvalues as a function of velocity. Figure 5.18b shows the dependence of the eigenvalues of A on the velocity v_0. The figures show that the bicycle is unstable for low velocities because two eigenvalues are in the right half-plane. As the velocity increases, these eigenvalues move into the left half-plane, indicating that the bicycle becomes self-stabilizing. As the velocity is increased further, there is an eigenvalue close to the origin that moves into the right half-plane, making the bicycle unstable again. However, this eigenvalue is small and so it can easily be stabilized by a rider. Figure 5.18a shows that the bicycle is self-stabilizing for velocities between 6 and 10 m/s. ∇

Parametric stability diagrams and bifurcation diagrams can provide valuable insights into the dynamics of a nonlinear system. It is usually necessary to carefully choose the parameters that one plots, including combining the natural parameters of the system to eliminate extra parameters when possible. Computer programs such as `AUTO`, `LOCBIF` and `XPPAUT` provide numerical algorithms for producing stability and bifurcation diagrams.

Design of Nonlinear Dynamics Using Feedback

In most of the text we will rely on linear approximations to design feedback laws that stabilize an equilibrium point and provide a desired level of performance. However, for some classes of problems the feedback controller must be nonlinear

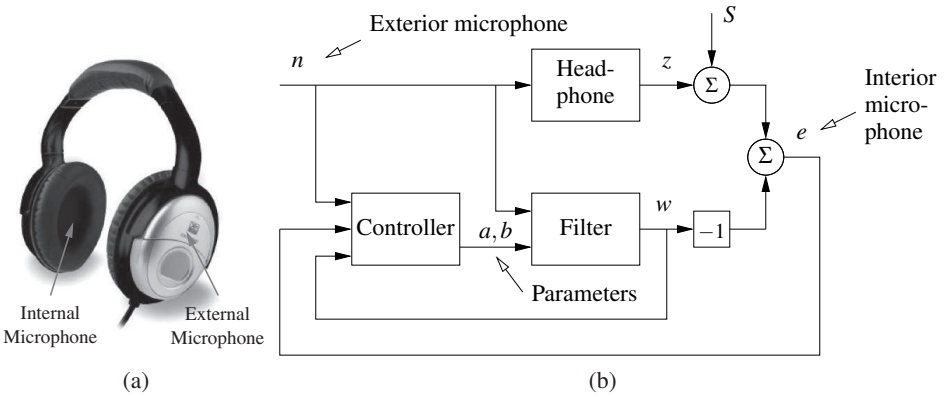


Figure 5.19: Headphones with noise cancellation. Noise is sensed by the exterior microphone (a) and sent to a filter in such a way that it cancels the noise that penetrates the headphone (b). The filter parameters a and b are adjusted by the controller. S represents the input signal to the headphones.

to accomplish its function. By making use of Lyapunov functions we can often design a nonlinear control law that provides stable behavior, as we saw in Example 5.12.

One way to systematically design a nonlinear controller is to begin with a candidate Lyapunov function $V(x)$ and a control system $\dot{x} = f(x,u)$. We say that $V(x)$ is a *control Lyapunov function* if for every x there exists a u such that $\dot{V}(x) = \frac{\partial V}{\partial x} f(x,u) < 0$. In this case, it may be possible to find a function $\alpha(x)$ such that $u = \alpha(x)$ stabilizes the system. The following example illustrates the approach.

Example 5.16 Noise cancellation

Noise cancellation is used in consumer electronics and in industrial systems to reduce the effects of noise and vibrations. The idea is to locally reduce the effect of noise by generating opposing signals. A pair of headphones with noise cancellation such as those shown in Figure 5.19a is a typical example. A schematic diagram of the system is shown in Figure 5.19b. The system has two microphones, one outside the headphones that picks up exterior noise n and another inside the headphones that picks up the signal e, which is a combination of the desired signal S and the external noise that penetrates the headphone. The signal from the exterior microphone is filtered and sent to the headphones in such a way that it cancels the external noise that penetrates into the headphones. The parameters of the filter are adjusted by a feedback mechanism to make the noise signal in the internal microphone as small as possible. The feedback is inherently nonlinear because it acts by changing the parameters of the filter.

To analyze the system we assume for simplicity that the propagation of external noise into the headphones is modeled by a first-order dynamical system described by

$$\frac{dz}{dt} = a_0 z + b_0 n, \tag{5.23}$$

where z is the sound level and the parameters $a_0 < 0$ and b_0 are not known. Assume that the filter is a dynamical system of the same type:

$$\frac{dw}{dt} = aw + bn,$$

where the parameters a and b are adjustable. We wish to find a controller that updates a and b so that they converge to the (unknown) parameters a_0 and b_0. If $a = a_0$ and $b = b_0$ we have $e = S$ and the noise effect of the noise is eliminated. Assuming for simplicity that $S = 0$, introduce $x_1 = e = w - z$, $x_2 = a - a_0$ and $x_3 = b - b_0$; then

$$\frac{dx_1}{dt} = a_0(w-z) + (a-a_0)w + (b-b_0)n = a_0x_1 + x_2w + x_3n. \tag{5.24}$$

We will achieve noise cancellation if we can find a feedback law for changing the parameters a and b so that the error e goes to zero. To do this we choose

$$V(x_1,x_2,x_3) = \frac{1}{2}\left(\alpha x_1^2 + x_2^2 + x_3^2\right)$$

as a candidate Lyapunov function for (5.24). The derivative of V is

$$\dot{V} = \alpha x_1\dot{x}_1 + x_2\dot{x}_2 + x_3\dot{x}_3 = \alpha a_0 x_1^2 + x_2(\dot{x}_2 + \alpha w x_1) + x_3(\dot{x}_3 + \alpha n x_1).$$

Choosing

$$\dot{a} = \dot{x}_2 = -\alpha w x_1 = -\alpha we, \qquad \dot{b} = \dot{x}_3 = -\alpha n x_1 = -\alpha ne, \tag{5.25}$$

we find that $\dot{V} = \alpha a_0 x_1^2 < 0$, and it follows that the quadratic function will decrease as long as $e = x_1 = w - z \neq 0$. The nonlinear feedback (5.25) thus attempts to change the parameters so that the error between the signal and the noise is small. Notice that feedback law (5.25) does not use the model (5.23) explicitly.

A simulation of the system is shown in Figure 5.20. In the simulation we have represented the signal as a pure sinusoid and the noise as broad band noise. The figure shows the dramatic improvement with noise cancellation. The sinusoidal signal is not visible without noise cancellation. The filter parameters change quickly from their initial values $a = b = 0$. Filters of higher order with more coefficients are used in practice. ∇

5.6 Further Reading

The field of dynamical systems has a rich literature that characterizes the possible features of dynamical systems and describes how parametric changes in the dynamics can lead to topological changes in behavior. Readable introductions to dynamical systems are given by Strogatz [Str94] and the highly illustrated text by Abraham and Shaw [AS82]. More technical treatments include Andronov, Vitt and Khaikin [AVK87], Guckenheimer and Holmes [GH83] and Wiggins [Wig90]. For students with a strong interest in mechanics, the texts by Arnold [Arn87] and

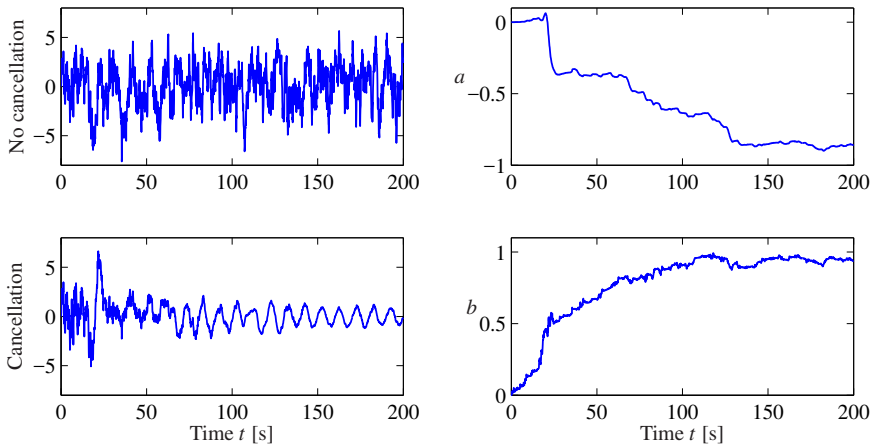


Figure 5.20: Simulation of noise cancellation. The top left figure shows the headphone signal without noise cancellation, and the bottom left figure shows the signal with noise cancellation. The right figures show the parameters a and b of the filter.

Marsden and Ratiu [MR94] provide an elegant approach using tools from differential geometry. Finally, good treatments of dynamical systems methods in biology are given by Wilson [Wil99] and Ellner and Guckenheimer [EG05]. There is a large literature on Lyapunov stability theory, including the classic texts by Malkin [Mal59], Hahn [Hah67] and Krasovski [Kra63]. We highly recommend the comprehensive treatment by Khalil [Kha01].

Exercises

5.1 (Time-invariant systems) Show that if we have a solution of the differential equation (5.1) given by $x(t)$ with initial condition $x(t_0) = x_0$, then $\tilde{x}(\tau) = x(t - t_0)$ is a solution of the differential equation

$$\frac{d\tilde{x}}{d\tau} = F(\tilde{x})$$

with initial condition $\tilde{x}(0) = x_0$, where $\tau = t - t_0$.

5.2 (Flow in a tank) A cylindrical tank has cross section A m^2, effective outlet area a m^2 and inflow q_{in} m^3/s. An energy balance shows that the outlet velocity is $v = \sqrt{2gh}$ m/s, where g m/s^2 is the acceleration of gravity and h is the distance between the outlet and the water level in the tank (in meters). Show that the system can be modeled by

$$\frac{dh}{dt} = -\frac{a}{A}\sqrt{2gh} + \frac{1}{A}q_{in}, \qquad q_{out} = a\sqrt{2gh}.$$

Use the parameters $A = 0.2$, $a = 0.01$. Simulate the system when the inflow is zero and the initial level is $h = 0.2$. Do you expect any difficulties in the simulation?

5.3 (Cruise control) Consider the cruise control system described in Section 4.1. Generate a phase portrait for the closed loop system on flat ground ($\theta = 0$), in third gear, using a PI controller (with $k_p = 0.5$ and $k_i = 0.1$), $m = 1000$ kg and desired speed 20 m/s. Your system model should include the effects of saturating the input between 0 and 1.

5.4 (Lyapunov functions) Consider the second-order system

$$\frac{dx_1}{dt} = -ax_1, \qquad \frac{dx_2}{dt} = -bx_1 - cx_2,$$

where $a, b, c > 0$. Investigate whether the functions

$$V_1(x) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2, \qquad V_2(x) = \frac{1}{2}x_1^2 + \frac{1}{2}(x_2 + \frac{b}{c-a}x_1)^2$$

are Lyapunov functions for the system and give any conditions that must hold.

5.5 (Damped spring–mass system) Consider a damped spring–mass system with dynamics

$$m\ddot{q} + c\dot{q} + kq = 0.$$

A natural candidate for a Lyapunov function is the total energy of the system, given by

$$V = \frac{1}{2}m\dot{q}^2 + \frac{1}{2}kq^2.$$

Use the Krasovski–Lasalle theorem to show that the system is asymptotically stable.

5.6 (Electric generator) The following simple model for an electric generator connected to a strong power grid was given in Exercise 3.7:

$$J\frac{d^2\varphi}{dt^2} = P_m - P_e = P_m - \frac{EV}{X}\sin\varphi.$$

The parameter

$$a = \frac{P_{\max}}{P_m} = \frac{EV}{XP_m} \tag{5.26}$$

is the ratio between the maximum deliverable power $P_{\max} = EV/X$ and the mechanical power P_m.

(a) Consider a as a bifurcation parameter and discuss how the equilibria depend on a.

(b) For $a > 1$, show that there is a center at $\varphi_0 = \arcsin(1/a)$ and a saddle at $\varphi = \pi - \varphi_0$.

(c) Show that if $P_m/J = 1$ there is a solution through the saddle that satisfies

$$\frac{1}{2}\Big(\frac{d\varphi}{dt}\Big)^2 - \varphi + \varphi_0 - a\cos\varphi - \sqrt{a^2-1} = 0. \tag{5.27}$$

Use simulation to show that the stability region is the interior of the area enclosed by this solution. Investigate what happens if the system is in equilibrium with a

value of a that is slightly larger than 1 and a suddenly decreases, corresponding to the reactance of the line suddenly increasing.

5.7 (Lyapunov equation) Show that Lyapunov equation (5.15) always has a solution if all of the eigenvalues of A are in the left half-plane. (Hint: Use the fact that the Lyapunov equation is linear in P and start with the case where A has distinct eigenvalues.)

5.8 (Congestion control) Consider the congestion control problem described in Section 4.4. Confirm that the equilibrium point for the system is given by equation (4.21) and compute the stability of this equilibrium point using a linear approximation.

5.9 (Swinging up a pendulum) Consider the inverted pendulum, discussed in Example 5.4, that is described by

$$\ddot{\theta} = \sin\theta + u\cos\theta,$$

where θ is the angle between the pendulum and the vertical and the control signal u is the acceleration of the pivot. Using the energy function

$$V(\theta,\dot{\theta}) = \cos\theta - 1 + \frac{1}{2}\dot{\theta}^2,$$

show that the state feedback $u = k(V_0 - V)\dot{\theta}\cos\theta$ causes the pendulum to "swing up" to the upright position.

5.10 (Root locus diagram) Consider the linear system

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & -3 \end{pmatrix} x + \begin{pmatrix} -1 \\ 4 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} x,$$

with the feedback $u = -ky$. Plot the location of the eigenvalues as a function the parameter k.

5.11 (Discrete-time Lyapunov function) Consider a nonlinear discrete-time system with dynamics $x[k+1] = f(x[k])$ and equilibrium point $x_e = 0$. Suppose there exists a smooth, positive definite function $V : \mathbb{R}^n \to \mathbb{R}$ such that $V(f(x)) - V(x) < 0$ for $x \neq 0$ and V(0) = 0. Show that $x_e = 0$ is (locally) asymptotically stable.

5.12 (Operational amplifier oscillator) An op amp circuit for an oscillator was shown in Exercise 4.5. The oscillatory solution for that linear circuit was stable but not asymptotically stable. A schematic of a modified circuit that has nonlinear elements is shown in the figure below. The modification is obtained by making a feedback around each operational amplifier that has capacitors using multipliers. The signal $a_e = v_1^2 + v_2^2/\alpha^2 - v_0^2$ is the amplitude error. Show that the system is modeled by

$$\frac{dv_1}{dt} = \frac{R_4}{R_1R_3C_1}v_2 + \frac{1}{R_{11}C_1}v_1(v_0^2 - v_1^2 - \frac{v_2^2}{\alpha^2}),$$
$$\frac{dv_2}{dt} = -\frac{1}{R_2C_2}v_1 + \frac{1}{R_{22}C_2}v_2(v_0^2 - v_1^2 - \frac{v_2^2}{\alpha^2}).$$

Determine α so that the the circuit gives an oscillation with a stable limit cycle with amplitude v_0. (Hint: Use the results of Example 5.8.)

5.13 (Self-activating genetic circuit) Consider the dynamics of a genetic circuit that implements *self-activation*: the protein produced by the gene is an activator for the protein, thus stimulating its own production through positive feedback. Using the models presented in Example 3.15, the dynamics for the system can be written as

$$\frac{dm}{dt} = \frac{\alpha p^2}{1+kp^2} + \alpha_0 - \gamma m, \qquad \frac{dp}{dt} = \beta m - \delta p, \tag{5.28}$$

for $p, m \geq 0$. Find the equilibrium points for the system and analyze the local stability of each using Lyapunov analysis.

5.14 (Diagonal systems) Let $A \in \mathbb{R}^{n\times n}$ be a square matrix with real eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $v_1, \ldots, v_n$. Assume that the eigenvalues are distinct ($\lambda_i \neq \lambda_j$ for $i \neq j$).

(a) Show that $v_i \neq v_j$ for $i \neq j$.

(b) Show that the eigenvectors form a basis for $\mathbb{R}^n$ so that any vector x can be written as $x = \sum \alpha_i v_i$ for $\alpha_i \in \mathbb{R}$.

(c) Let $T = \begin{pmatrix} v_1 & v_2 & \ldots & v_n \end{pmatrix}$ and show that $T^{-1}AT$ is a diagonal matrix of the form (5.8).

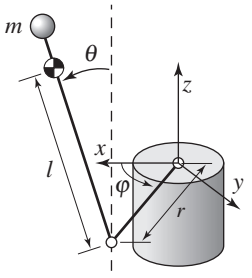
(d) Show that if some of the λ_i are complex numbers, then A can be written as

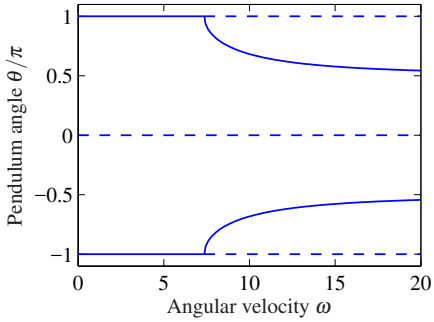
$$A = \begin{pmatrix} \Lambda_1 & & 0 \\ & \ddots & \\ 0 & & \Lambda_k \end{pmatrix} \qquad \text{where} \qquad \Lambda_i = \lambda \in \mathbb{R} \quad \text{or} \quad \Lambda_i = \begin{pmatrix} \sigma & \omega \\ -\omega & \sigma \end{pmatrix}.$$

in an appropriate set of coordinates.

This form of the dynamics of a linear system is often referred to as *block diagonal form*.

5.15 (Furuta pendulum) The Furuta pendulum, an inverted pendulum on a rotating arm, is shown to the left in the figure below.





Consider the situation when the pendulum arm is spinning with constant rate. The system has multiple equilibrium points that depend on the angular velocity ω, as shown in the bifurcation diagram on the right.

The equations of motion for the system are given by

$$J_p\ddot{\theta} - J_p\omega_0^2 \sin\theta\cos\theta - m_p g l \sin\theta = 0,$$

where J_p is the moment of inertia of the pendulum with respect to its pivot, m_p is the pendulum mass, l is the distance between the pivot and the center of mass of the pendulum and ω_0 is the the rate of rotation of the arm.

(a) Determine the equilibria for the system and the condition(s) for stability of each equilibrium point (in terms of ω_0).

(b) Consider the angular velocity as a bifurcation parameter and verify the bifurcation diagram given above. This is an example of a *pitchfork bifurcation*.

Chapter Six
Linear Systems

Few physical elements display truly linear characteristics. For example the relation between force on a spring and displacement of the spring is always nonlinear to some degree. The relation between current through a resistor and voltage drop across it also deviates from a straight-line relation. However, if in each case the relation is reasonably *linear, then it will be found that the system behavior will be very close to that obtained by assuming an ideal, linear physical element, and the analytical simplification is so enormous that we make linear assumptions wherever we can possibly do so in good conscience.*

Robert H. Cannon, *Dynamics of Physical Systems*, 1967 [Can03].

In Chapters 3–5 we considered the construction and analysis of differential equation models for dynamical systems. In this chapter we specialize our results to the case of linear, time-invariant input/output systems. Two central concepts are the matrix exponential and the convolution equation, through which we can completely characterize the behavior of a linear system. We also describe some properties of the input/output response and show how to approximate a nonlinear system by a linear one.

6.1 Basic Definitions

We have seen several instances of linear differential equations in the examples in the previous chapters, including the spring–mass system (damped oscillator) and the operational amplifier in the presence of small (nonsaturating) input signals. More generally, many dynamical systems can be modeled accurately by linear differential equations. Electrical circuits are one example of a broad class of systems for which linear models can be used effectively. Linear models are also broadly applicable in mechanical engineering, for example, as models of small deviations from equilibria in solid and fluid mechanics. Signal-processing systems, including digital filters of the sort used in CD and MP3 players, are another source of good examples, although these are often best modeled in discrete time (as described in more detail in the exercises).

In many cases, we *create* systems with a linear input/output response through the use of feedback. Indeed, it was the desire for linear behavior that led Harold S. Black to the invention of the negative feedback amplifier. Almost all modern signal processing systems, whether analog or digital, use feedback to produce linear or near-linear input/output characteristics. For these systems, it is often useful to represent the input/output characteristics as linear, ignoring the internal details required to get that linear response.

For other systems, nonlinearities cannot be ignored, especially if one cares about the global behavior of the system. The predator–prey problem is one example of this: to capture the oscillatory behavior of the interdependent populations we must include the nonlinear coupling terms. Other examples include switching behavior and generating periodic motion for locomotion. However, if we care about what happens near an equilibrium point, it often suffices to approximate the nonlinear dynamics by their local linearization, as we already explored briefly in Section 5.3. The linearization is essentially an approximation of the nonlinear dynamics around the desired operating point.

Linearity

We now proceed to define linearity of input/output systems more formally. Consider a state space system of the form

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x,u), \tag{6.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$. As in the previous chapters, we will usually restrict ourselves to the single-input, single-output case by taking $p = q = 1$. We also assume that all functions are smooth and that for a reasonable class of inputs (e.g., piecewise continuous functions of time) the solutions of equation (6.1) exist for all time.

It will be convenient to assume that the origin $x = 0$, $u = 0$ is an equilibrium point for this system ($\dot{x} = 0$) and that $h(0,0) = 0$. Indeed, we can do so without loss of generality. To see this, suppose that $(x_e, u_e) \neq (0,0)$ is an equilibrium point of the system with output $y_e = h(x_e, u_e)$. Then we can define a new set of states, inputs and outputs,

$$\tilde{x} = x - x_e, \qquad \tilde{u} = u - u_e, \qquad \tilde{y} = y - y_e,$$

and rewrite the equations of motion in terms of these variables:

$$\begin{aligned} \frac{d}{dt}\tilde{x} &= f(\tilde{x} + x_e, \tilde{u} + u_e) =: \tilde{f}(\tilde{x}, \tilde{u}), \\ \tilde{y} &= h(\tilde{x} + x_e, \tilde{u} + u_e) - y_e =: \tilde{h}(\tilde{x}, \tilde{u}). \end{aligned}$$

In the new set of variables, the origin is an equilibrium point with output 0, and hence we can carry out our analysis in this set of variables. Once we have obtained our answers in this new set of variables, we simply "translate" them back to the original coordinates using $x = \tilde{x} + x_e$, $u = \tilde{u} + u_e$ and $y = \tilde{y} + y_e$.

Returning to the original equations (6.1), now assuming without loss of generality that the origin is the equilibrium point of interest, we write the output $y(t)$ corresponding to the initial condition $x(0) = x_0$ and input $u(t)$ as $y(t; x_0, u)$. Using this notation, a system is said to be a *linear input/output system* if the following

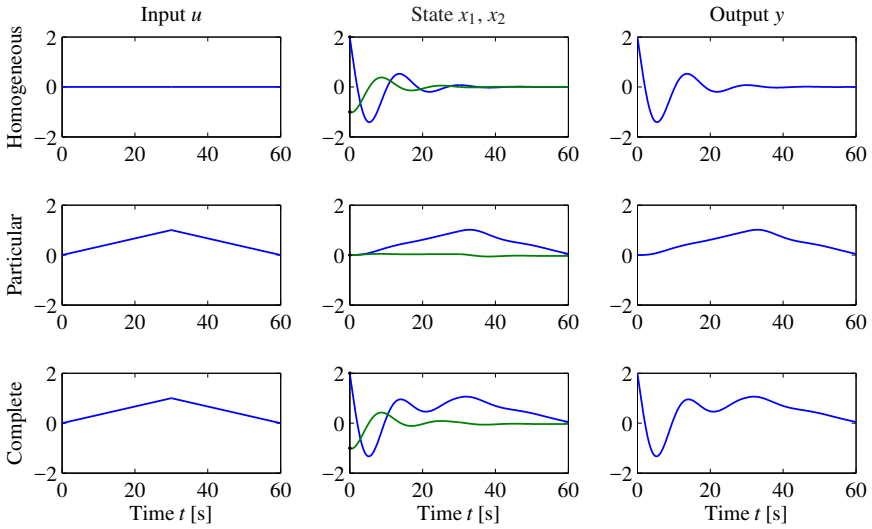


Figure 6.1: Superposition of homogeneous and particular solutions. The first row shows the input, state and output corresponding to the initial condition response. The second row shows the same variables corresponding to zero initial condition but nonzero input. The third row is the complete solution, which is the sum of the two individual solutions.

conditions are satisfied:

$$\begin{aligned} &\text{(i)} \quad y(t;\alpha x_1+\beta x_2,0) = \alpha y(t;x_1,0)+\beta y(t;x_2,0), \\ &\text{(ii)} \quad y(t;\alpha x_0,\delta u) = \alpha y(t;x_0,0)+\delta y(t;0,u), \\ &\text{(iii)} \quad y(t;0,\delta u_1+\gamma u_2) = \delta y(t;0,u_1)+\gamma y(t;0,u_2). \end{aligned} \tag{6.2}$$

Thus, we define a system to be linear if the outputs are jointly linear in the initial condition response $(u = 0)$ and the forced response $(x(0) = 0)$. Property (iii) is a statement of the *principle of superposition*: the response of a linear system to the sum of two inputs u_1 and u_2 is the sum of the outputs y_1 and y_2 corresponding to the individual inputs.

The general form of a linear state space system is

$$\frac{dx}{dt} = Ax+Bu, \qquad y = Cx+Du, \tag{6.3}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{q\times n}$ and $D \in \mathbb{R}^{q\times p}$. In the special case of a single-input, single-output system, B is a column vector, C is a row vector and D is scalar. Equation (6.3) is a system of linear first-order differential equations with input u, state x and output y. It is easy to show that given solutions $x_1(t)$ and $x_2(t)$ for this set of equations, the corresponding outputs satisfy the linearity conditions (6.2).

We define $x_h(t)$ to be the solution with zero input (the *homogeneous solution*) and the solution $x_p(t)$ to be the input dependent solution with zero initial condition (the *forced solution* or a *particular solution*). Figure 6.1 illustrates how these two individual solutions can be superimposed to form the complete solution.

It is also possible to show that if a dynamical system with a finite number of states is input/output linear in the sense we have described, it can always be represented by a state space equation of the form (6.3) through an appropriate choice of state variables. In Section 6.2 we will give an explicit solution of equation (6.3), but we illustrate the basic form through a simple example.

Example 6.1 Scalar system
Consider the first-order differential equation

$$\frac{dx}{dt} = ax + u, \qquad y = x,$$

with $x(0) = x_0$. Let $u_1 = A\sin\omega_1 t$ and $u_2 = B\cos\omega_2 t$. The homogeneous solution is $x_h(t) = e^{at}x_0$, and two particular solutions with $x(0) = 0$ are

$$x_{p1}(t) = -A\frac{-\omega_1 e^{at} + \omega_1\cos\omega_1 t + a\sin\omega_1 t}{a^2+\omega_1^2},$$
$$x_{p2}(t) = B\frac{ae^{at} - a\cos\omega_2 t + \omega_2\sin\omega_2 t}{a^2+\omega_2^2}.$$

Suppose that we now choose $x(0) = \alpha x_0$ and $u = u_1 + u_2$. Then the resulting solution is the weighted sum of the individual solutions:

$$\begin{aligned} x(t) = e^{at}&\left(\alpha x_0 + \frac{A\omega_1}{a^2+\omega_1^2} + \frac{Ba}{a^2+\omega_2^2}\right) \\ &- A\frac{\omega_1\cos\omega_1 t + a\sin\omega_1 t}{a^2+\omega_1^2} + B\frac{-a\cos\omega_2 t + \omega_2\sin\omega_2 t}{a^2+\omega_2^2}. \end{aligned} \tag{6.4}$$

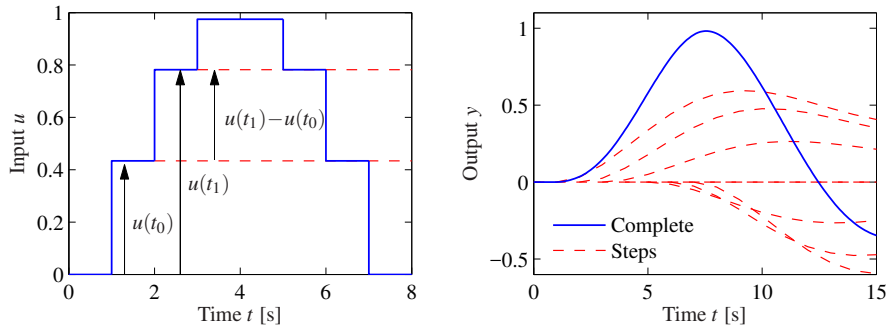
To see this, substitute equation (6.4) into the differential equation. Thus, the properties of a linear system are satisfied. ∇

Time Invariance

Time invariance is an important concept that is used to describe a system whose properties do not change with time. More precisely, for a time-invariant system if the input $u(t)$ gives output $y(t)$, then if we shift the time at which the input is applied by a constant amount a, $u(t+a)$ gives the output $y(t+a)$. Systems that are linear and time-invariant, often called *LTI systems*, have the interesting property that their response to an arbitrary input is completely characterized by their response to step inputs or their response to short "impulses."

To explore the consequences of time invariance, we first compute the response to a piecewise constant input. Assume that the system is initially at rest and consider the piecewise constant input shown in Figure 6.2a. The input has jumps at times t_k, and its values after the jumps are $u(t_k)$. The input can be viewed as a combination of steps: the first step at time t_0 has amplitude $u(t_0)$, the second step at time t_1 has amplitude $u(t_1) - u(t_0)$, etc.

Assuming that the system is initially at an equilibrium point (so that the initial condition response is zero), the response to the input can be obtained by superim-



(a) Piecewise constant input (b) Output response

Figure 6.2: Response to piecewise constant inputs. A piecewise constant signal can be represented as a sum of step signals (a), and the resulting output is the sum of the individual outputs (b).

posing the responses to a combination of step inputs. Let $H(t)$ be the response to a unit step applied at time 0. The response to the first step is then $H(t-t_0)u(t_0)$, the response to the second step is $H(t-t_1)\big(u(t_1)-u(t_0)\big)$, and we find that the complete response is given by

$$\begin{aligned} y(t) &= H(t-t_0)u(t_0) + H(t-t_1)\big(u(t_1)-u(t_0)\big) + \cdots \\ &= \big(H(t-t_0)-H(t-t_1)\big)u(t_0) + \big(H(t-t_1)-H(t-t_2)\big)u(t_1) + \cdots \\ &= \sum_{n=0}^{t_n<t} \big(H(t-t_n)-H(t-t_{n+1})\big)u(t_n) \\ &= \sum_{n=0}^{t_n<t} \frac{H(t-t_n)-H(t-t_{n+1})}{t_{n+1}-t_n} u(t_n)\big(t_{n+1}-t_n\big). \end{aligned}$$

An example of this computation is shown in Figure 6.2b.

The response to a continuous input signal is obtained by taking the limit as $t_{n+1}-t_n \to 0$, which gives

$$y(t) = \int_0^t H'(t-\tau)u(\tau)d\tau, \tag{6.5}$$

where H' is the derivative of the step response, also called the *impulse response*. The response of a linear time-invariant system to any input can thus be computed from the step response. Notice that the output depends only on the input since we assumed the system was initially at rest, $x(0)=0$. We will derive equation (6.5) in a slightly different way in the Section 6.3.

6.2 The Matrix Exponential

Equation (6.5) shows that the output of a linear system with zero initial state can be written as an integral over the inputs $u(t)$. In this section and the next we derive a more general version of this formula, which includes nonzero initial conditions. We begin by exploring the initial condition response using the matrix exponential.

Initial Condition Response

We will now show explicitly show the output of a linear system depends on the input and the initial conditions. We begin by considering the homogeneous response corresponding to the system

$$\frac{dx}{dt} = Ax. \tag{6.6}$$

For the *scalar* differential equation

$$\frac{dx}{dt} = ax, \qquad x \in \mathbb{R}, a \in \mathbb{R},$$

the solution is given by the exponential

$$x(t) = e^{at}x(0).$$

We wish to generalize this to the vector case, where A becomes a matrix. We define the *matrix exponential* as the infinite series

$$e^X = I + X + \frac{1}{2}X^2 + \frac{1}{3!}X^3 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!}X^k, \tag{6.7}$$

where $X \in \mathbb{R}^{n\times n}$ is a square matrix and I is the $n \times n$ identity matrix. We make use of the notation

$$X^0 = I, \qquad X^2 = XX, \qquad X^n = X^{n-1}X,$$

which defines what we mean by the "power" of a matrix. Equation (6.7) is easy to remember since it is just the Taylor series for the scalar exponential, applied to the matrix X. It can be shown that the series in equation (6.7) converges for any matrix $X \in \mathbb{R}^{n\times n}$ in the same way that the normal exponential is defined for any scalar $a \in \mathbb{R}$.

Replacing X in equation (6.7) by At, where $t \in \mathbb{R}$, we find that

$$e^{At} = I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!}A^kt^k,$$

and differentiating this expression with respect to t gives

$$\frac{d}{dt}e^{At} = A + A^2t + \frac{1}{2}A^3t^2 + \cdots = A\sum_{k=0}^{\infty} \frac{1}{k!}A^kt^k = Ae^{At}. \tag{6.8}$$

Multiplying by $x(0)$ from the right, we find that $x(t) = e^{At}x(0)$ is the solution to the differential equation (6.6) with initial condition $x(0)$. We summarize this important result as a proposition.

Proposition 6.1. *The solution to the homogeneous system of differential equations* (6.6) *is given by*

$$x(t) = e^{At}x(0).$$

Notice that the form of the solution is exactly the same as for scalar equations, but we must put the vector $x(0)$ on the right of the matrix e^{At}.

The form of the solution immediately allows us to see that the solution is linear in the initial condition. In particular, if $x_{h1}(t)$ is the solution to equation (6.6) with initial condition $x(0) = x_{01}$ and $x_{h2}(t)$ with initial condition $x(0) = x_{02}$, then the solution with initial condition $x(0) = \alpha x_{01} + \beta x_{02}$ is given by

$$x(t) = e^{At}\left(\alpha x_{01} + \beta x_{02}\right) = \left(\alpha e^{At}x_{01} + \beta e^{At}x_{02}\right) = \alpha x_{h1}(t) + \beta x_{h2}(t).$$

Similarly, we see that the corresponding output is given by

$$y(t) = Cx(t) = \alpha y_{h1}(t) + \beta y_{h2}(t),$$

where $y_{h1}(t)$ and $y_{h2}(t)$ are the outputs corresponding to $x_{h1}(t)$ and $x_{h2}(t)$.

We illustrate computation of the matrix exponential by two examples.

Example 6.2 Double integrator

A very simple linear system that is useful in understanding basic concepts is the second-order system given by

$$\ddot{q} = u, \qquad y = q.$$

This system is called a *double integrator* because the input u is integrated twice to determine the output y.

In state space form, we write $x = (q, \dot{q})$ and

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u.$$

The dynamics matrix of a double integrator is

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

and we find by direct calculation that $A^2 = 0$ and hence

$$e^{At} = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}.$$

Thus the homogeneous solution ($u = 0$) for the double integrator is given by

$$x(t) = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} x_1(0) + tx_2(0) \\ x_2(0) \end{pmatrix},$$
$$y(t) = x_1(0) + tx_2(0).$$

∇

Example 6.3 Undamped oscillator

A model for an oscillator, such as the spring–mass system with zero damping, is

$$\ddot{q} + \omega_0^2 q = u.$$

Putting the system into state space form using $x_1 = q$, $x_2 = \dot{q}/\omega_0$, the dynamics matrix for this system can be written as

$$A = \begin{pmatrix} 0 & \omega_0 \\ -\omega_0 & 0 \end{pmatrix} \qquad \text{and} \qquad e^{At} = \begin{pmatrix} \cos\omega_0 t & \sin\omega_0 t \\ -\sin\omega_0 t & \cos\omega_0 t \end{pmatrix}.$$

This expression for e^{At} can be verified by differentiation:

$$\begin{aligned} \frac{d}{dt}e^{At} &= \begin{pmatrix} -\omega_0\sin\omega_0 t & \omega_0\cos\omega_0 t \\ -\omega_0\cos\omega_0 t & -\omega_0\sin\omega_0 t \end{pmatrix} \\ &= \begin{pmatrix} 0 & \omega_0 \\ -\omega_0 & 0 \end{pmatrix} \begin{pmatrix} \cos\omega_0 t & \sin\omega_0 t \\ -\sin\omega_0 t & \cos\omega_0 t \end{pmatrix} = Ae^{At}. \end{aligned}$$

The solution is then given by

$$x(t) = e^{At}x(0) = \begin{pmatrix} \cos\omega_0 t & \sin\omega_0 t \\ -\sin\omega_0 t & \cos\omega_0 t \end{pmatrix} \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix}.$$

The solution is more complicated if the system has damping:

$$\ddot{q} + 2\zeta\omega_0\dot{q} + \omega_0^2 q = u.$$

If $\zeta < 1$ we have

$$\exp\begin{pmatrix} -\zeta\omega_0 & \omega_d \\ -\omega_d & -\zeta\omega_0 \end{pmatrix} t = e^{-\zeta\omega_0 t}\begin{pmatrix} \cos\omega_d t & \sin\omega_d t \\ -\sin\omega_d t & \cos\omega_d t \end{pmatrix}.$$

where $\omega_d = \omega_0\sqrt{1-\zeta^2}$. The result can be proven by differentiating the exponential matrix. The corresponding results for $\zeta \geq 1$ are given in Exercise 6.4. ∇

An important class of linear systems are those that can be converted into diagonal form by a linear change of coordinates. Suppose that we are given a system

$$\frac{dx}{dt} = Ax$$

such that all the eigenvalues of A are distinct. It can be shown (Exercise 5.14) that there exists an invertible matrix T such that TAT^{-1} is diagonal. If we choose a set of coordinates $z = Tx$, then in the new coordinates the dynamics become

$$\frac{dz}{dt} = T\frac{dx}{dt} = TAx = TAT^{-1}z.$$

By definition of T, this system will be diagonal.

Now consider a diagonal matrix A and the corresponding kth power of At, which is also diagonal:

$$A = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix}, \qquad (At)^k = \begin{pmatrix} \lambda_1^k t^k & & & 0 \\ & \lambda_2^k t^k & & \\ & & \ddots & \\ 0 & & & \lambda_n^k t^k \end{pmatrix},$$

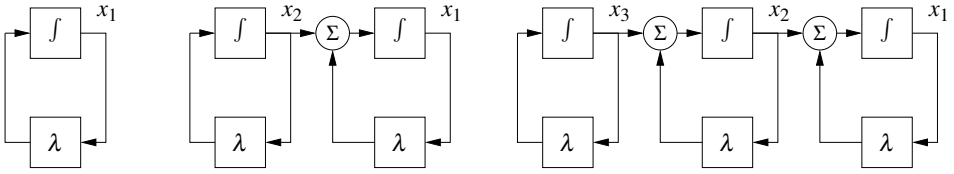


Figure 6.3: Representations of linear systems where the dynamics matrices are Jordan blocks. A first-order Jordan block can be represented as an integrator with feedback λ, as shown on the left. Second- and third-order Jordan blocks can be represented as series connections of integrators with identical feedback, as shown on the right.

It follows from the series expansion that the matrix exponential is given by

$$e^{At} = \begin{pmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n t} \end{pmatrix}.$$

A similar expansion can be done in the case where the eigenvalues are complex, using a block diagonal matrix, similar to what was done in Section 5.3.

Jordan Form

Some matrices with repeated eigenvalues cannot be transformed to diagonal form. They can, however, be transformed to a closely related form, called the *Jordan form*, in which the dynamics matrix has the eigenvalues along the diagonal. When there are equal eigenvalues, there may be 1's appearing in the superdiagonal indicating that there is coupling between the states.

Specifically, we define a matrix to be in Jordan form if it can be written as

$$J = \begin{pmatrix} J_1 & 0 & \dots & 0 & 0 \\ 0 & J_2 & \ddots & 0 & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & & J_{k-1} & 0 \\ 0 & 0 & \dots & 0 & J_k \end{pmatrix}, \quad \text{where} \quad J_i = \begin{pmatrix} \lambda_i & 1 & 0 & \dots & 0 \\ 0 & \lambda_i & 1 & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & & \lambda_i & 1 \\ 0 & 0 & \dots & 0 & \lambda_i \end{pmatrix}, \quad (6.9)$$

where λ_i is an eigenvalue of J_i. Each matrix J_i is called a *Jordan block*. A first-order Jordan block can be represented as a system consisting of an integrator with feedback λ. A Jordan block of higher order can be represented as series connections of such systems, as illustrated in Figure 6.3.

Theorem 6.2 (Jordan decomposition)**.** *Any matrix $A \in \mathbb{R}^{n\times n}$ can be transformed into Jordan form with the eigenvalues of A determining λ_i in the Jordan form.*

Proof. See any standard text on linear algebra, such as Strang [Str88]. The special case where the eigenvalues are distinct is examined in Exercise 5.14. □

Converting a matrix into Jordan form can be complicated, although MATLAB can do this conversion for numerical matrices using the `jordan` function. There is no requirement that the individual λ_i's be distinct, and hence for a given eigenvalue we can have one or more Jordan blocks of different sizes.

Once a matrix is in Jordan form, the exponential of the matrix can be computed in terms of the Jordan blocks:

$$e^{Jt} = \begin{pmatrix} e^{J_1 t} & 0 & \dots & 0 \\ 0 & e^{J_2 t} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & e^{J_k t}. \end{pmatrix}. \tag{6.10}$$

This follows from the block diagonal form of J. The exponentials of the Jordan blocks can in turn be written as

$$e^{J_i t} = \begin{pmatrix} 1 & t & \frac{t^2}{2!} & \dots & \frac{t^{n-1}}{(n-1)!} \\ 0 & 1 & t & \dots & \frac{t^{n-2}}{(n-2)!} \\ \vdots & & 1 & \ddots & \vdots \\ & & & \ddots & t \\ 0 & \dots & & 0 & 1 \end{pmatrix} e^{\lambda_i t}. \tag{6.11}$$

When there are multiple eigenvalues, the invariant subspaces associated with each eigenvalue correspond to the Jordan blocks of the matrix A. Note that some eigenvalues of A may be complex, in which case the transformation T that converts a matrix into Jordan form will also be complex. When λ has a nonzero imaginary component, the solutions will have oscillatory components since

$$e^{(\sigma + i\omega)t} = e^{\sigma t}(\cos \omega t + i \sin \omega t).$$

We can now use these results to prove Theorem 5.1, which states that the equilibrium point $x_e = 0$ of a linear system is asymptotically stable if and only if $\operatorname{Re} \lambda_i < 0$ for all i.

Proof of Theorem 5.1. Let $T \in \mathbb{C}^{n \times n}$ be an invertible matrix that transforms A into Jordan form, $J = TAT^{-1}$. Using coordinates $z = Tx$, we can write the solution $z(t)$ as

$$z(t) = e^{Jt} z(0),$$

where $z(0) = Tx(0)$, so that $x(t) = T^{-1} e^{Jt} z(0)$.

The solution $z(t)$ can be written in terms of the elements of the matrix exponential. From equation (6.11) these elements all decay to zero for arbitrary $z(0)$ if and only if $\operatorname{Re} \lambda_i < 0$ for all i. Furthermore, if any λ_i has positive real part, then there exists an initial condition $z(0)$ such that the corresponding solution increases without bound. Since we can scale this initial condition to be arbitrarily small, it follows that the equilibrium point is unstable if any eigenvalue has positive real part. □

The existence of a canonical form allows us to prove many properties of linear systems by changing to a set of coordinates in which the A matrix is in Jordan form. We illustrate this in the following proposition, which follows along the same lines as the proof of Theorem 5.1.

Proposition 6.3. *Suppose that the system*

$$\frac{dx}{dt} = Ax$$

has no eigenvalues with strictly positive real part and one or more eigenvalues with zero real part. Then the system is stable if and only if the Jordan blocks corresponding to each eigenvalue with zero real part are scalar (1×1) blocks.

Proof. See Exercise 6.6b. □

The following example illustrates the use of the Jordan form.

Example 6.4 Linear model of a vectored thrust aircraft
Consider the dynamics of a vectored thrust aircraft such as that described in Example 3.11. Suppose that we choose $u_1 = u_2 = 0$ so that the dynamics of the system become

$$\frac{dz}{dt} = \begin{pmatrix} z_4 \\ z_5 \\ z_6 \\ -g\sin z_3 - \frac{c}{m} z_4 \\ g(\cos z_3 - 1) - \frac{c}{m} z_5 \\ 0 \end{pmatrix}, \tag{6.12}$$

where $z = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})$. The equilibrium points for the system are given by setting the velocities $\dot{x}$, $\dot{y}$ and $\dot{\theta}$ to zero and choosing the remaining variables to satisfy

$$\begin{aligned} -g\sin z_{3,e} &= 0 \\ g(\cos z_{3,e} - 1) &= 0 \end{aligned} \qquad \Longrightarrow \qquad z_{3,e} = \theta_e = 0.$$

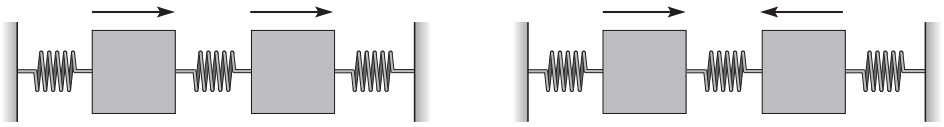
This corresponds to the upright orientation for the aircraft. Note that x_e and y_e are not specified. This is because we can translate the system to a new (upright) position and still obtain an equilibrium point.

To compute the stability of the equilibrium point, we compute the linearization using equation (5.11):

$$A = \left.\frac{\partial F}{\partial z}\right|_{z_e} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -g & -c/m & 0 & 0 \\ 0 & 0 & 0 & 0 & -c/m & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The eigenvalues of the system can be computed as

$$\lambda(A) = \{0, 0, 0, 0, -c/m, -c/m\}.$$



(a) Mode 1 (b) Mode 2

Figure 6.4: Modes of vibration for a system consisting of two masses connected by springs. In (a) the masses move left and right in synchronization in (b) they move toward or against each other.

We see that the linearized system is not asymptotically stable since not all of the eigenvalues have strictly negative real part.

To determine whether the system is stable in the sense of Lyapunov, we must make use of the Jordan form. It can be shown that the Jordan form of A is given by

$$J = \left(\begin{array}{c|ccc|c|c} 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -c/m & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -c/m \end{array}\right).$$

Since the second Jordan block has eigenvalue 0 and is not a simple eigenvalue, the linearization is unstable. ∇

Eigenvalues and Modes

The eigenvalues and eigenvectors of a system provide a description of the types of behavior the system can exhibit. For oscillatory systems, the term *mode* is often used to describe the vibration patterns that can occur. Figure 6.4 illustrates the modes for a system consisting of two masses connected by springs. One pattern is when both masses oscillate left and right in unison, and another is when the masses move toward and away from each other.

The initial condition response of a linear system can be written in terms of a matrix exponential involving the dynamics matrix A. The properties of the matrix A therefore determine the resulting behavior of the system. Given a matrix $A \in \mathbb{R}^{n\times n}$, recall that v is an eigenvector of A with eigenvalue λ if

$$Av = \lambda v.$$

In general λ and v may be complex-valued, although if A is real-valued, then for any eigenvalue λ its complex conjugate λ^* will also be an eigenvalue (with v^* as the corresponding eigenvector).

Suppose first that λ and v are a real-valued eigenvalue/eigenvector pair for A. If we look at the solution of the differential equation for $x(0) = v$, it follows from

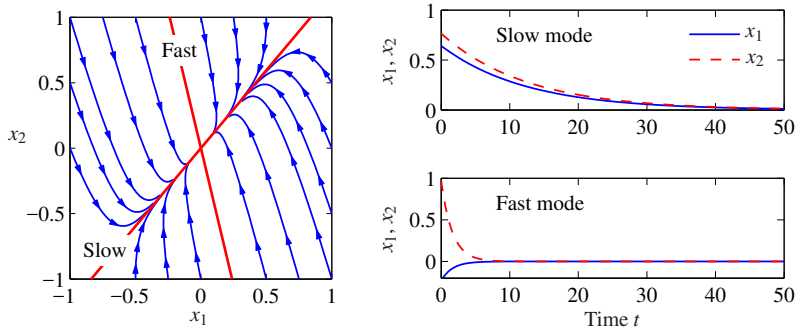


Figure 6.5: The notion of modes for a second-order system with real eigenvalues. The left figure shows the phase portrait and the modes corresponding to solutions that start on the eigenvectors (bold lines). The corresponding time functions are shown on the right.

the definition of the matrix exponential that

$$e^{At}v = \left(I + At + \frac{1}{2}A^2t^2 + \cdots\right)v = v + \lambda tv + \frac{\lambda^2 t^2}{2}v + \cdots = e^{\lambda t}v.$$

The solution thus lies in the subspace spanned by the eigenvector. The eigenvalue λ describes how the solution varies in time, and this solution is often called a *mode* of the system. (In the literature, the term "mode" is also often used to refer to the eigenvalue rather than the solution.)

If we look at the individual elements of the vectors x and v, it follows that

$$\frac{x_i(t)}{x_j(t)} = \frac{e^{\lambda t}v_i}{e^{\lambda t}v_j} = \frac{v_i}{v_j},$$

and hence the ratios of the components of the state x are constants for a (real) mode. The eigenvector thus gives the "shape" of the solution and is also called a *mode shape* of the system. Figure 6.5 illustrates the modes for a second-order system consisting of a fast mode and a slow mode. Notice that the state variables have the same sign for the slow mode and different signs for the fast mode.

The situation is more complicated when the eigenvalues of A are complex. Since A has real elements, the eigenvalues and the eigenvectors are complex conjugates $\lambda = \sigma \pm i\omega$ and $v = u \pm iw$, which implies that

$$u = \frac{v + v^*}{2}, \qquad w = \frac{v - v^*}{2i}.$$

Making use of the matrix exponential, we have

$$e^{At}v = e^{\lambda t}(u + iw) = e^{\sigma t}\big((u\cos\omega t - w\sin\omega t) + i(u\sin\omega t + w\cos\omega t)\big),$$

from which it follows that

$$e^{At}u = \frac{1}{2}\left(e^{At}v + e^{At}v^*\right) = ue^{\sigma t}\cos\omega t - we^{\sigma t}\sin\omega t,$$
$$e^{At}w = \frac{1}{2i}\left(e^{At}v - e^{At}v^*\right) = ue^{\sigma t}\sin\omega t + we^{\sigma t}\cos\omega t.$$

A solution with initial conditions in the subspace spanned by the real part u and imaginary part w of the eigenvector will thus remain in that subspace. The solution will be a logarithmic spiral characterized by σ and ω. We again call the solution corresponding to λ a mode of the system, and v the mode shape.

If a matrix A has n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$, then the initial condition response can be written as a linear combination of the modes. To see this, suppose for simplicity that we have all real eigenvalues with corresponding unit eigenvectors $v_1, \ldots, v_n$. From linear algebra, these eigenvectors are linearly independent, and we can write the initial condition $x(0)$ as

$$x(0) = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n.$$

Using linearity, the initial condition response can be written as

$$x(t) = \alpha_1 e^{\lambda_1 t} v_1 + \alpha_2 e^{\lambda_2 t} v_2 + \cdots + \alpha_n e^{\lambda_n t} v_n.$$

Thus, the response is a linear combination of the modes of the system, with the amplitude of the individual modes growing or decaying as $e^{\lambda_i t}$. The case for distinct complex eigenvalues follows similarly (the case for nondistinct eigenvalues is more subtle and requires making use of the Jordan form discussed in the previous section).

Example 6.5 Coupled spring–mass system
Consider the spring–mass system shown in Figure 6.4, but with the addition of dampers on each mass. The equations of motion of the system are

$$m\ddot{q}_1 = -2kq_1 - c\dot{q}_1 + kq_2, \qquad m\ddot{q}_2 = kq_1 - 2kq_2 - c\dot{q}_2.$$

In state space form, we define the state to be $x = (q_1, q_2, \dot{q}_1, \dot{q}_2)$, and we can rewrite the equations as

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{2k}{m} & \dfrac{k}{m} & -\dfrac{c}{m} & 0 \\ \dfrac{k}{m} & -\dfrac{2k}{m} & 0 & -\dfrac{c}{m} \end{pmatrix} x.$$

We now define a transformation $z = Tx$ that puts this system into a simpler form.

Let $z_1 = \frac{1}{2}(q_1 + q_2)$, $z_2 = \dot{z}_1$, $z_3 = \frac{1}{2}(q_1 - q_2)$ and $z_4 = \dot{z}_3$, so that

$$z = Tx = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} x.$$

In the new coordinates, the dynamics become

$$\frac{dz}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{3k}{m} & -\frac{c}{m} \end{pmatrix} z,$$

and we see that the system is in block diagonal form.

In the z coordinates, the states z_1 and z_2 parameterize one mode with eigenvalues $\lambda \approx -c/(2m) \pm i\sqrt{k/m}$, and the states z_3 and z_4 another mode with $\lambda \approx -c/(2m) \pm i\sqrt{3k/m}$. From the form of the transformation T we see that these modes correspond exactly to the modes in Figure 6.4, in which q_1 and q_2 move either toward or against each other. The real and imaginary parts of the eigenvalues give the decay rates σ and frequencies ω for each mode. ∇

6.3 Input/Output Response

In the previous section we saw how to compute the initial condition response using the matrix exponential. In this section we derive the convolution equation, which includes the inputs and outputs as well.

The Convolution Equation

We return to the general input/output case in equation (6.3), repeated here:

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du. \tag{6.13}$$

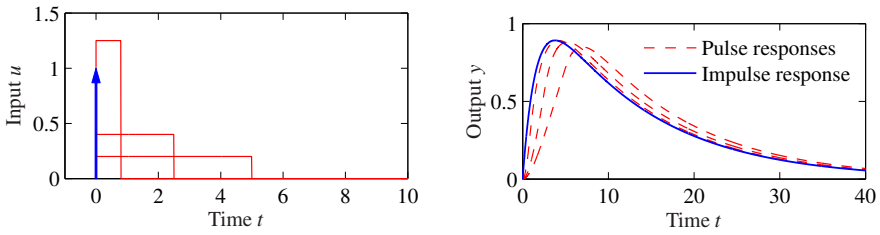
Using the matrix exponential, the solution to equation (6.13) can be written as follows.

Theorem 6.4. *The solution to the linear differential equation* (6.13) *is given by*

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \tag{6.14}$$

Proof. To prove this, we differentiate both sides and use the property (6.8) of the matrix exponential. This gives

$$\frac{dx}{dt} = Ae^{At}x(0) + \int_0^t Ae^{A(t-\tau)}Bu(\tau)d\tau + Bu(t) = Ax + Bu,$$



(a) Pulse and impulse functions (b) Pulse and impulse responses

Figure 6.6: Pulse response and impulse response. (a) The rectangles show pulses of width 5, 2.5 and 0.8, each with total area equal to 1. The arrow denotes an impulse $\delta(t)$ defined by equation (6.17). The corresponding pulse responses for a linear system with eigenvalues $\lambda = \{-0.08, -0.62\}$ are shown in (b) as dashed lines. The solid line is the true impulse response, which is well approximated by a pulse of duration 0.8.

which proves the result since the initial conditions are also met. Notice that the calculation is essentially the same as for proving the result for a first-order equation. □

It follows from equations (6.13) and (6.14) that the input/output relation for a linear system is given by

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t). \tag{6.15}$$

It is easy to see from this equation that the output is jointly linear in both the initial conditions and the input, which follows from the linearity of matrix/vector multiplication and integration.

Equation (6.15) is called the *convolution equation*, and it represents the general form of the solution of a system of coupled linear differential equations. We see immediately that the dynamics of the system, as characterized by the matrix A, play a critical role in both the stability and performance of the system. Indeed, the matrix exponential describes *both* what happens when we perturb the initial condition and how the system responds to inputs.

Another interpretation of the convolution equation can be given using the concept of the *impulse response* of a system. Consider the application of an input signal $u(t)$ given by the following equation:

$$u(t) = p_\varepsilon(t) = \begin{cases} 0 & t < 0 \\ 1/\varepsilon & 0 \le t < \varepsilon \\ 0 & t \ge \varepsilon. \end{cases} \tag{6.16}$$

This signal is a *pulse* of duration ε and amplitude $1/\varepsilon$, as illustrated in Figure 6.6a. We define an *impulse* $\delta(t)$ to be the limit of this signal as $\varepsilon \to 0$:

$$\delta(t) = \lim_{\varepsilon \to 0} p_\varepsilon(t). \tag{6.17}$$

This signal, sometimes called a *delta function,* is not physically achievable but provides a convenient abstraction in understanding the response of a system. Note that the integral of an impulse is 1:

$$\int_0^t \delta(\tau)\,d\tau = \int_0^t \lim_{\varepsilon\to 0} p_\varepsilon(t)\,d\tau = \lim_{\varepsilon\to 0}\int_0^t p_\varepsilon(t)\,d\tau$$
$$= \lim_{\varepsilon\to 0}\int_0^\varepsilon 1/\varepsilon\,d\tau = 1 \qquad t > 0.$$

In particular, the integral of an impulse over an arbitrarily short period of time that includes the origin is identically 1.

We define the *impulse response* $h(t)$ for a system as the output of the system with zero initial condition and having an impulse as its input:

$$h(t) = \int_0^t Ce^{A(t-\tau)}B\delta(\tau)\,d\tau + D\delta(t) = Ce^{At}B + D\delta(t), \tag{6.18}$$

where the second equality follows from the fact that $\delta(t)$ is zero everywhere except the origin and its integral is identically 1. We can now write the convolution equation in terms of the initial condition response and the convolution of the impulse response and the input signal:

$$y(t) = Ce^{At}x(0) + \int_0^t h(t-\tau)u(\tau)\,d\tau \tag{6.19}$$

One interpretation of this equation, explored in Exercise 6.2, is that the response of the linear system is the superposition of the response to an infinite set of shifted impulses whose magnitudes are given by the input $u(t)$. This is essentially the argument used in analyzing Figure 6.2 and deriving equation (6.5). Note that the second term in equation (6.19) is identical to equation (6.5), and it can be shown that the impulse response is the derivative of the step response.

The use of pulses $p_\varepsilon(t)$ as approximations of the impulse function $\delta(t)$ also provides a mechanism for identifying the dynamics of a system from experiments. Figure 6.6b shows the pulse responses of a system for different pulse widths. Notice that the pulse responses approach the impulse response as the pulse width goes to zero. As a general rule, if the fastest eigenvalue of a stable system has real part $-\sigma_{\max}$, then a pulse of length ε will provide a good estimate of the impulse response if $\varepsilon\sigma_{\max} \ll 1$. Note that for Figure 6.6, a pulse width of $\varepsilon = 1$ s gives $\varepsilon\sigma_{\max} = 0.62$ and the pulse response is already close to the impulse response.

Coordinate Invariance

The components of the input vector u and the output vector y are determined by the chosen inputs and outputs of a model, but the state variables depend on the coordinate frame chosen to represent the state. This choice of coordinates affects the values of the matrices A, B and C that are used in the model. (The direct term D is not affected since it maps inputs to outputs.) We now investigate some of the consequences of changing coordinate systems.

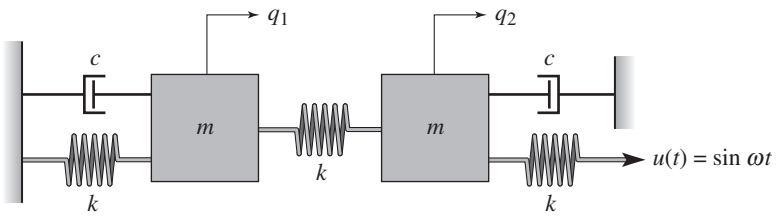


Figure 6.7: Coupled spring mass system. Each mass is connected to two springs with stiffness k and a viscous damper with damping coefficient c. The mass on the right is driven through a spring connected to a sinusoidally varying attachment.

Introduce new coordinates z by the transformation $z = Tx$, where T is an invertible matrix. It follows from equation (6.3) that

$$\begin{aligned}\frac{dz}{dt} &= T(Ax+Bu) = TAT^{-1}z + TBu =: \tilde{A}z + \tilde{B}u,\\ y &= Cx + Du = CT^{-1}z + Du =: \tilde{C}z + Du.\end{aligned}$$

The transformed system has the same form as equation (6.3), but the matrices A, B and C are different:

$$\tilde{A} = TAT^{-1}, \qquad \tilde{B} = TB, \qquad \tilde{C} = CT^{-1}. \tag{6.20}$$

There are often special choices of coordinate systems that allow us to see a particular property of the system, hence coordinate transformations can be used to gain new insight into the dynamics. The eigenvalues of $\tilde{A}$ are the same as those of A, so stability is not affected.

We can also compare the solution of the system in transformed coordinates to that in the original state coordinates. We make use of an important property of the exponential map,

$$e^{TST^{-1}} = Te^{S}T^{-1},$$

which can be verified by substitution in the definition of the matrix exponential. Using this property, it is easy to show that

$$x(t) = T^{-1}z(t) = T^{-1}e^{\tilde{A}t}Tx(0) + T^{-1}\int_0^t e^{\tilde{A}(t-\tau)}\tilde{B}u(\tau)\,d\tau.$$

From this form of the equation, we see that if it is possible to transform A into a form $\tilde{A}$ for which the matrix exponential is easy to compute, we can use that computation to solve the general convolution equation for the untransformed state x by simple matrix multiplications. This technique is illustrated in the following example.

Example 6.6 Coupled spring–mass system

Consider the coupled spring–mass system shown in Figure 6.7. The input to this system is the sinusoidal motion of the position of the rightmost spring, and the output is the position of each mass, q_1 and q_2. The equations of motion are given

by

$$m\ddot{q}_1 = -2kq_1 - c\dot{q}_1 + kq_2, \qquad m\ddot{q}_2 = kq_1 - 2kq_2 - c\dot{q}_2 + ku.$$

In state space form, we define the state to be $x = (q_1, q_2, \dot{q}_1, \dot{q}_2)$, and we can rewrite the equations as

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{2k}{m} & \frac{k}{m} & -\frac{c}{m} & 0 \\ \frac{k}{m} & -\frac{2k}{m} & 0 & -\frac{c}{m} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{k}{m} \end{pmatrix} u.$$

This is a coupled set of four differential equations and is quite complicated to solve in analytical form.

The dynamics matrix is the same as in Example 6.5, and we can use the coordinate transformation defined there to put the system in block diagonal form:

$$\frac{dz}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{3k}{m} & -\frac{c}{m} \end{pmatrix} z + \begin{pmatrix} 0 \\ \frac{k}{2m} \\ 0 \\ -\frac{k}{2m} \end{pmatrix} u.$$

Note that the resulting matrix equations are block diagonal and hence decoupled. We can solve for the solutions by computing the solutions of two sets of second-order systems represented by the states (z_1, z_2) and (z_3, z_4). Indeed, the functional form of each set of equations is identical to that of a single spring–mass system. (The explicit solution is derived in Section 7.3.)

Once we have solved the two sets of independent second-order equations, we can recover the dynamics in the original coordinates by inverting the state transformation and writing $x = T^{-1}z$. We can also determine the stability of the system by looking at the stability of the independent second-order systems. ∇

Steady-State Response

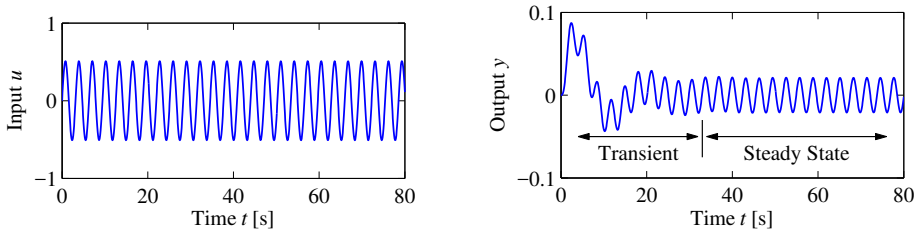
Given a linear input/output system

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du, \tag{6.21}$$

the general form of the solution to equation (6.21) is given by the convolution equation:

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t).$$

We see from the form of this equation that the solution consists of an initial condition response and an input response.



(a) Input (b) Output

Figure 6.8: Transient versus steady-state response. The input to a linear system is shown in (a), and the corresponding output with $x(0) = 0$ is shown in (b). The output signal initially undergoes a transient before settling into its steady-state behavior.

The input response, corresponding to the last two terms in the equation above, itself consists of two components—the *transient response* and the *steady-state response*. The transient response occurs in the first period of time after the input is applied and reflects the mismatch between the initial condition and the steady-state solution. The steady-state response is the portion of the output response that reflects the long-term behavior of the system under the given inputs. For inputs that are periodic the steady-state response will often be periodic, and for constant inputs the response will often be constant. An example of the transient and the steady-state response for a periodic input is shown in Figure 6.8.

A particularly common form of input is a *step input*, which represents an abrupt change in input from one value to another. A *unit step* (sometimes called the Heaviside step function) is defined as

$$u(t) = S(t) = \begin{cases} 0 & t = 0 \\ 1 & t > 0. \end{cases}$$

The *step response* of the system (6.21) is defined as the output $y(t)$ starting from zero initial condition (or the appropriate equilibrium point) and given a step input. We note that the step input is discontinuous and hence is not practically implementable. However, it is a convenient abstraction that is widely used in studying input/output systems.

We can compute the step response to a linear system using the convolution equation. Setting $x(0) = 0$ and using the definition of the step input above, we have

$$\begin{aligned} y(t) &= \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t) = C\int_0^t e^{A(t-\tau)}Bd\tau + D \\ &= C\int_0^t e^{A\sigma}Bd\sigma + D = C\left(A^{-1}e^{A\sigma}B\right)\Big|_{\sigma=0}^{\sigma=t} + D \\ &= CA^{-1}e^{At}B - CA^{-1}B + D. \end{aligned}$$

If all eigenvalues of A have negative real parts (implying that the origin is a stable

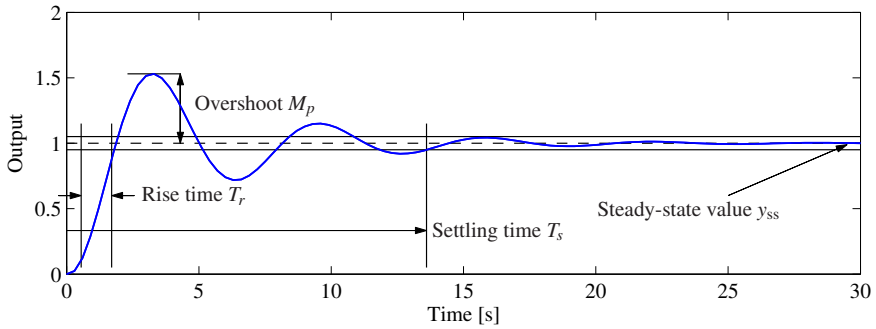


Figure 6.9: Sample step response. The rise time, overshoot, settling time and steady-state value give the key performance properties of the signal.

equilibrium point in the absence of any input), then we can rewrite the solution as

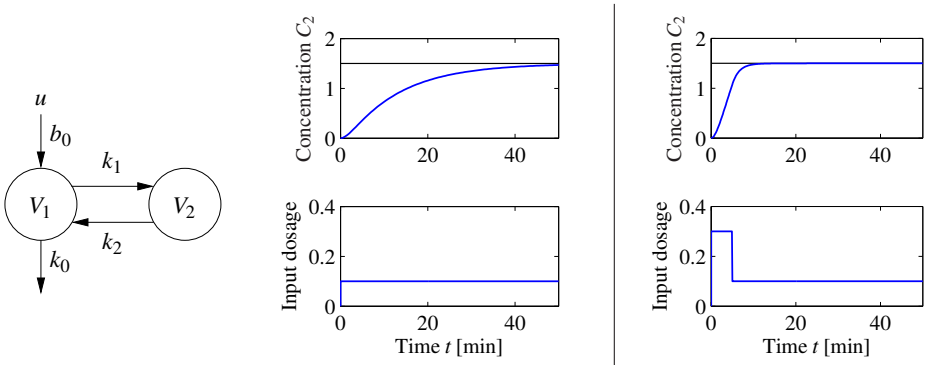
$$y(t) = \underbrace{CA^{-1}e^{At}B}_{\text{transient}} + \underbrace{D - CA^{-1}B}_{\text{steady-state}}, \qquad t > 0. \tag{6.22}$$

The first term is the transient response and decays to zero as $t \to \infty$. The second term is the steady-state response and represents the value of the output for large time.

A sample step response is shown in Figure 6.9. Several key properties are used when describing a step response. The *steady-state value* y_{ss} of a step response is the final level of the output, assuming it converges. The *rise time* T_r is the amount of time required for the signal to first go from 10% of its final value to 90% of its final value. It is possible to define other limits as well, but in this book we shall use these percentages unless otherwise indicated. The *overshoot* M_p is the percentage of the final value by which the signal initially rises above the final value. This usually assumes that future values of the signal do not overshoot the final value by more than this initial transient, otherwise the term can be ambiguous. Finally, the *settling time* T_s is the amount of time required for the signal to stay within 2% of its final value for all future times. The settling time is also sometimes defined as reaching 1% or 5% of the final value (see Exercise 6.7). In general these performance measures can depend on the amplitude of the input step, but for linear systems the last three quantities defined above are independent of the size of the step.

Example 6.7 Compartment model

Consider the compartment model illustrated in Figure 6.10 and described in more detail in Section 4.6. Assume that a drug is administered by constant infusion in compartment V_1 and that the drug has its effect in compartment V_2. To assess how quickly the concentration in the compartment reaches steady state we compute the step response, which is shown in Figure 6.10b. The step response is quite slow, with a settling time of 39 min. It is possible to obtain the steady-state concentration much faster by having a faster injection rate initially, as shown in Figure 6.10c.



(a) Schematic (b) Step input (c) Pulse input

Figure 6.10: Response of a compartment model to a constant drug infusion. A simple diagram of the system is shown in (a). The step response (b) shows the rate of concentration buildup in compartment 2. In (c) a pulse of initial concentration is used to speed up the response.

The response of the system in this case can be computed by combining two step responses (Exercise 6.3). ∇

Another common input signal to a linear system is a sinusoid (or a combination of sinusoids). The *frequency response* of an input/output system measures the way in which the system responds to a sinusoidal excitation on one of its inputs. As we have already seen for scalar systems, the particular solution associated with a sinusoidal excitation is itself a sinusoid at the same frequency. Hence we can compare the magnitude and phase of the output sinusoid to the input.

To see this in more detail, we must evaluate the convolution equation (6.15) for $u = \cos \omega t$. This turns out to be a very messy calculation, but we can make use of the fact that the system is linear to simplify the derivation. In particular, we note that

$$\cos \omega t = \frac{1}{2}\left(e^{i\omega t} + e^{-i\omega t}\right).$$

Since the system is linear, it suffices to compute the response of the system to the complex input $u(t) = e^{st}$ and we can then reconstruct the input to a sinusoid by averaging the responses corresponding to $s = i\omega$ and $s = -i\omega$.

Applying the convolution equation to the input $u = e^{st}$ we have

$$\begin{aligned} y(t) &= Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Be^{s\tau}d\tau + De^{st} \\ &= Ce^{At}x(0) + Ce^{At}\int_0^t e^{(sI-A)\tau}Bd\tau + De^{st}. \end{aligned}$$

If we assume that none of the eigenvalues of A are equal to $s = \pm i\omega$, then the

matrix $sI - A$ is invertible, and we can write

$$\begin{aligned} y(t) &= Ce^{At}x(0) + Ce^{At}\left((sI-A)^{-1}e^{(sI-A)\tau}B\right)\Big|_0^t + De^{st} \\ &= Ce^{At}x(0) + Ce^{At}(sI-A)^{-1}\left(e^{(sI-A)t} - I\right)B + De^{st} \\ &= Ce^{At}x(0) + C(sI-A)^{-1}e^{st}B - Ce^{At}(sI-A)^{-1}B + De^{st}, \end{aligned}$$

and we obtain

$$y(t) = \underbrace{Ce^{At}\left(x(0) - (sI-A)^{-1}B\right)}_{\text{transient}} + \underbrace{\left(C(sI-A)^{-1}B + D\right)e^{st}}_{\text{steady-state}}. \tag{6.23}$$

Notice that once again the solution consists of both a transient component and a steady-state component. The transient component decays to zero if the system is asymptotically stable and the steady-state component is proportional to the (complex) input $u = e^{st}$.

We can simplify the form of the solution slightly further by rewriting the steady-state response as

$$y_{\text{ss}}(t) = Me^{i\theta}e^{st} = Me^{(st+i\theta)},$$

where

$$Me^{i\theta} = C(sI-A)^{-1}B + D \tag{6.24}$$

and M and θ represent the magnitude and phase of the complex number $C(sI - A)^{-1}B + D$. When $s = i\omega$, we say that M is the *gain* and θ is the *phase* of the system at a given forcing frequency ω. Using linearity and combining the solutions for $s = +i\omega$ and $s = -i\omega$, we can show that if we have an input $u = A_u \sin(\omega t + \psi)$ and an output $y = A_y \sin(\omega t + \varphi)$, then

$$\text{gain}(\omega) = \frac{A_y}{A_u} = M, \qquad \text{phase}(\omega) = \varphi - \psi = \theta.$$

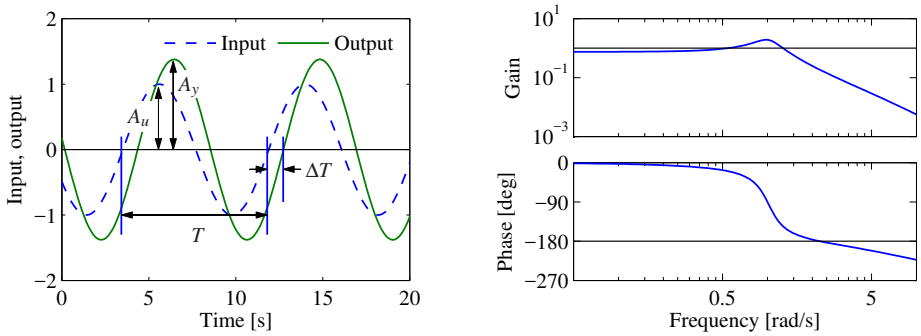
The steady-state solution for a sinusoid $u = \cos\omega t = \sin(\omega t + \pi/2)$ is now given by

$$y_{\text{ss}}(t) = M\cos(\omega t + \theta).$$

If the phase θ is positive, we say that the output *leads* the input, otherwise we say it *lags* the input.

A sample steady-state sinusoidal response is illustrated in Figure 6.11a. The dashed line shows the input sinusoid, which has amplitude 1. The output sinusoid is shown as a solid line and has a different amplitude plus a shifted phase. The gain is the ratio of the amplitudes of the sinusoids, which can be determined by measuring the height of the peaks. The phase is determined by comparing the ratio of the time between zero crossings of the input and output to the overall period of the sinusoid:

$$\theta = -2\pi \cdot \frac{\Delta T}{T}.$$



(a) Input/output response (b) Frequency response

Figure 6.11: Steady-state response of an asymptotically stable linear system to a sinusoid. (a) A sinusoidal input of magnitude A_u (dashed) gives a sinusoidal output of magnitude A_y (solid), delayed by ΔT seconds. (b) Frequency response, showing gain and phase. The gain is given by the ratio of the output amplitude to the input amplitude, $M = A_y/A_u$. The phase lag is given by $\theta = -2\pi\Delta T/T$; it is negative for the case shown because the output lags the input.

A convenient way to view the frequency response is to plot how the gain and phase in equation (6.24) depend on ω (through $s = i\omega$). Figure 6.11b shows an example of this type of representation.

Example 6.8 Active band-pass filter

Consider the op amp circuit shown in Figure 6.12a. We can derive the dynamics of the system by writing the *nodal equations*, which state that the sum of the currents at any node must be zero. Assuming that $v_- = v_+ = 0$, as we did in Section 4.3, we have

$$0 = \frac{v_1 - v_2}{R_1} - C_1\frac{dv_2}{dt}, \qquad 0 = C_1\frac{dv_2}{dt} + \frac{v_3}{R_2} + C_2\frac{dv_3}{dt}.$$

Choosing v_2 and v_3 as our states and using these equations, we obtain

$$\frac{dv_2}{dt} = \frac{v_1 - v_2}{R_1C_1}, \qquad \frac{dv_3}{dt} = \frac{-v_3}{R_2C_2} - \frac{v_1 - v_2}{R_1C_2}.$$

Rewriting these in linear state space form, we obtain

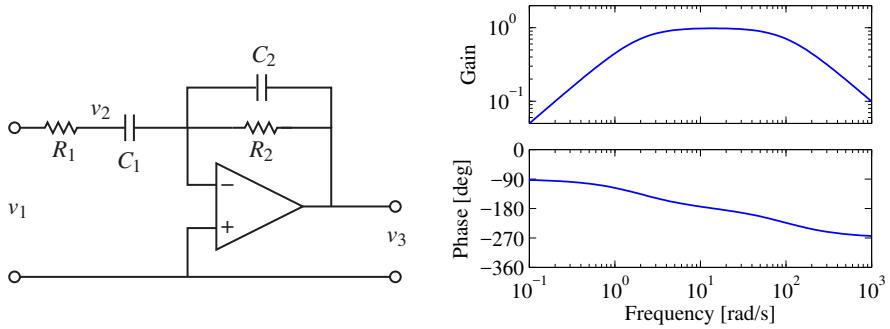
$$\frac{dx}{dt} = \begin{pmatrix} -\dfrac{1}{R_1C_1} & 0 \\ \dfrac{1}{R_1C_2} & -\dfrac{1}{R_2C_2} \end{pmatrix} x + \begin{pmatrix} \dfrac{1}{R_1C_1} \\ \dfrac{-1}{R_1C_2} \end{pmatrix} u, \qquad y = \begin{pmatrix} 0 & 1 \end{pmatrix} x, \tag{6.25}$$

where $x = (v_2, v_3)$, $u = v_1$ and $y = v_3$.

The frequency response for the system can be computed using equation (6.24):

$$Me^{j\theta} = C(sI - A)^{-1}B + D = -\frac{R_2}{R_1}\frac{R_1C_1s}{(1 + R_1C_1s)(1 + R_2C_2s)}, \qquad s = i\omega.$$

The magnitude and phase are plotted in Figure 6.12b for $R_1 = 100\ \Omega$, $R_2 = 5\ \text{k}\Omega$



(a) Circuit diagram (b) Frequency response

Figure 6.12: Active band-pass filter. The circuit diagram (a) shows an op amp with two *RC* filters arranged to provide a band-pass filter. The plot in (b) shows the gain and phase of the filter as a function of frequency. Note that the phase starts at -90° due to the negative gain of the operational amplifier.

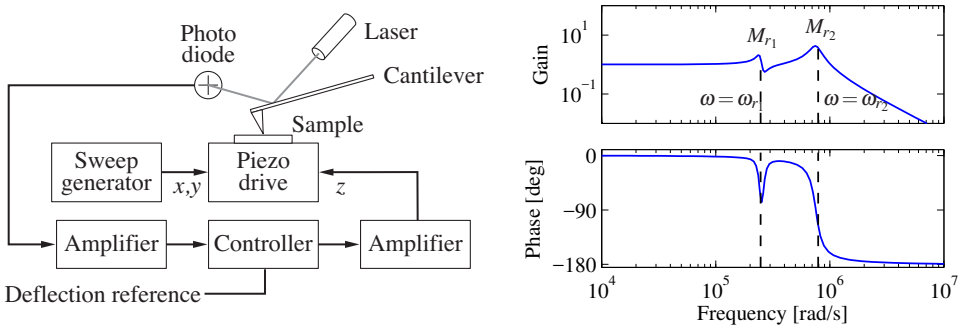
and $C_1 = C_2 = 100$ μF. We see that the circuit passes through signals with frequencies at about 10 rad/s, but attenuates frequencies below 5 rad/s and above 50 rad/s. At 0.1 rad/s the input signal is attenuated by 20× (0.05). This type of circuit is called a *band-pass filter* since it passes through signals in the band of frequencies between 5 and 50 rad/s. ∇

As in the case of the step response, a number of standard properties are defined for frequency responses. The gain of a system at $\omega = 0$ is called the *zero frequency gain* and corresponds to the ratio between a constant input and the steady output:

$$M_0 = -CA^{-1}B + D$$

(compare to equation (6.20)). The zero frequency gain is well defined only if A is invertible (i.e., if it does not have eigenvalues at 0). It is also important to note that the zero frequency gain is a relevant quantity only when a system is stable about the corresponding equilibrium point. So, if we apply a constant input $u = r$, then the corresponding equilibrium point $x_e = -A^{-1}Br$ must be stable in order to talk about the zero frequency gain. (In electrical engineering, the zero frequency gain is often called the *DC gain*. DC stands for direct current and reflects the common separation of signals in electrical engineering into a direct current (zero frequency) term and an alternating current (AC) term.)

The *bandwidth* ω_b of a system is the frequency range over which the gain has decreased by no more than a factor of $1/\sqrt{2}$ from its reference value. For systems with nonzero, finite zero frequency gain, the reference value is taken as the zero frequency gain. For systems that attenuate low frequencies but pass through high frequencies, the reference gain is taken as the high-frequency gain. For a system such as the band-pass filter in Example 6.8, bandwidth is defined as the range of frequencies where the gain is larger than $1/\sqrt{2}$ of the gain at the center of the band. (For Example 6.8 this would give a bandwidth of approximately 50 rad/s.)



(a) AFM block diagram (b) Frequency response

Figure 6.13: AFM frequency response. (a) A block diagram for the vertical dynamics of an atomic force microscope in contact mode. The plot in (b) shows the gain and phase for the piezo stack. The response contains two frequency peaks at resonances of the system, along with an antiresonance at $\omega = 268$ krad/s. The combination of a resonant peak followed by an antiresonance is common for systems with multiple lightly damped modes.

Other important properties of the frequency response are the *resonant peak* M_r, the largest value of the frequency response, and the *peak frequency* ω_{mr}, the frequency where the maximum occurs. These two properties describe the frequency of the sinusoidal input that produces the largest possible output and the gain at the frequency.

Example 6.9 Atomic force microscope in contact mode

Consider the model for the vertical dynamics of the atomic force microscope in contact mode, discussed in Section 4.5. The basic dynamics are given by equation (4.23). The piezo stack can be modeled by a second-order system with undamped natural frequency ω_3 and damping ratio ζ_3. The dynamics are then described by the linear system

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -k_2/(m_1+m_2) & -c_2/(m_1+m_2) & 1/m_2 & 0 \\ 0 & 0 & 0 & \omega_3 \\ 0 & 0 & -\omega_3 & -2\zeta_3\omega_3 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \omega_3 \end{pmatrix} u,$$

$$y = \frac{m_2}{m_1+m_2} \begin{pmatrix} \frac{m_1 k_2}{m_1+m_2} & \frac{m_1 c_2}{m_1+m_2} & 1 & 0 \end{pmatrix} x,$$

where the input signal is the drive signal to the amplifier and the output is the elongation of the piezo. The frequency response of the system is shown in Figure 6.13b. The zero frequency gain of the system is $M_0 = 1$. There are two resonant poles with peaks $M_{r1} = 2.12$ at $\omega_{mr1} = 238$ krad/s and $M_{r2} = 4.29$ at $\omega_{mr2} = 746$ krad/s. The bandwidth of the system, defined as the lowest frequency where the gain is $\sqrt{2}$ less than the zero frequency gain, is $\omega_b = 292$ krad/s. There is also a dip in the gain $M_d = 0.556$ for $\omega_{md} = 268$ krad/s. This dip, called an *antiresonance*, is associated with a dip in the phase and limits the performance when the system is controlled by simple controllers, as we will see in Chapter 11. ∇

Sampling

It is often convenient to use both differential and difference equations in modeling and control. For linear systems it is straightforward to transform from one to the other. Consider the general linear system described by equation (6.13) and assume that the control signal is constant over a sampling interval of constant length h. It follows from equation (6.14) of Theorem 6.4 that

$$x(t+h) = e^{Ah}x(t) + \int_t^{t+h} e^{A(t+h-\tau)}Bu(\tau)\,d\tau = \Phi x(t) + \Gamma u(t), \tag{6.26}$$

where we have assumed that the discontinuous control signal is continuous from the right. The behavior of the system at the sampling times $t = kh$ is described by the difference equation

$$x[k+1] = \Phi x[k] + \Gamma u[k], \qquad y[k] = Cx[k] + Du[k], \tag{6.27}$$

where

$$\Phi = e^{Ah}, \quad \Gamma = \Big(\int_0^h e^{As}\,ds\Big)B.$$

Notice that the difference equation (6.27) is an exact representation of the behavior of the system at the sampling instants. Similar expressions can also be obtained if the control signal is linear over the sampling interval.

The transformation from (6.26) to (6.27) is called *sampling*. The relations between the system matrices in the continuous and sampled representations are as follows:

$$A = \frac{1}{h}\log\Phi, \quad B = \Big(\int_0^h e^{As}\,ds\Big)^{-1}\Gamma. \tag{6.28}$$

Notice that if A is invertible, we have

$$\Gamma = A^{-1}\big(e^{Ah} - I\big)B.$$

All continuous-time systems can be sampled to obtain a discrete-time version, but there are discrete-time systems that do not have a continuous-time equivalent. The precise condition is that the matrix Φ cannot have real eigenvalues on the negative real axis.

Example 6.10 IBM Lotus server
In Example 3.4 we described how the dynamics of an IBM Lotus server were obtained as the discrete-time system

$$y[k+1] = ay[k] + bu[k],$$

where $a = 0.43$, $b = 0.47$ and the sampling period is $h = 60$ s. A differential equation model is needed if we would like to design control systems based on continuous-time theory. Such a model is obtained by applying equation (6.28); hence

$$A = \frac{\log a}{h} = -0.0141, \qquad B = \Big(\int_0^h e^{At}\,dt\Big)^{-1} b = 0.0116,$$

and we find that the difference equation can be interpreted as a sampled version of the ordinary differential equation

$$\frac{dx}{dt} = -0.0141x + 0.0116u.$$

∇

6.4 Linearization

As described at the beginning of the chapter, a common source of linear system models is through the approximation of a nonlinear system by a linear one. These approximations are aimed at studying the local behavior of a system, where the nonlinear effects are expected to be small. In this section we discuss how to locally approximate or convert a nonlinear system to a linear one, and what can be said about the stability of the original system in each case. We begin with an illustration of the basic concept using the cruise control example, which is described in more detail in Chapter 4.

Example 6.11 Cruise control
The dynamics for the cruise control system are derived in Section 4.1 and have the form

$$m\frac{dv}{dt} = \alpha_n u T(\alpha_n v) - mgC_r \operatorname{sgn}(v) - \frac{1}{2}\rho C_d A v^2 - mg\sin\theta, \tag{6.29}$$

where the first term on the right-hand side of the equation is the force generated by the engine and the remaining three terms are the rolling friction, aerodynamic drag and gravitational disturbance force. There is an equilibrium (v_e, u_e) when the force applied by the engine balances the disturbance forces.

To explore the behavior of the system near the equilibrium we will linearize the system. A Taylor series expansion of equation (6.29) around the equilibrium gives

$$\frac{d(v-v_e)}{dt} = a(v_e - v) - b_g(\theta - \theta_e) + b(u - u_e) + \text{higher-order terms}, \tag{6.30}$$

where

$$a = -\frac{u_e\alpha_n^2 T'(\alpha_n v_e) + \rho C_d A v_e}{m}, \qquad b_g = g\cos\theta_e, \qquad b = \frac{\alpha_n T(\alpha_n v_e)}{m}. \tag{6.31}$$

Notice that the term corresponding to rolling friction disappears if $v = 0$. For a car in fourth gear with $v_e = 20$ m/s, $\theta_e = 0$ and the numerical values for the car from Section 4.1, the equilibrium value for the throttle is $u_e = 0.1687$ and the parameters are $a = 0.0101$, $b = 1.32$ and $b_g = 9.8$. This linear model describes how small perturbations in the velocity about the nominal speed evolve in time.

Figure 6.14 shows a simulation of a cruise controller with linear and nonlinear models; the differences between the linear and nonlinear models are small, and hence the linearized model provides a reasonable approximation. ∇

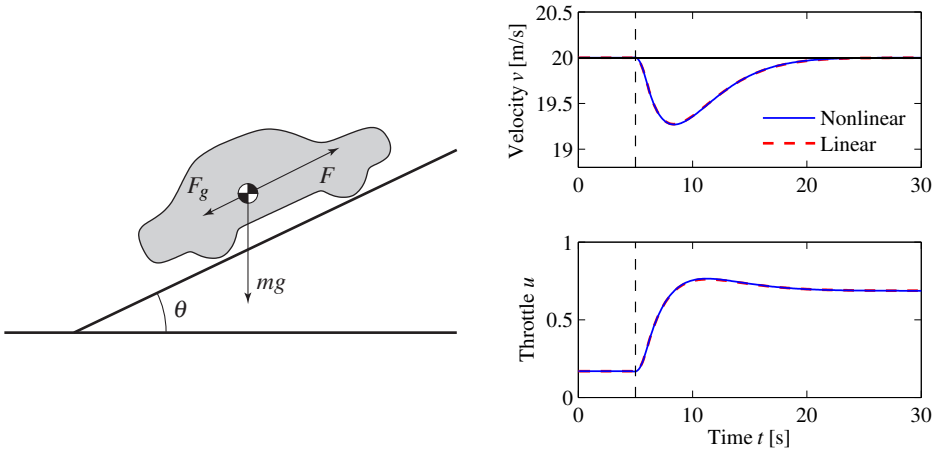


Figure 6.14: Simulated response of a vehicle with PI cruise control as it climbs a hill with a slope of 4°. The solid line is the simulation based on a nonlinear model, and the dashed line shows the corresponding simulation using a linear model. The controller gains are $k_p = 0.5$ and $k_i = 0.1$.

Jacobian Linearization Around an Equilibrium Point

To proceed more formally, consider a single-input, single-output nonlinear system

$$\begin{aligned} \frac{dx}{dt} &= f(x,u), \qquad x \in \mathbb{R}^n, u \in \mathbb{R}, \\ y &= h(x,u), \qquad y \in \mathbb{R}, \end{aligned} \tag{6.32}$$

with an equilibrium point at $x = x_e$, $u = u_e$. Without loss of generality we can assume that $x_e = 0$ and $u_e = 0$, although initially we will consider the general case to make the shift of coordinates explicit.

To study the *local* behavior of the system around the equilibrium point (x_e, u_e), we suppose that $x - x_e$ and $u - u_e$ are both small, so that nonlinear perturbations around this equilibrium point can be ignored compared with the (lower-order) linear terms. This is roughly the same type of argument that is used when we do small-angle approximations, replacing $\sin\theta$ with θ and $\cos\theta$ with 1 for θ near zero.

As we did in Chapter 5, we define a new set of state variables z, as well as inputs v and outputs w:

$$z = x - x_e, \qquad v = u - u_e, \qquad w = y - h(x_e, u_e).$$

These variables are all close to zero when we are near the equilibrium point, and so in these variables the nonlinear terms can be thought of as the higher-order terms in a Taylor series expansion of the relevant vector fields (assuming for now that these exist).

Formally, the *Jacobian linearization* of the nonlinear system (6.32) is

$$\frac{dz}{dt} = Az + Bv, \qquad w = Cz + Dv, \tag{6.33}$$

where

$$A = \left.\frac{\partial f}{\partial x}\right|_{(x_e,u_e)}, \quad B = \left.\frac{\partial f}{\partial u}\right|_{(x_e,u_e)}, \quad C = \left.\frac{\partial h}{\partial x}\right|_{(x_e,u_e)}, \quad D = \left.\frac{\partial h}{\partial u}\right|_{(x_e,u_e)}. \tag{6.34}$$

The system (6.33) approximates the original system (6.32) when we are near the equilibrium point about which the system was linearized. It follows from Theorem 5.3 that if the linearization is asymptotically stable, then the equilibrium point x_e is locally asymptotically stable for the full nonlinear system.

It is important to note that we can define the linearization of a system only near an equilibrium point. To see this, consider a polynomial system

$$\frac{dx}{dt} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + u,$$

where $a_0 \neq 0$. A set of equilibrium points for this system is given by $(x_e, u_e) = (x_e, -a_0 - a_1 x_e - a_2 x_e^2 - a_3 x_e^3)$, and we can linearize around any of them. Suppose that we try to linearize around the origin of the system $x = 0$, $u = 0$. If we drop the higher-order terms in x, then we get

$$\frac{dx}{dt} = a_0 + a_1 x + u,$$

which is *not* the Jacobian linearization if $a_0 \neq 0$. The constant term must be kept, and it is not present in (6.33). Furthermore, even if we kept the constant term in the approximate model, the system would quickly move away from this point (since it is "driven" by the constant term a_0), and hence the approximation could soon fail to hold.

Software for modeling and simulation frequently has facilities for performing linearization symbolically or numerically. The MATLAB command `trim` finds the equilibrium, and `linmod` extracts linear state space models from a SIMULINK system around an operating point.

Example 6.12 Vehicle steering

Consider the vehicle steering system introduced in Example 3.10. The nonlinear equations of motion for the system are given by equations (3.26)–(3.28) and can be written as

$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} v\cos(\alpha(\delta) + \theta) \\ v\sin(\alpha(\delta) + \theta) \\ \dfrac{v_0}{b}\tan\delta \end{pmatrix}, \qquad \alpha(\delta) = \arctan\left(\frac{a\tan\delta}{b}\right),$$

where x, y and θ are the position and orientation of the center of mass of the vehicle, v_0 is the velocity of the rear wheel, b is the distance between the front and rear wheels, δ is the angle of the front wheel and v is the velocity of the center of

mass. The function $\alpha(\delta)$ is the angle between the velocity vector and the vehicle's length axis.

We are interested in the motion of the vehicle about a straight-line path ($\theta = \theta_0$) with fixed velocity $v_0 \neq 0$. To find the relevant equilibrium point, we first set $\dot{\theta} = 0$ and we see that we must have $\delta = 0$, corresponding to the steering wheel being straight. This also yields $\alpha = 0$. Looking at the first two equations in the dynamics, we see that the motion in the xy direction is by definition *not* at equilibrium since $\dot{x}^2 + \dot{y}^2 = v^2 \neq 0$. Therefore we cannot formally linearize the full model.

Suppose instead that we are concerned with the lateral deviation of the vehicle from a straight line. For simplicity, we let $\theta_e = 0$, which corresponds to driving along the x axis. We can then focus on the equations of motion in the y and θ directions. With some abuse of notation we introduce the state $x = (y, \theta)$ and $u = \delta$. The system is then in standard form with

$$f(x,u) = \begin{pmatrix} v \sin(\alpha(u) + x_2) \\ \dfrac{v_0}{b} \tan u \end{pmatrix}, \qquad \alpha(u) = \arctan\Big(\frac{a \tan u}{b}\Big), \qquad h(x,u) = x_1.$$

The equilibrium point of interest is given by $x = (0,0)$ and $u = 0$. To compute the linearized model around this equilibrium point, we make use of the formulas (6.34). A straightforward calculation yields

$$A = \left.\frac{\partial f}{\partial x}\right|_{\substack{x=0\\u=0}} = \begin{pmatrix} 0 & v_0 \\ 0 & 0 \end{pmatrix}, \qquad B = \left.\frac{\partial f}{\partial u}\right|_{\substack{x=0\\u=0}} = \begin{pmatrix} a v_0/b \\ v_0/b \end{pmatrix},$$
$$C = \left.\frac{\partial h}{\partial x}\right|_{\substack{x=0\\u=0}} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \qquad D = \left.\frac{\partial h}{\partial u}\right|_{\substack{x=0\\u=0}} = 0,$$

and the linearized system

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du \tag{6.35}$$

thus provides an approximation to the original nonlinear dynamics.

The linearized model can be simplified further by introducing normalized variables, as discussed in Section 3.3. For this system, we choose the wheel base b as the length unit and the time unit as the time required to travel a wheel base. The normalized state is thus $z = (x_1/b, x_2)$, and the new time variable is $\tau = v_0 t/b$. The model (6.35) then becomes

$$\frac{dz}{d\tau} = \begin{pmatrix} z_2 + \gamma u \\ u \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} z + \begin{pmatrix} \gamma \\ 1 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} z, \tag{6.36}$$

where $\gamma = a/b$. The normalized linear model for vehicle steering with nonslipping wheels is thus a linear system with only one parameter. ∇

Feedback Linearization

Another type of linearization is the use of feedback to convert the dynamics of a nonlinear system into those of a linear one. We illustrate the basic idea with an example.

Example 6.13 Cruise control
Consider again the cruise control system from Example 6.11, whose dynamics are given in equation (6.29):

$$m\frac{dv}{dt} = \alpha_n u T(\alpha_n v) - mgC_r \operatorname{sgn}(v) - \frac{1}{2}\rho C_d A v^2 - mg\sin\theta.$$

If we choose u as a feedback law of the form

$$u = \frac{1}{\alpha_n T(\alpha_n v)}\left(u' + mgC_r \operatorname{sgn}(v) + \frac{1}{2}\rho C_d A v^2\right), \tag{6.37}$$

then the resulting dynamics become

$$m\frac{dv}{dt} = u' + d, \tag{6.38}$$

where $d(t) = -mg\sin\theta(t)$ is the disturbance force due the slope of the road (which may be changing as we drive). If we now define a feedback law for u' (such as a proportional-integral-derivative [PID] controller), we can use equation (6.37) to compute the final input that should be commanded.

Equation (6.38) is a linear differential equation. We have essentially "inverted" the nonlinearity through the use of the feedback law (6.37). This requires that we have an accurate measurement of the vehicle velocity v as well as an accurate model of the torque characteristics of the engine, gear ratios, drag and friction characteristics and mass of the car. While such a model is not generally available (remembering that the parameter values can change), if we design a good feedback law for u', then we can achieve robustness to these uncertainties. ∇

More generally, we say that a system of the form

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x),$$

is *feedback linearizable* if there exists a control law $u = \alpha(x,v)$ such that the resulting closed loop system is input/output linear with input v and output y, as shown in Figure 6.15. To fully characterize such systems is beyond the scope of this text, but we note that in addition to changes in the input, the general theory also allows for (nonlinear) changes in the states that are used to describe the system, keeping only the input and output variables fixed. More details of this process can be found in the textbooks by Isidori [Isi95] and Khalil [Kha01].

One case that comes up relatively frequently, and is hence worth special mention, is the set of mechanical systems of the form

$$M(q)\ddot{q} + C(q,\dot{q}) = B(q)u.$$

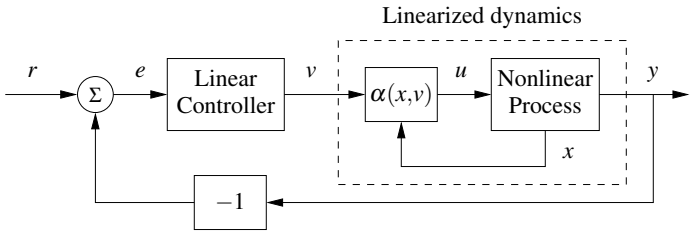


Figure 6.15: Feedback linearization. A nonlinear feedback of the form $u = \alpha(x, v)$ is used to modify the dynamics of a nonlinear process so that the response from the input v to the output y is linear. A linear controller can then be used to regulate the system's dynamics.

Here $q \in \mathbb{R}^n$ is the configuration of the mechanical system, $M(q) \in \mathbb{R}^{n\times n}$ is the configuration-dependent inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^n$ represents the Coriolis forces and additional nonlinear forces (such as stiffness and friction) and $B(q) \in \mathbb{R}^{n\times p}$ is the input matrix. If $p = n$, then we have the same number of inputs and configuration variables, and if we further have that $B(q)$ is an invertible matrix for all configurations q, then we can choose

$$u = B^{-1}(q)\big(M(q)v + C(q,\dot{q})\big). \tag{6.39}$$

The resulting dynamics become

$$M(q)\ddot{q} = M(q)v \qquad \Longrightarrow \qquad \ddot{q} = v,$$

which is a linear system. We can now use the tools of linear system theory to analyze and design control laws for the linearized system, remembering to apply equation (6.39) to obtain the actual input that will be applied to the system.

This type of control is common in robotics, where it goes by the name of *computed torque*, and in aircraft flight control, where it is called *dynamic inversion*. Some modeling tools like Modelica can generate the code for the inverse model automatically. One caution is that feedback linearization can often cancel out beneficial terms in the natural dynamics, and hence it must be used with care. Extensions that do not require complete cancellation of nonlinearities are discussed in Khalil [Kha01] and Krstić et al. [KKK95].

6.5 Further Reading

The majority of the material in this chapter is classical and can be found in most books on dynamics and control theory, including early works on control such as James, Nichols and Phillips [JNP47] and more recent textbooks such as Dorf and Bishop [DB04], Franklin, Powell and Emami-Naeini [FPEN05] and Ogata [Oga01]. An excellent presentation of linear systems based on the matrix exponential is given in the book by Brockett [Bro70], a more comprehensive treatment is given by Rugh [Rug95] and an elegant mathematical treatment is given in Sontag [Son98]. Material on feedback linearization can be found in books on nonlinear control theory such as Isidori [Isi95] and Khalil [Kha01]. The idea of characterizing dynamics

by considering the responses to step inputs is due to Heaviside, he also introduced an operator calculus to analyze linear systems. The unit step is therefore also called the *Heaviside step function*. Analysis of linear systems was simplified significantly, but Heaviside's work was heavily criticized because of lack of mathematical rigor, as described in the biography by Nahin [Nah88]. The difficulties were cleared up later by the mathematician Laurent Schwartz who developed *distribution theory* in the late 1940s. In engineering, linear systems have traditionally been analyzed using Laplace transforms as described in Gardner and Barnes [GB42]. Use of the matrix exponential started with developments of control theory in the 1960s, strongly stimulated by a textbook by Zadeh and Desoer [ZD63]. Use of matrix techniques expanded rapidly when the powerful methods of numeric linear algebra were packaged in programs like LabVIEW, MATLAB and Mathematica.

Exercises

6.1 (Response to the derivative of a signal) Show that if $y(t)$ is the output of a linear system corresponding to input $u(t)$, then the output corresponding to an input $\dot{u}(t)$ is given by $\dot{y}(t)$. (Hint: Use the definition of the derivative: $\dot{y}(t) = \lim_{\varepsilon \to 0} \big(y(t+\varepsilon) - y(t)\big)/\varepsilon$.)

6.2 (Impulse response and convolution) Show that a signal $u(t)$ can be decomposed in terms of the impulse function $\delta(t)$ as

$$u(t) = \int_0^t \delta(t-\tau)u(\tau)\,d\tau$$

and use this decomposition plus the principle of superposition to show that the response of a linear, time-invariant system to an input $u(t)$ (assuming a zero initial condition) can be written as

$$y(t) = \int_0^t h(t-\tau)u(\tau)\,d\tau,$$

where $h(t)$ is the impulse response of the system.

6.3 (Pulse response for a compartment model) Consider the compartment model given in Example 6.7. Compute the step response for the system and compare it with Figure 6.10b. Use the principle of superposition to compute the response to the 5 s pulse input shown in Figure 6.10c. Use the parameter values $k_0 = 0.1$, $k_1 = 0.1$, $k_2 = 0.5$ and $b_0 = 1.5$.

6.4 (Matrix exponential for second-order system) Assume that $\zeta > 1$ and let $\omega_d = \omega_0\sqrt{\zeta^2 - 1}$. Show that

$$\exp\begin{pmatrix} -\zeta\omega_0 & \omega_d \\ \omega_d & -\zeta\omega_0 \end{pmatrix} t = e^{-\zeta\omega_0 t}\begin{pmatrix} \cos\omega_d t & \sin\omega_d t \\ \sin\omega_d t & \cos\omega_d t \end{pmatrix}.$$

Also show that

$$\exp\begin{pmatrix} -\omega_0 & \omega_0 \\ 0 & -\omega_0 \end{pmatrix} t = e^{-\omega_0 t}\begin{pmatrix} 1 & \omega_0 t \\ 0 & 1 \end{pmatrix}.$$

6.5 (Lyapunov function for a linear system) Consider a linear system $\dot{x} = Ax$ with $\operatorname{Re}\lambda_j < 0$ for all eigenvalues λ_j of the matrix A. Show that the matrix

$$P = \int_0^\infty e^{A^T\tau} Q e^{A\tau}\, d\tau$$

defines a Lyapunov function of the form $V(x) = x^T P x$.

6.6 (Nondiagonal Jordan form) Consider a linear system with a Jordan form that is non-diagonal.

(a) Prove Proposition 6.3 by showing that if the system contains a real eigenvalue $\lambda = 0$ with a nontrivial Jordan block, then there exists an initial condition with a solution that grows in time.

(b) Extend this argument to the case of complex eigenvalues with $\operatorname{Re}\lambda = 0$ by using the block Jordan form

$$J_i = \begin{pmatrix} 0 & \omega & 1 & 0 \\ -\omega & 0 & 0 & 1 \\ 0 & 0 & 0 & \omega \\ 0 & 0 & -\omega & 0 \end{pmatrix}.$$

6.7 (Rise time for a first-order system) Consider a first-order system of the form

$$\tau\frac{dx}{dt} = -x + u, \qquad y = x.$$

We say that the parameter τ is the *time constant* for the system since the zero input system approaches the origin as $e^{-t/\tau}$. For a first-order system of this form, show that the rise time for a step response of the system is approximately 2τ, and that 1%, 2%, and 5% settling times approximately corresponds to 4.6τ, 4τ and 3τ.

6.8 (Discrete-time systems) Consider a linear discrete-time system of the form

$$x[k+1] = Ax[k] + Bu[k], \qquad y[k] = Cx[k] + Du[k].$$

(a) Show that the general form of the output of a discrete-time linear system is given by the discrete-time convolution equation:

$$y[k] = CA^k x[0] + \sum_{j=0}^{k-1} CA^{k-j-1}Bu[j] + Du[k].$$

(b) Show that a discrete-time linear system is asymptotically stable if and only if all the eigenvalues of A have a magnitude strictly less than 1.

(c) Let $u[k] = \sin(\omega k)$ represent an oscillatory input with frequency $\omega < \pi$ (to avoid "aliasing"). Show that the steady-state component of the response has gain M and phase θ, where

$$Me^{i\theta} = C(e^{i\omega}I - A)^{-1}B + D.$$

(d) Show that if we have a nonlinear discrete-time system

$$\begin{aligned} x[k] &= f(x[k], u[k]), \qquad x[k] \in \mathbb{R}^n, u \in \mathbb{R}, \\ y[k] &= h(x[k], u[k]), \qquad y \in \mathbb{R}, \end{aligned}$$

then we can linearize the system around an equilibrium point (x_e, u_e) by defining the matrices A, B, C and D as in equation (6.34).

6.9 (Keynesian economics) Consider the following simple Keynesian macroeconomic model in the form of a linear discrete-time system discussed in Exercise 6.8:

$$\begin{aligned} \begin{pmatrix} C[t+1] \\ I[t+1] \end{pmatrix} &= \begin{pmatrix} a & a \\ ab-b & ab \end{pmatrix} \begin{pmatrix} C[t] \\ I[t] \end{pmatrix} + \begin{pmatrix} a \\ ab \end{pmatrix} G[t], \\ Y[t] &= C[t] + I[t] + G[t]. \end{aligned}$$

Determine the eigenvalues of the dynamics matrix. When are the magnitudes of the eigenvalues less than 1? Assume that the system is in equilibrium with constant values capital spending C, investment I and government expenditure G. Explore what happens when government expenditure increases by 10%. Use the values $a = 0.25$ and $b = 0.5$.

6.10 Consider a scalar system

$$\frac{dx}{dt} = 1 - x^3 + u.$$

Compute the equilibrium points for the unforced system ($u = 0$) and use a Taylor series expansion around the equilibrium point to compute the linearization. Verify that this agrees with the linearization in equation (6.33).

6.11 (Transcriptional regulation) Consider the dynamics of a genetic circuit that implements *self-repression*: the protein produced by a gene is a repressor for that gene, thus restricting its own production. Using the models presented in Example 3.15, the dynamics for the system can be written as

$$\frac{dm}{dt} = \frac{\alpha}{1 + kp^2} + \alpha_0 - \gamma m - u, \qquad \frac{dp}{dt} = \beta m - \delta p, \tag{6.40}$$

where u is a disturbance term that affects RNA transcription and $m, p \geq 0$. Find the equilibrium points for the system and use the linearized dynamics around each equilibrium point to determine the local stability of the equilibrium point and the step response of the system to a disturbance.

Chapter Seven
State Feedback

> *Intuitively, the state may be regarded as a kind of information storage or memory or accumulation of past causes. We must, of course, demand that the set of internal states* Σ *be sufficiently rich to carry all information about the past history of* Σ *to predict the effect of the past upon the future. We do not insist, however, that the state is the* least *such information although this is often a convenient assumption.*
>
> R. E. Kalman, P. L. Falb and M. A. Arbib, *Topics in Mathematical System Theory*, 1969 [KFA69].

This chapter describes how the feedback of a system's state can be used to shape the local behavior of a system. The concept of reachability is introduced and used to investigate how to design the dynamics of a system through assignment of its eigenvalues. In particular, it will be shown that under certain conditions it is possible to assign the system eigenvalues arbitrarily by appropriate feedback of the system state.

7.1 Reachability

One of the fundamental properties of a control system is what set of points in the state space can be reached through the choice of a control input. It turns out that the property of reachability is also fundamental in understanding the extent to which feedback can be used to design the dynamics of a system.

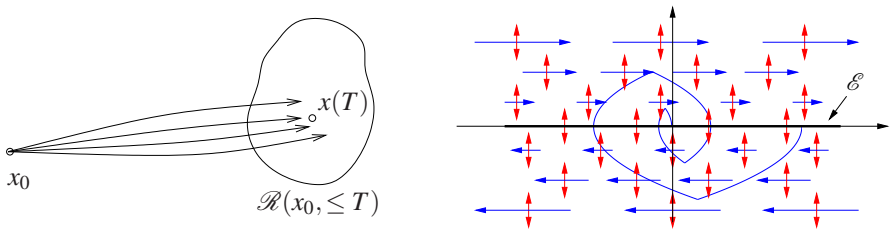
Definition of Reachability

We begin by disregarding the output measurements of the system and focusing on the evolution of the state, given by

$$\frac{dx}{dt} = Ax + Bu, \tag{7.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$, A is an $n \times n$ matrix and B a column vector. A fundamental question is whether it is possible to find control signals so that any point in the state space can be reached through some choice of input. To study this, we define the *reachable set* $\mathcal{R}(x_0, \leq T)$ as the set of all points x_f such that there exists an input $u(t)$, $0 \leq t \leq T$ that steers the system from $x(0) = x_0$ to $x(T) = x_f$, as illustrated in Figure 7.1a.

Definition 7.1 (Reachability)**.** A linear system is *reachable* if for any $x_0, x_f \in \mathbb{R}^n$ there exists a $T > 0$ and $u\colon [0,T] \to \mathbb{R}$ such that if $x(0) = x_0$ then the corresponding solution satisfies $x(T) = x_f$.



(a) Reachable set (b) Reachability through control

Figure 7.1: The reachable set for a control system. The set $\mathscr{R}(x_0, \leq T)$ shown in (a) is the set of points reachable from x_0 in time less than T. The phase portrait in (b) shows the dynamics for a double integrator, with the natural dynamics drawn as horizontal arrows and the control inputs drawn as vertical arrows. The set of achievable equilibrium points is the x axis. By setting the control inputs as a function of the state, it is possible to steer the system to the origin, as shown on the sample path.

The definition of reachability addresses whether it is possible to reach all points in the state space in a *transient* fashion. In many applications, the set of points that we are most interested in reaching is the set of equilibrium points of the system (since we can remain at those points with constant input u). The set of all possible equilibria for constant controls is given by

$$\mathscr{E} = \{x_e : Ax_e + Bu_e = 0 \text{ for some } u_e \in \mathbb{R}\}.$$

This means that possible equilibria lie in a one- (or possibly higher) dimensional subspace. If the matrix A is invertible, this subspace is one-dimensional and is spanned by $A^{-1}B$.

The following example provides some insight into the possibilities.

Example 7.1 Double integrator

Consider a linear system consisting of a double integrator whose dynamics are given by

$$\frac{dx_1}{dt} = x_2, \qquad \frac{dx_2}{dt} = u.$$

Figure 7.1b shows a phase portrait of the system. The open loop dynamics ($u = 0$) are shown as horizontal arrows pointed to the right for $x_2 > 0$ and to the left for $x_2 < 0$. The control input is represented by a double-headed arrow in the vertical direction, corresponding to our ability to set the value of $\dot{x}_2$. The set of equilibrium points $\mathscr{E}$ corresponds to the x_1 axis, with $u_e = 0$.

Suppose first that we wish to reach the origin from an initial condition $(a, 0)$. We can directly move the state up and down in the phase plane, but we must rely on the natural dynamics to control the motion to the left and right. If $a > 0$, we can move toward the origin by first setting $u < 0$, which will cause x_2 to become negative. Once $x_2 < 0$, the value of x_1 will begin to decrease and we will move to the left. After a while, we can set u_2 to be positive, moving x_2 back toward zero

and slowing the motion in the x_1 direction. If we bring x_2 to a positive value, we can move the system state in the opposite direction.

Figure 7.1b shows a sample trajectory bringing the system to the origin. Note that if we steer the system to an equilibrium point, it is possible to remain there indefinitely (since $\dot{x}_1 = 0$ when $x_2 = 0$), but if we go to any other point in the state space, we can pass through the point only in a transient fashion. ∇

To find general conditions under which a linear system is reachable, we will first give a heuristic argument based on formal calculations with impulse functions. We note that if we can reach all points in the state space through some choice of input, then we can also reach all equilibrium points.

Testing for Reachability

When the initial state is zero, the response of the system (7.1) to an input $u(t)$ is given by

$$x(t) = \int_0^t e^{A(t-\tau)} B u(\tau)\, d\tau. \tag{7.2}$$

If we choose the input to be a impulse function $\delta(t)$ as defined in Section 6.3, the state becomes

$$x_\delta = \int_0^t e^{A(t-\tau)} B \delta(\tau)\, d\tau = e^{At} B.$$

(Note that the state changes instantaneously in response to the impulse.) We can find the response to the derivative of an impulse function by taking the derivative of the impulse response (Exercise 6.1):

$$x_{\dot{\delta}} = \frac{dx_\delta}{dt} = A e^{At} B.$$

Continuing this process and using the linearity of the system, the input

$$u(t) = \alpha_1 \delta(t) + \alpha_2 \dot{\delta}(t) + \alpha_3 \ddot{\delta}(t) + \cdots + \alpha_n \delta^{(n-1)}(t)$$

gives the state

$$x(t) = \alpha_1 e^{At} B + \alpha_2 A e^{At} B + \alpha_3 A^2 e^{At} B + \cdots + \alpha_n A^{n-1} e^{At} B.$$

Taking the limit as t goes to zero through positive values, we get

$$\lim_{t \to 0+} x(t) = \alpha_1 B + \alpha_2 AB + \alpha_3 A^2 B + \cdots + \alpha_n A^{n-1} B.$$

On the right is a linear combination of the columns of the matrix

$$W_r = \begin{pmatrix} B & AB & \cdots & A^{n-1}B \end{pmatrix}. \tag{7.3}$$

To reach an arbitrary point in the state space, we thus require that W_r is nonsingular (and hence invertible). The matrix W_r is called the *reachability matrix*.

Although we have only considered the scalar input case, it turns out that this same test works in the multi-input case, where we require that W_r be full rank (have n linearly independent columns). In addition, it can be shown that only the terms

up to $A^{n-1}B$ must be computed; additional terms add no new directions to W_r (see Exercise 7.10).

An input consisting of a sum of impulse functions and their derivatives is a very violent signal. To see that an arbitrary point can be reached with smoother signals we can make use of the convolution equation. Assuming that the initial condition is zero, the state of a linear system is given by

$$x(t) = \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau = \int_0^t e^{A\tau}Bu(t-\tau)d\tau.$$

It follows from the theory of matrix functions, specifically the Cayley–Hamilton theorem (see Exercise 7.10), that

$$e^{A\tau} = I\alpha_0(\tau) + A\alpha_1(\tau) + \cdots + A^{n-1}\alpha_{n-1}(\tau),$$

where $\alpha_i(\tau)$ are scalar functions, and we find that

$$\begin{aligned} x(t) = B\int_0^t \alpha_0(\tau)u(t-\tau)\,d\tau + AB\int_0^t \alpha_1(\tau)u(t-\tau)\,d\tau \\ + \cdots + A^{n-1}B\int_0^t \alpha_{n-1}(\tau)u(t-\tau)\,d\tau. \end{aligned}$$

Again we observe that the right-hand side is a linear combination of the columns of the reachability matrix W_r given by equation (7.3). This basic approach leads to the following theorem.

Theorem 7.1 (Reachability rank condition)**.** *A linear system of the form* (7.1) *is reachable if and only if the reachability matrix W_r is invertible (full rank).*

The formal proof of this theorem is beyond the scope of this text but follows along the lines of the sketch above and can be found in most books on linear control theory, such as Callier and Desoer [CD91] or Lewis [Lew03]. It is also interesting to note that Theorem 7.1 makes no mention of the time T that was in our definition of reachability. For a linear system, it turns out that we can find an input taking x_0 to x_f for *any* $T > 0$, though the size of the input required can be very large when T is very small.

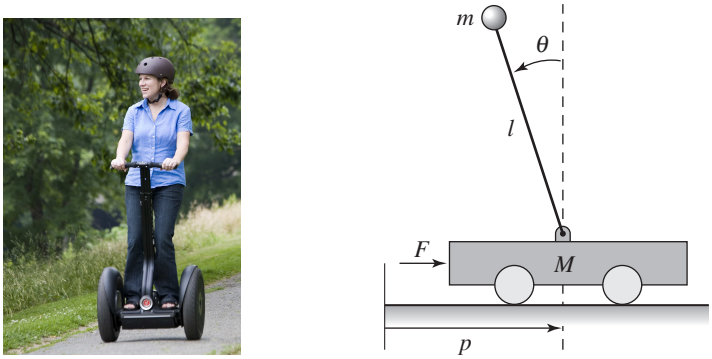
We illustrate the concept of reachability with the following example.

Example 7.2 Balance system

Consider the balance system introduced in Example 3.1 and shown in Figure 7.2. Recall that this system is a model for a class of examples in which the center of mass is balanced above a pivot point. One example is the Segway Personal Transporter shown in Figure 7.2a, about which a natural question to ask is whether we can move from one stationary point to another by appropriate application of forces through the wheels.

The nonlinear equations of motion for the system are given in equation (3.10) and repeated here:

$$\begin{aligned} (M+m)\ddot{p} - ml\cos\theta\,\ddot{\theta} &= -c\dot{p} - ml\sin\theta\,\dot{\theta}^2 + F, \\ (J+ml^2)\ddot{\theta} - ml\cos\theta\,\ddot{p} &= -\gamma\dot{\theta} + mgl\sin\theta. \end{aligned} \tag{7.4}$$



(a) Segway (b) Cart-pendulum system

Figure 7.2: Balance system. The Segway Personal Transporter shown in (a) is an example of a balance system that uses torque applied to the wheels to keep the rider upright. A simplified diagram for a balance system is shown in (b). The system consists of a mass m on a rod of length l connected by a pivot to a cart with mass M.

For simplicity, we take $c = \gamma = 0$. Linearizing around the equilibrium point $x_e = (p,0,0,0)$, the dynamics matrix and the control matrix are

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & m^2l^2g/\mu & 0 & 0 \\ 0 & M_t mgl/\mu & 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 0 \\ J_t/\mu \\ lm/\mu \end{pmatrix},$$

where $\mu = M_t J_t - m^2l^2$, $M_t = M + m$ and $J_t = J + ml^2$. The reachability matrix is

$$W_r = \begin{pmatrix} 0 & J_t/\mu & 0 & gl^3m^3/\mu^2 \\ 0 & lm/\mu & 0 & gl^2m^2(m+M)/\mu^2 \\ J_t/\mu & 0 & gl^3m^3/\mu^2 & 0 \\ lm/\mu & 0 & gl^2m^2(m+M)/\mu^2 & 0 \end{pmatrix}. \tag{7.5}$$

The determinant of this matrix is

$$\det(W_r) = \frac{g^2l^4m^4}{(\mu)^4} \neq 0,$$

and we can conclude that the system is reachable. This implies that we can move the system from any initial state to any final state and, in particular, that we can always find an input to bring the system from an initial state to an equilibrium point. ∇

It is useful to have an intuitive understanding of the mechanisms that make a system unreachable. An example of such a system is given in Figure 7.3. The system consists of two identical systems with the same input. We cannot separately cause the first and the second systems to do something different since they have the same input. Hence we cannot reach arbitrary states, and so the system is not

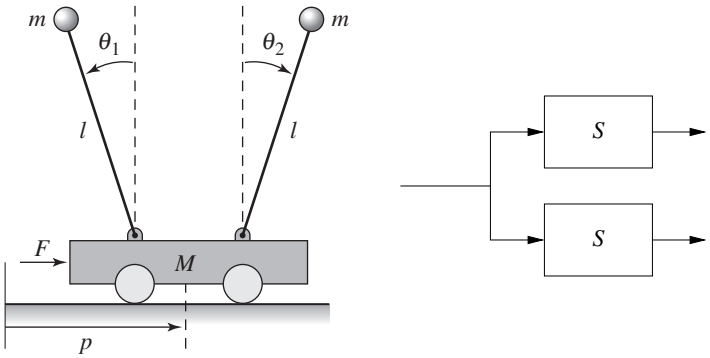


Figure 7.3: An unreachable system. The cart–pendulum system shown on the left has a single input that affects two pendula of equal length and mass. Since the forces affecting the two pendula are the same and their dynamics are identical, it is not possible to arbitrarily control the state of the system. The figure on the right is a block diagram representation of this situation.

reachable (Exercise 7.3).

More subtle mechanisms for nonreachability can also occur. For example, if there is a linear combination of states that always remains constant, then the system is not reachable. To see this, suppose that there exists a row vector H such that

$$0 = \frac{d}{dt}Hx = H(Ax + Bu), \quad \text{for all } x \text{ and } u.$$

Then H is in the left null space of both A and B and it follows that

$$HW_r = H \begin{pmatrix} B & AB & \cdots & A^{n-1}B \end{pmatrix} = 0.$$

Hence the reachability matrix is not full rank. In this case, if we have an initial condition x_0 and we wish to reach a state x_f for which $Hx_0 \neq Hx_f$, then since $Hx(t)$ is constant, no input u can move the state from x_0 to x_f.

Reachable Canonical Form

As we have already seen in previous chapters, it is often convenient to change coordinates and write the dynamics of the system in the transformed coordinates $z = Tx$. One application of a change of coordinates is to convert a system into a canonical form in which it is easy to perform certain types of analysis.

A linear state space system is in *reachable canonical form* if its dynamics are

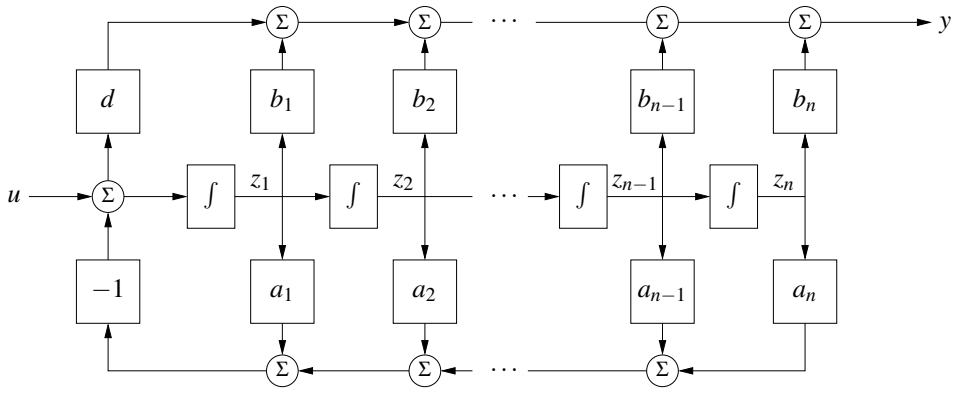


Figure 7.4: Block diagram for a system in reachable canonical form. The individual states of the system are represented by a chain of integrators whose input depends on the weighted values of the states. The output is given by an appropriate combination of the system input and other states.

given by

$$\frac{dz}{dt} = \begin{pmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u, \tag{7.6}$$

$$y = \begin{pmatrix} b_1 & b_2 & b_3 & \dots & b_n \end{pmatrix} z + du.$$

A block diagram for a system in reachable canonical form is shown in Figure 7.4. We see that the coefficients that appear in the A and B matrices show up directly in the block diagram. Furthermore, the output of the system is a simple linear combination of the outputs of the integration blocks.

The characteristic polynomial for a system in reachable canonical form is given by

$$\lambda(s) = s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n. \tag{7.7}$$

The reachability matrix also has a relatively simple structure:

$$\tilde{W}_r = \begin{pmatrix} \tilde{B} & \tilde{A}\tilde{B} & \dots & \tilde{A}^{n-1}\tilde{B} \end{pmatrix} = \begin{pmatrix} 1 & -a_1 & a_1^2 - a_2 & \cdots & * \\ 0 & 1 & -a_1 & \cdots & * \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & * \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix},$$

where $*$ indicates a possibly nonzero term and we use a tilde to remind us that A and B are in a special form. The matrix W_r is full rank since no column can be written as a linear combination of the others because of the triangular structure of the matrix.

We now consider the problem of finding a change of coordinates such that the dynamics of a system can be written in reachable canonical form. Let A, B represent the dynamics of a given system and $\tilde{A}, \tilde{B}$ be the dynamics in reachable canonical form. Suppose that we wish to transform the original system into reachable canonical form using a coordinate transformation $z = Tx$. As shown in the previous chapter, the dynamics matrix and the control matrix for the transformed system are

$$\tilde{A} = TAT^{-1}, \qquad \tilde{B} = TB.$$

The reachability matrix for the transformed system then becomes

$$\tilde{W}_r = \begin{pmatrix} \tilde{B} & \tilde{A}\tilde{B} & \cdots & \tilde{A}^{n-1}\tilde{B} \end{pmatrix}.$$

Transforming each element individually, we have

$$\begin{aligned} \tilde{A}\tilde{B} &= TAT^{-1}TB = TAB, \\ \tilde{A}^2\tilde{B} &= (TAT^{-1})^2 TB = TAT^{-1}TAT^{-1}TB = TA^2B, \\ &\vdots \\ \tilde{A}^n\tilde{B} &= TA^nB, \end{aligned}$$

and hence the reachability matrix for the transformed system is

$$\tilde{W}_r = T \begin{pmatrix} B & AB & \cdots & A^{n-1}B \end{pmatrix} = TW_r. \tag{7.8}$$

If W_r is invertible, we can thus solve for the transformation T that takes the system into reachable canonical form:

$$T = \tilde{W}_r W_r^{-1}.$$

The following example illustrates the approach.

Example 7.3 Transformation to reachable form
Consider a simple two-dimensional system of the form

$$\frac{dx}{dt} = \begin{pmatrix} \alpha & \omega \\ -\omega & \alpha \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u.$$

We wish to find the transformation that converts the system into reachable canonical form:

$$\tilde{A} = \begin{pmatrix} -a_1 & -a_2 \\ 1 & 0 \end{pmatrix}, \qquad \tilde{B} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The coefficients a_1 and a_2 can be determined from the characteristic polynomial for the original system:

$$\lambda(s) = \det(sI - A) = s^2 - 2\alpha s + (\alpha^2 + \omega^2) \quad \Longrightarrow \quad \begin{aligned} a_1 &= -2\alpha, \\ a_2 &= \alpha^2 + \omega^2. \end{aligned}$$

The reachability matrix for each system is

$$W_r = \begin{pmatrix} 0 & \omega \\ 1 & \alpha \end{pmatrix}, \qquad \tilde{W}_r = \begin{pmatrix} 1 & -a_1 \\ 0 & 1 \end{pmatrix}.$$

The transformation T becomes

$$T = \tilde{W}_r W_r^{-1} = \begin{pmatrix} -(a_1+\alpha)/\omega & 1 \\ 1/\omega & 0 \end{pmatrix} = \begin{pmatrix} \alpha/\omega & 1 \\ 1/\omega & 0 \end{pmatrix},$$

and hence the coordinates

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = Tx = \begin{pmatrix} \alpha x_1/\omega + x_2 \\ x_1/\omega \end{pmatrix}$$

put the system in reachable canonical form. ∇

We summarize the results of this section in the following theorem.

Theorem 7.2 (Reachable canonical form)**.** *Let A and B be the dynamics and control matrices for a reachable system and suppose that the characteristic polynomial for A is given by*

$$\det(sI - A) = s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n.$$

Then there exists a transformation $z = Tx$ such that in the transformed coordinates the dynamics and control matrices are in reachable canonical form (7.6)*.*

One important implication of this theorem is that for any reachable system, we can assume without loss of generality that the coordinates are chosen such that the system is in reachable canonical form. This is particularly useful for proofs, as we shall see later in this chapter. However, for high-order systems, small changes in the coefficients a_i can give large changes in the eigenvalues. Hence, the reachable canonical form is not always well conditioned and must be used with some care.

7.2 Stabilization by State Feedback

The state of a dynamical system is a collection of variables that permits prediction of the future evolution of a system given its future inputs. We now explore the idea of designing the dynamics of a system through feedback of the state. We will assume that the system to be controlled is described by a linear state model and has a single input (for simplicity). The feedback control law will be developed step by step using a single idea: the positioning of closed loop eigenvalues in desired locations.

State Space Controller Structure

Figure 7.5 is a diagram of a typical control system using state feedback. The full system consists of the process dynamics, which we take to be linear, the controller

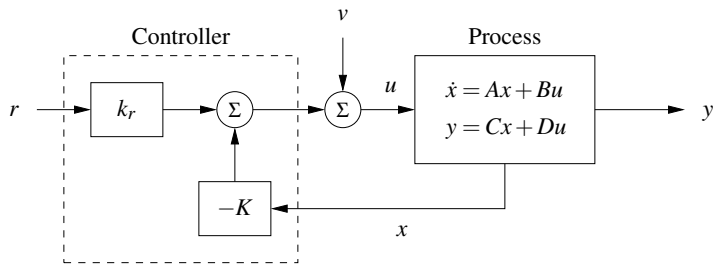


Figure 7.5: A feedback control system with state feedback. The controller uses the system state x and the reference input r to command the process through its input u. We model disturbances via the additive input v.

elements K and k_r, the reference input (or command signal) r and process disturbances v. The goal of the feedback controller is to regulate the output of the system y such that it tracks the reference input in the presence of disturbances and also uncertainty in the process dynamics.

An important element of the control design is the performance specification. The simplest performance specification is that of stability: given a constant reference r and in the absence of any disturbances, we would like the equilibrium point of the system to be asymptotically stable. More sophisticated performance specifications typically involve giving desired properties of the step or frequency response of the system, such as specifying the desired rise time, overshoot and settling time of the step response. Finally, we are often concerned with the disturbance attenuation properties of the system: to what extent can we experience disturbance inputs v and still hold the output y near the desired value?

Consider a system described by the linear differential equation

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du, \tag{7.9}$$

where we have ignored the disturbance signal v for now. Our goal is to drive the output y to a given reference value r and hold it there.

We begin by assuming that all components of the state vector are measured. Since the state at time t contains all the information necessary to predict the future behavior of the system, the most general time-invariant control law is a function of the state and the reference input:

$$u = \alpha(x, r).$$

If the control law is restricted to be linear, it can be written as

$$u = -Kx + k_r r, \tag{7.10}$$

where r is the reference value, assumed for now to be a constant.

This control law corresponds to the structure shown in Figure 7.5. The negative sign is a convention to indicate that negative feedback is the normal situation. The term $k_r r$ represents a feedforward signal from the reference to the control. The closed loop system obtained when the feedback (7.10) is applied to the sys-

tem (7.9) is given by

$$\frac{dx}{dt} = (A - BK)x + Bk_r r. \tag{7.11}$$

We attempt to determine the feedback gain K so that the closed loop system has the characteristic polynomial

$$p(s) = s^n + p_1 s^{n-1} + \cdots + p_{n-1}s + p_n. \tag{7.12}$$

This control problem is called the *eigenvalue assignment problem* or *pole placement problem* (we will define poles more formally in Chapter 9).

Note that k_r does not affect the stability of the system (which is determined by the eigenvalues of $A - BK$) but does affect the steady-state solution. In particular, the equilibrium point and steady-state output for the closed loop system are given by

$$x_e = -(A - BK)^{-1}Bk_r r, \qquad y_e = Cx_e + Du_e,$$

hence k_r should be chosen such that $y_e = r$ (the desired output value). Since k_r is a scalar, we can easily solve to show that if $D = 0$ (the most common case),

$$k_r = -1/\big(C(A - BK)^{-1}B\big). \tag{7.13}$$

Notice that k_r is exactly the inverse of the zero frequency gain of the closed loop system. The solution for $D \neq 0$ is left as an exercise.

Using the gains K and k_r, we are thus able to design the dynamics of the closed loop system to satisfy our goal. To illustrate how to construct such a state feedback control law, we begin with a few examples that provide some basic intuition and insights.

Example 7.4 Vehicle steering

In Example 6.12 we derived a normalized linear model for vehicle steering. The dynamics describing the lateral deviation were given by

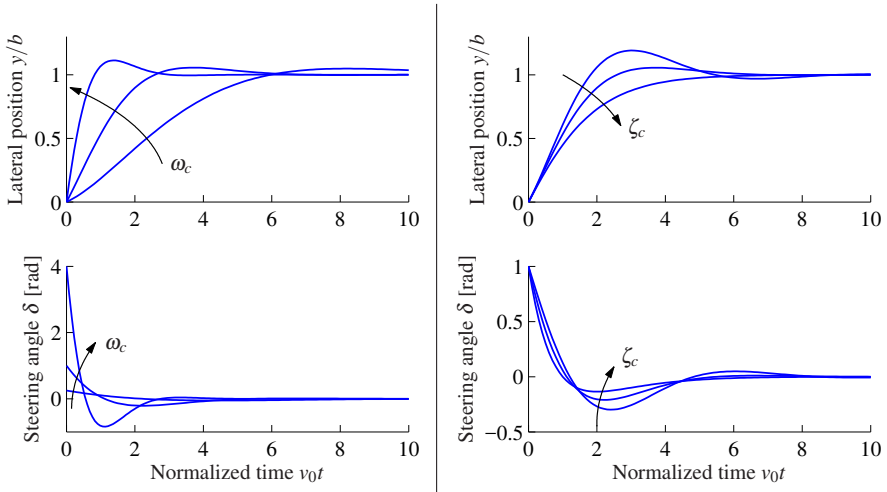
$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} \gamma \\ 1 \end{pmatrix},$$
$$C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \qquad D = 0.$$

We want to design a controller that stabilizes the dynamics and tracks a given reference value r of the lateral position of the vehicle. To do this we introduce the feedback

$$u = -Kx + k_r r = -k_1 x_1 - k_2 x_2 + k_r r,$$

and the closed loop system becomes

$$\begin{aligned} \frac{dx}{dt} &= (A - BK)x + Bk_r r = \begin{pmatrix} -\gamma k_1 & 1 - \gamma k_2 \\ -k_1 & -k_2 \end{pmatrix} x + \begin{pmatrix} \gamma k_r \\ k_r \end{pmatrix} r, \\ y &= Cx + Du = \begin{pmatrix} 1 & 0 \end{pmatrix} x. \end{aligned} \tag{7.14}$$



(a) Step response for varying ω_c (b) Step response for varying ζ_c

Figure 7.6: State feedback control of a steering system. Step responses (from zero initial condition) obtained with controllers designed with $\zeta_c = 0.7$ and $\omega_c = 0.5$, 1 and 2 [rad/s] are shown in (a). Notice that response speed increases with increasing ω_c, but that large ω_c also give large initial control actions. Step responses obtained with a controller designed with $\omega_c = 1$ and $\zeta_c = 0.5$, 0.7 and 1 are shown in (b).

The closed loop system has the characteristic polynomial

$$\det(sI - A + BK) = \det \begin{pmatrix} s + \gamma k_1 & \gamma k_2 - 1 \\ k_1 & s + k_2 \end{pmatrix} = s^2 + (\gamma k_1 + k_2)s + k_1.$$

Suppose that we would like to use feedback to design the dynamics of the system to have the characteristic polynomial

$$p(s) = s^2 + 2\zeta_c \omega_c s + \omega_c^2.$$

Comparing this polynomial with the characteristic polynomial of the closed loop system, we see that the feedback gains should be chosen as

$$k_1 = \omega_c^2, \qquad k_2 = 2\zeta_c \omega_c - \gamma \omega_c^2.$$

Equation (7.13) gives $k_r = k_1 = \omega_c^2$, and the control law can be written as

$$u = k_1(r - x_1) - k_2 x_2 = \omega_c^2 (r - x_1) - (2\zeta_c \omega_c - \gamma \omega_c^2) x_2.$$

The step responses for the closed loop system for different values of the design parameters are shown in Figure 7.6. The effect of ω_c is shown in Figure 7.6a, which shows that the response speed increases with increasing ω_c. The responses for $\omega_c = 0.5$ and 1 have reasonable overshoot. The settling time is about 15 car lengths for $\omega_c = 0.5$ (beyond the end of the plot) and decreases to about 6 car lengths for $\omega_c = 1$. The initial value of the control signal is $u(0) = k_r r = \omega_c^2 r$, and thus the achievable response time is limited by the available actuator signal. Notice in particular the dramatic increase in control signal when ω_c changes from 1 to 2.

The effect of ζ_c is shown in Figure 7.6b. The response speed and the overshoot increase with decreasing damping. Using these plots, we conclude that reasonable values of the design parameters are to have ω_c in the range of 0.5 to 1 and $\zeta_c \approx 0.7$. ∇

The example of the vehicle steering system illustrates how state feedback can be used to set the eigenvalues of a closed loop system to arbitrary values. We see that for this example we can set the eigenvalues to any location. We now show that this is a general property for reachable systems.

State Feedback for Systems in Reachable Canonical Form

The reachable canonical form has the property that the parameters of the system are the coefficients of the characteristic polynomial. It is therefore natural to consider systems in this form when solving the eigenvalue assignment problem.

Consider a system in reachable canonical form, i.e,

$$\frac{dz}{dt} = \tilde{A}z + \tilde{B}u = \begin{pmatrix} -a_1 & -a_2 & -a_3 & \dots & -a_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & 1 & 0 \end{pmatrix} z + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} u \qquad (7.15)$$

$$y = \tilde{C}z = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} z.$$

It follows from equation (7.7) that the open loop system has the characteristic polynomial

$$\det(sI - A) = s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n.$$

Before making a formal analysis we can gain some insight by investigating the block diagram of the system shown in Figure 7.4. The characteristic polynomial is given by the parameters a_k in the figure. Notice that the parameter a_k can be changed by feedback from state z_k to the input u. It is thus straightforward to change the coefficients of the characteristic polynomial by state feedback.

Returning to equations, introducing the control law

$$u = -\tilde{K}z + k_r r = -\tilde{k}_1 z_1 - \tilde{k}_2 z_2 - \cdots - \tilde{k}_n z_n + k_r r, \qquad (7.16)$$

the closed loop system becomes

$$\frac{dz}{dt} = \begin{pmatrix} -a_1 - \tilde{k}_1 & -a_2 - \tilde{k}_2 & -a_3 - \tilde{k}_3 & \dots & -a_n - \tilde{k}_n \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & 1 & 0 \end{pmatrix} z + \begin{pmatrix} k_r \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} r, \qquad (7.17)$$

$$y = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} z.$$

The feedback changes the elements of the first row of the A matrix, which corresponds to the parameters of the characteristic polynomial. The closed loop system thus has the characteristic polynomial

$$s^n + (a_1 + \tilde{k}_1)s^{n-1} + (a_2 + \tilde{k}_2)s^{n-2} + \cdots + (a_{n-1} + \tilde{k}_{n-1})s + a_n + \tilde{k}_n.$$

Requiring this polynomial to be equal to the desired closed loop polynomial

$$p(s) = s^n + p_1 s^{n-1} + \cdots + p_{n-1}s + p_n,$$

we find that the controller gains should be chosen as

$$\tilde{k}_1 = p_1 - a_1, \qquad \tilde{k}_2 = p_2 - a_2, \qquad \ldots \qquad \tilde{k}_n = p_n - a_n.$$

This feedback simply replaces the parameters a_i in the system (7.15) by p_i. The feedback gain for a system in reachable canonical form is thus

$$\tilde{K} = \begin{pmatrix} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{pmatrix}. \tag{7.18}$$

To have zero frequency gain equal to unity, we compute the equilibrium point z_e by setting the right hand side of equation (7.17) to zero and then compute the corresponding output. It can be seen that $z_{e,1}, \ldots, z_{e,n-1}$ must all be zero and we are left with

$$(-a_n - \tilde{k}_n)z_{e,n} + k_r r = 0 \quad \text{and} \quad y_e = b_n z_{e,n}.$$

It follows that in order for y_e to be equal to r then the parameter k_r should be chosen as

$$k_r = \frac{a_n + \tilde{k}_n}{b_n} = \frac{p_n}{b_n}. \tag{7.19}$$

Notice that it is essential to know the precise values of parameters a_n and b_n in order to obtain the correct zero frequency gain. The zero frequency gain is thus obtained by precise calibration. This is very different from obtaining the correct steady-state value by integral action, which we shall see in later sections.

Eigenvalue Assignment

We have seen through the examples how feedback can be used to design the dynamics of a system through assignment of its eigenvalues. To solve the problem in the general case, we simply change coordinates so that the system is in reachable canonical form. Consider the system

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du. \tag{7.20}$$

We can change the coordinates by a linear transformation $z = Tx$ so that the transformed system is in reachable canonical form (7.15). For such a system the feedback is given by equation (7.16), where the coefficients are given by equation (7.18). Transforming back to the original coordinates gives the control law

$$u = -\tilde{K}z + k_r r = -\tilde{K}Tx + k_r r.$$

The form of the controller is a feedback term $-Kx$ and a feedforward term $k_r r$.

The results obtained can be summarized as follows.

Theorem 7.3 (Eigenvalue assignment by state feedback)**.** *Consider the system given by equation* (7.20)*, with one input and one output. Let* $\lambda(s) = s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n$ *be the characteristic polynomial of A. If the system is reachable, then there exists a control law*

$$u = -Kx + k_r r$$

that gives a closed loop system with the characteristic polynomial

$$p(s) = s^n + p_1 s^{n-1} + \cdots + p_{n-1}s + p_n$$

and unity zero frequency gain between r and y. The feedback gain is given by

$$K = \tilde{K}T = \begin{pmatrix} p_1 - a_1 & p_2 - a_2 & \cdots & p_n - a_n \end{pmatrix} \tilde{W}_r W_r^{-1}, \tag{7.21}$$

where a_i *are the coefficients of the characteristic polynomial of the matrix A and the matrices* W_r *and* $\tilde{W}_r$ *are given by*

$$W_r = \begin{pmatrix} B & AB & \cdots & A^{n-1}B \end{pmatrix}, \qquad \tilde{W}_r = \begin{pmatrix} 1 & a_1 & a_2 & \cdots & a_{n-1} \\ 0 & 1 & a_1 & \cdots & a_{n-2} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & a_1 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}^{-1}.$$

The reference gain is given by

$$k_r = -1/\big(C(A - BK)^{-1}B\big).$$

For simple problems, the eigenvalue assignment problem can be solved by introducing the elements k_i of K as unknown variables. We then compute the characteristic polynomial

$$\lambda(s) = \det(sI - A + BK)$$

and equate coefficients of equal powers of s to the coefficients of the desired characteristic polynomial

$$p(s) = s^n + p_1 s^{n-1} + \cdots + p_{n-1}s + p_n.$$

This gives a system of linear equations to determine k_i. The equations can always be solved if the system is reachable, exactly as we did in Example 7.4.

Equation (7.21), which is called Ackermann's formula [Ack72, Ack85], can be used for numeric computations. It is implemented in the MATLAB function `acker`. The MATLAB function `place` is preferable for systems of high order because it is better conditioned numerically.

Example 7.5 Predator–prey
Consider the problem of regulating the population of an ecosystem by modulating

the food supply. We use the predator–prey model introduced in Example 5.14 and described in more detail in Section 4.7. The dynamics for the system are given by

$$\frac{dH}{dt} = (r+u)H\left(1-\frac{H}{k}\right) - \frac{aHL}{c+H}, \qquad H \geq 0,$$
$$\frac{dL}{dt} = b\frac{aHL}{c+H} - dL, \qquad L \geq 0.$$

We choose the following nominal parameters for the system, which correspond to the values used in previous simulations:

$$a = 3.2, \qquad b = 0.6, \qquad c = 50,$$
$$d = 0.56, \qquad k = 125 \qquad r = 1.6.$$

We take the parameter r, corresponding to the growth rate for hares, as the input to the system, which we might modulate by controlling a food source for the hares. This is reflected in our model by the term $(r+u)$ in the first equation, where here r represents a constant parameter (not the reference signal) and u represents the controlled modulation. We choose the number of lynxes L as the output of our system.

To control this system, we first linearize the system around the equilibrium point of the system (H_e, L_e), which can be determined numerically to be $x_e \approx (20.6, 29.5)$. This yields a linear dynamical system

$$\frac{d}{dt}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0.13 & -0.93 \\ 0.57 & 0 \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} 17.2 \\ 0 \end{pmatrix} v, \qquad w = \begin{pmatrix} 0 & 1 \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix},$$

where $z_1 = H - H_e$, $z_2 = L - L_e$ and $v = u$. It is easy to check that the system is reachable around the equilibrium $(z, v) = (0, 0)$, and hence we can assign the eigenvalues of the system using state feedback.

Selecting the eigenvalues of the closed loop system requires balancing the ability to modulate the input against the natural dynamics of the system. This can be done by the process of trial and error or by using some of the more systematic techniques discussed in the remainder of the text. For now, we simply choose the desired closed loop eigenvalues to be at $\lambda = \{-0.1, -0.2\}$. We can then solve for the feedback gains using the techniques described earlier, which results in

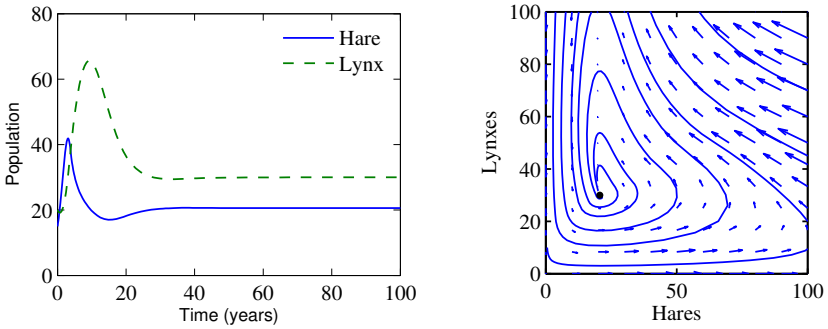
$$K = \begin{pmatrix} 0.025 & -0.052 \end{pmatrix}.$$

Finally, we solve for the reference gain k_r, using equation (7.13) to obtain $k_r = 0.002$.

Putting these steps together, our control law becomes

$$v = -Kz + k_r L_d,$$

where L_d is the desired number of lynxes. In order to implement the control law,



(a) Initial condition response (b) Phase portrait

Figure 7.7: Simulation results for the controlled predator–prey system. The population of lynxes and hares as a function of time is shown in (a), and a phase portrait for the controlled system is shown in (b). Feedback is used to make the population stable at $H_e = 20.6$ and $L_e = 30$.

we must rewrite it using the original coordinates for the system, yielding

$$\begin{aligned} u &= u_e - K(x - x_e) + k_r(L_d - y_e) \\ &= -\begin{pmatrix} 0.025 & -0.052 \end{pmatrix} \begin{pmatrix} H - 20.6 \\ L - 29.5 \end{pmatrix} + 0.002\,(L_d - 29.5). \end{aligned}$$

This rule tells us how much we should modulate u as a function of the current number of lynxes and hares in the ecosystem. Figure 7.7a shows a simulation of the resulting closed loop system using the parameters defined above and starting with an initial population of 15 hares and 20 lynxes. Note that the system quickly stabilizes the population of lynxes at the reference value ($L_d = 30$). A phase portrait of the system is given in Figure 7.7b, showing how other initial conditions converge to the stabilized equilibrium population. Notice that the dynamics are very different from the natural dynamics (shown in Figure 4.20). ∇

The results of this section show that we can use state feedback to design the dynamics of a reachable system, under the strong assumption that we can measure all of the states. We shall address the availability of the states in the next chapter, when we consider output feedback and state estimation. In addition, Theorem 7.3, which states that the eigenvalues can be assigned to arbitrary locations, is also highly idealized and assumes that the dynamics of the process are known to high precision. The robustness of state feedback combined with state estimators is considered in Chapter 13 after we have developed the requisite tools.

7.3 Design Considerations

The location of the eigenvalues determines the behavior of the closed loop dynamics, and hence where we place the eigenvalues is the main design decision to be

made. As with all other feedback design problems, there are trade-offs among the magnitude of the control inputs, the robustness of the system to perturbations and the closed loop performance of the system. In this section we examine some of these trade-offs starting with the special case of second-order systems.

Second-Order Systems

One class of systems that occurs frequently in the analysis and design of feedback systems is second-order linear differential equations. Because of their ubiquitous nature, it is useful to apply the concepts of this chapter to that specific class of systems and build more intuition about the relationship between stability and performance.

A canonical second-order system is a differential equation of the form

$$\ddot{q} + 2\zeta\omega_0\dot{q} + \omega_0^2 q = k\omega_0^2 u, \qquad y = q. \tag{7.22}$$

In state space form, this system can be represented as

$$\frac{dx}{dt} = \begin{pmatrix} 0 & \omega_0 \\ -\omega_0 & -2\zeta\omega_0 \end{pmatrix} x + \begin{pmatrix} 0 \\ k\omega_0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} x, \tag{7.23}$$

where $x = (q, \dot{q}/\omega_0)$ represents a normalized choice of states. The eigenvalues of this system are given by

$$\lambda = -\zeta\omega_0 \pm \omega_0\sqrt{(\zeta^2 - 1)},$$

and we see that the system is stable if $\omega_0 > 0$ and $\zeta > 0$. Note that the eigenvalues are complex if $\zeta < 1$ and real otherwise. Equations (7.22) and (7.23) can be used to describe many second-order systems, including damped oscillators, active filters and flexible structures, as shown in the examples below.

The form of the solution depends on the value of ζ, which is referred to as the *damping ratio* for the system. If $\zeta > 1$, we say that the system is *overdamped*, and the natural response ($u = 0$) of the system is given by

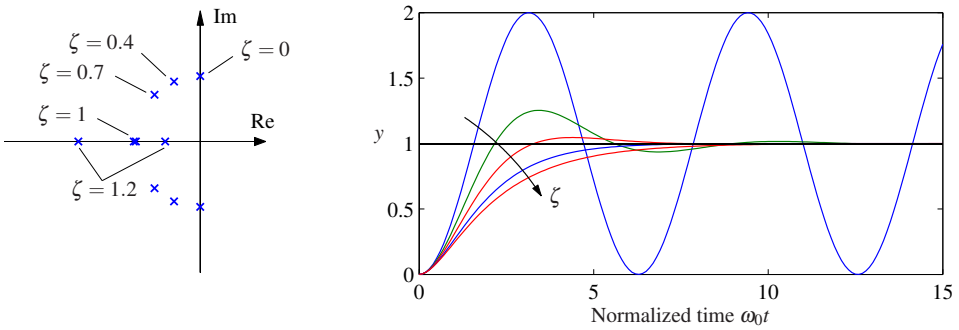
$$y(t) = \frac{\beta x_{10} + x_{20}}{\beta - \alpha} e^{-\alpha t} - \frac{\alpha x_{10} + x_{20}}{\beta - \alpha} e^{-\beta t},$$

where $\alpha = \omega_0(\zeta + \sqrt{\zeta^2 - 1})$ and $\beta = \omega_0(\zeta - \sqrt{\zeta^2 - 1})$. We see that the response consists of the sum of two exponentially decaying signals. If $\zeta = 1$, then the system is *critically damped* and solution becomes

$$y(t) = e^{-\zeta\omega_0 t}\big(x_{10} + (x_{20} + \zeta\omega_0 x_{10})t\big).$$

Note that this is still asymptotically stable as long as $\omega_0 > 0$, although the second term within the outer parentheses is increasing with time (but more slowly than the decaying exponential that is multiplying it).

Finally, if $0 < \zeta < 1$, then the solution is oscillatory and equation (7.22) is said



(a) Eigenvalues (b) Step responses

Figure 7.8: Step response for a second-order system. Normalized step responses for the system (7.23) for $\zeta = 0$, 0.4, 0.7, 1 and 1.2. As the damping ratio is increased, the rise time of the system gets longer, but there is less overshoot. The horizontal axis is in scaled units $\omega_0 t$; higher values of ω_0 result in a faster response (rise time and settling time).

to be *underdamped*. The natural response of the system is given by

$$y(t) = e^{-\zeta\omega_0 t}\left(x_{10}\cos\omega_d t + \left(\frac{\zeta\omega_0}{\omega_d}x_{10} + \frac{1}{\omega_d}x_{20}\right)\sin\omega_d t\right),$$

where $\omega_d = \omega_0\sqrt{1-\zeta^2}$ is called the *damped frequency*. For $\zeta \ll 1$, $\omega_d \approx \omega_0$ defines the oscillation frequency of the solution and ζ gives the damping rate relative to ω_0. The parameter ω_0 is referred to as the *natural frequency* of the system, stemming from the fact that for $\zeta = 0$ the oscillation frequency is given by ω_0.

Because of the simple form of a second-order system, it is possible to solve for the step and frequency responses in analytical form. The solution for the step response depends on the magnitude of ζ:

$$y(t) = \begin{cases} k\left(1 - e^{-\zeta\omega_0 t}\cos\omega_d t - \frac{\zeta}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_0 t}\sin\omega_d t\right), & \zeta < 1; \\ k\left(1 - e^{-\omega_0 t}(1+\omega_0 t)\right), & \zeta = 1; \\ k\left(1 - \frac{1}{2}\left(\frac{\zeta}{\sqrt{\zeta^2-1}}+1\right)e^{-\omega_0 t(\zeta-\sqrt{\zeta^2-1})}\right. & \\ \qquad\left. + \frac{1}{2}\left(\frac{\zeta}{\sqrt{\zeta^2-1}}-1\right)e^{-\omega_0 t(\zeta+\sqrt{\zeta^2-1})}\right), & \zeta > 1, \end{cases} \tag{7.24}$$

where we have taken $x(0) = 0$. Note that for the lightly damped case ($\zeta < 1$) we have an oscillatory solution at frequency ω_d.

Step responses of systems with $k = 1$ and different values of ζ are shown in Figure 7.8. The shape of the response is determined by ζ, and the speed of the response is determined by ω_0 (included in the time axis scaling): the response is faster if ω_0 is larger.

In addition to the explicit form of the solution, we can also compute the properties of the step response that were defined in Section 6.3. For example, to compute

Table 7.1: Properties of the step response for a second-order system with $0 < \zeta < 1$.

Property	Value	$\zeta = 0.5$	$\zeta = 1/\sqrt{2}$	$\zeta = 1$
Steady-state value	k	k	k	k
Rise time	$T_r \approx 1/\omega_0 \cdot e^{\varphi/\tan\varphi}$	$1.8/\omega_0$	$2.2/\omega_0$	$2.7/\omega_0$
Overshoot	$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$	16%	4%	0%
Settling time (2%)	$T_s \approx 4/\zeta\omega_0$	$8.0/\omega_0$	$5.9/\omega_0$	$5.8/\omega_0$

the maximum overshoot for an underdamped system, we rewrite the output as

$$y(t) = k\left(1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_0 t}\sin(\omega_d t + \varphi)\right), \tag{7.25}$$

where $\varphi = \arccos\zeta$. The maximum overshoot will occur at the first time in which the derivative of y is zero, at which time the fraction of the final value can be shown to be

$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}.$$

Similar computations can be done for the other characteristics of a step response. Table 7.1 summarizes the calculations.

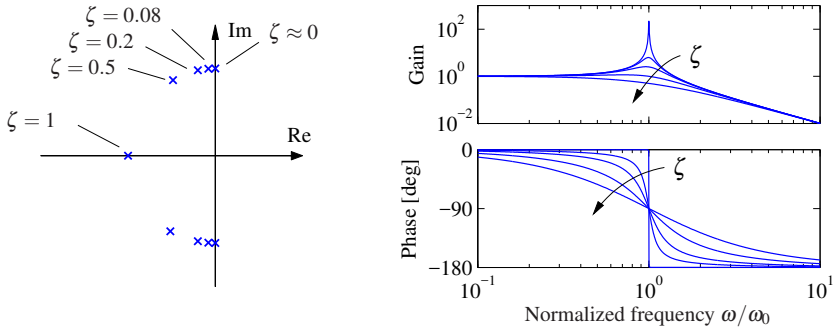
The frequency response for a second-order system can also be computed explicitly and is given by

$$Me^{j\theta} = \frac{k\omega_0^2}{(i\omega)^2 + 2\zeta\omega_0(i\omega) + \omega_0^2} = \frac{k\omega_0^2}{\omega_0^2 - \omega^2 + 2i\zeta\omega_0\omega}.$$

A graphical illustration of the frequency response is given in Figure 7.9. Notice the resonant peak that increases with decreasing ζ. The peak is often characterized by its *Q-value*, defined as $Q = 1/2\zeta$. The properties of the frequency response for a second-order system are summarized in Table 7.2.

Table 7.2: Properties of the frequency response for a second-order system with $0 < \zeta < 1$.

Property	Value	$\zeta = 0.1$	$\zeta = 0.5$	$\zeta = 1/\sqrt{2}$
Zero frequency gain	M_0	k	k	k
Bandwidth	ω_b	$1.54\,\omega_0$	$1.27\,\omega_0$	ω_0
Resonant peak gain	M_r	$1.54k$	$1.27k$	k
Resonant frequency	ω_{mr}	ω_0	$0.707\omega_0$	0



(a) Eigenvalues (b) Frequency responses

Figure 7.9: Frequency response of a second-order system (7.23). (a) Eigenvalues as a function of ζ. (b) Frequency response as a function of ζ. The upper curve shows the gain ratio M, and the lower curve shows the phase shift θ. For small ζ there is a large peak in the magnitude of the frequency response and a rapid change in phase centered at $\omega = \omega_0$. As ζ is increased, the magnitude of the peak drops and the phase changes more smoothly between 0° and -180°.

Example 7.6 Drug administration

To illustrate the use of these formulas, consider the two-compartment model for drug administration, described in Section 4.6. The dynamics of the system are

$$\frac{dc}{dt} = \begin{pmatrix} -k_0 - k_1 & k_1 \\ k_2 & -k_2 \end{pmatrix} c + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 0 & 1 \end{pmatrix} c,$$

where c_1 and c_2 are the concentrations of the drug in each compartment, $k_i, i = 0,\ldots,2$ and b_0 are parameters of the system, u is the flow rate of the drug into compartment 1 and y is the concentration of the drug in compartment 2. We assume that we can measure the concentrations of the drug in each compartment, and we would like to design a feedback law to maintain the output at a given reference value r.

We choose $\zeta = 1/\sqrt{2}$ to minimize the overshoot and additionally require the rise time to be $T_r = 10$ min. Using the formulas in Table 7.1, this gives a value for $\omega_0 = 0.22$. We can now compute the gains to place the eigenvalues at this location. Setting $u = -Kx + k_r r$, the closed loop eigenvalues for the system satisfy

$$\lambda(s) = -0.2 \pm 0.096i.$$

Choosing $\tilde{k}_1 = -0.2$ and $\tilde{k}_2 = 0.2$, with $K = (\tilde{k}_1, \tilde{k}_2)$ to avoid confusion with the rates k_i in the dynamics matrix, gives the desired closed loop behavior. Equation (7.13) gives the reference gain $k_r = 0.065$. The response of the controller is shown in Figure 7.10 and compared with an open loop strategy involving administering periodic doses of the drug. ∇

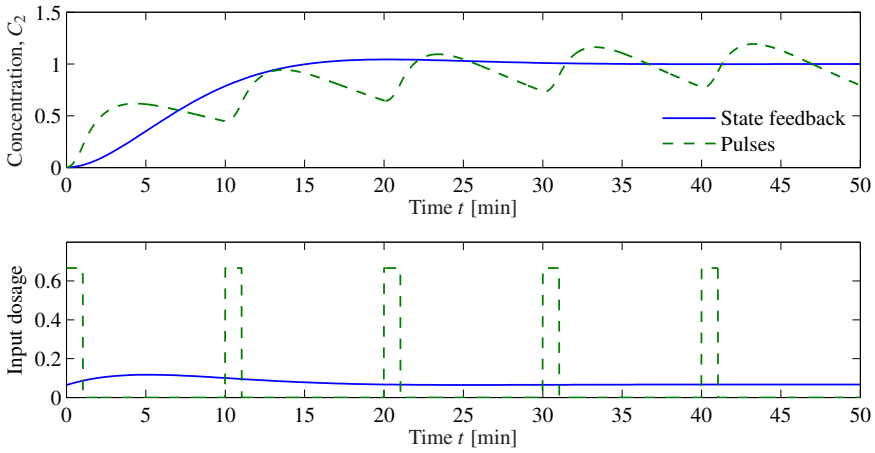


Figure 7.10: Open loop versus closed loop drug administration. Comparison between drug administration using a sequence of doses versus continuously monitoring the concentrations and adjusting the dosage continuously. In each case, the concentration is (approximately) maintained at the desired level, but the closed loop system has substantially less variability in drug concentration.

Higher-Order Systems

Our emphasis so far has considered only second-order systems. For higher-order systems, eigenvalue assignment is considerably more difficult, especially when trying to account for the many trade-offs that are present in a feedback design.

One of the other reasons why second-order systems play such an important role in feedback systems is that even for more complicated systems the response is often characterized by the *dominant eigenvalues*. To define these more precisely, consider a stable system with eigenvalues λ_j, $j = 1, \ldots, n$. We say that a complex conjugate pair of eigenvalues λ, λ^* is a *dominant pair* if they are the closest pair to the imaginary axis. In the case when multiple eigenvalues pairs are the same distance to the imaginary axis, a second criterion is to look at the relative damping of the system modes. We define the *damping ratio* for a complex eigenvalue λ to be

$$\zeta = \frac{-\operatorname{Re}\lambda}{|\lambda|}.$$

Given multiple complex conjugate pairs with the same real part, the dominant pair will the set with the lowest damping ratio.

Assuming that a system is stable, the dominant pair of eigenvalues tends to be the most important element of the response. To see this, assume that we have a system in Jordan form with a simple Jordan block corresponding to the dominant

pair of eigenvalues:

$$\frac{dz}{dt} = \begin{pmatrix} \lambda & & & & \\ & \lambda^* & & & \\ & & J_2 & & \\ & & & \ddots & \\ & & & & J_k \end{pmatrix} z + Bu, \qquad y = Cz.$$

(Note that the state z may be complex because of the Jordan transformation.) The response of the system will be a linear combination of the responses from each of the individual Jordan subsystems. As we see from Figure 7.8, for $\zeta < 1$ the subsystem with the slowest response is precisely the one with the whose eigenvalues are closest to the imaginary axis. Hence, when we add the responses from each of the individual subsystems, it is the dominant pair of eigenvalues that will be the primary factor after the initial transients due to the other terms in the solution die out. While this simple analysis does not always hold (e.g., if some nondominant terms have larger coefficients because of the particular form of the system), it is often the case that the dominant eigenvalues determine the (step) response of the system.

The only formal requirement for eigenvalue assignment is that the system be reachable. In practice there are many other constraints because the selection of eigenvalues has a strong effect on the magnitude and rate of change of the control signal. Large eigenvalues will in general require large control signals as well as fast changes of the signals. The capability of the actuators will therefore impose constraints on the possible location of closed loop eigenvalues. These issues will be discussed in depth in Chapters 12 and 13.

We illustrate some of the main ideas using the balance system as an example.

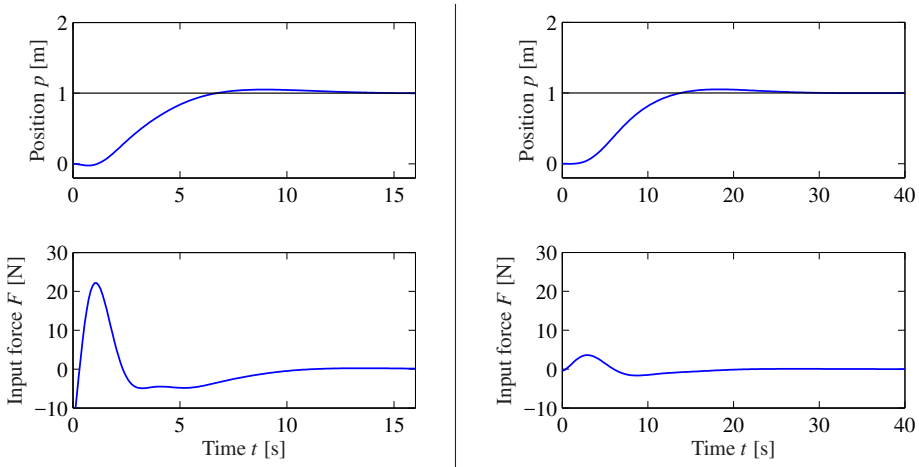
Example 7.7 Balance system
Consider the problem of stabilizing a balance system, whose dynamics were given in Example 7.2. The dynamics are given by

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & m^2 l^2 g/\mu & -cJ_t/\mu & -\gamma lm/\mu \\ 0 & M_t mgl/\mu & -clm/\mu & -\gamma M_t/\mu \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 0 \\ J_t/\mu \\ lm/\mu \end{pmatrix},$$

where $M_t = M + m$, $J_t = J + ml^2$, $\mu = M_t J_t - m^2 l^2$ and we have left c and γ nonzero. We use the following parameters for the system (corresponding roughly to a human being balanced on a stabilizing cart):

$$\begin{aligned} &M = 10\ \text{kg}, && m = 80\ \text{kg}, && c = 0.1\ \text{N s/m}, \\ &J = 100\ \text{kg m}^2/\text{s}^2, && l = 1\ \text{m}, && \gamma = 0.01\ \text{N m s}, \end{aligned} \qquad g = 9.8\ \text{m/s}^2.$$

The eigenvalues of the open loop dynamics are given by $\lambda \approx 0, -0.0011, -2.68, 2.68$. We have verified already in Example 7.2 that the system is reachable, and hence



(a) $\lambda_{1,2} = -1 \pm 2i$ (b) $\lambda_{1,2} = -0.33 \pm 0.66i$

Figure 7.11: State feedback control of a balance system. The step response of a controller designed to give fast performance is shown in (a). Although the response characteristics (top left) look very good, the input magnitude (bottom left) is very large. A less aggressive controller is shown in (b). Here the response time is slowed down, but the input magnitude is much more reasonable. Both step responses are applied to the linearized dynamics.

we can use state feedback to stabilize the system and provide a desired level of performance.

To decide where to place the closed loop eigenvalues, we note that the closed loop dynamics will roughly consist of two components: a set of fast dynamics that stabilize the pendulum in the inverted position and a set of slower dynamics that control the position of the cart. For the fast dynamics, we look to the natural period of the pendulum (in the hanging-down position), which is given by $\omega_0 = \sqrt{mgl/(J+ml^2)} \approx 2.1$ rad/s. To provide a fast response we choose a damping ratio of $\zeta = 0.5$ and try to place the first pair of eigenvalues at $\lambda_{1,2} \approx -\zeta\omega_0 \pm i\omega_0 \approx -1 \pm 2i$, where we have used the approximation that $\sqrt{1-\zeta^2} \approx 1$. For the slow dynamics, we choose the damping ratio to be 0.7 to provide a small overshoot and choose the natural frequency to be 0.5 to give a rise time of approximately 5 s. This gives eigenvalues $\lambda_{3,4} = -0.35 \pm 0.35i$.

The controller consists of a feedback on the state and a feedforward gain for the reference input. The feedback gain is given by

$$K = \begin{pmatrix} -15.6 & 1730 & -50.1 & 443 \end{pmatrix},$$

which can be computed using Theorem 7.3 or using the MATLAB `place` command. The feedforward gain is $k_r = -1/(C(A-BK)^{-1}B) = -15.6$. The step response for the resulting controller (applied to the linearized system) is given in Figure 7.11a. While the step response gives the desired characteristics, the input required (bottom left) is excessively large, almost three times the force of gravity at its peak.

To provide a more realistic response, we can redesign the controller to have slower dynamics. We see that the peak of the input force occurs on the fast time scale, and hence we choose to slow this down by approximately a factor of 3, leaving the damping ratio unchanged. We also slow down the second set of eigenvalues, with the intuition that we should move the position of the cart more slowly than we stabilize the pendulum dynamics. Leaving the damping ratio for the slow dynamics unchanged at 0.7 and changing the frequency to 1 (corresponding to a rise time of approximately 10 s), the desired eigenvalues become

$$\lambda = \{-0.33 \pm 0.66i,\ -0.18 \pm 0.18i\}.$$

The performance of the resulting controller is shown in Figure 7.11b. ∇

As we see from this example, it can be difficult to decide where to place the eigenvalues using state feedback. This is one of the principal limitations of this approach, especially for systems of higher dimension. Optimal control, such as the linear quadratic regulator problem discussed next, is one approach that is available. One can also focus on the frequency response for performing the design, which is the subject of Chapters 9–13.

Linear Quadratic Regulators

As an alternative to selecting the closed loop eigenvalue locations to accomplish a certain objective, the gains for a state feedback controller can instead be chosen is by attempting to optimize a cost function. This can be particularly useful in helping balance the performance of the system with the magnitude of the inputs required to achieve that level of performance.

The infinite horizon, linear quadratic regulator (LQR) problem is one of the most common optimal control problems. Given a multi-input linear system

$$\frac{dx}{dt} = Ax + Bu, \qquad x \in \mathbb{R}^n, u \in \mathbb{R}^p,$$

we attempt to minimize the quadratic cost function

$$\tilde{J} = \int_0^\infty \left(x^T Q_x x + u^T Q_u u\right) dt, \tag{7.26}$$

where $Q_x \succeq 0$ and $Q_u \succ 0$ are symmetric, positive (semi-) definite matrices of the appropriate dimensions. This cost function represents a trade-off between the distance of the state from the origin and the cost of the control input. By choosing the matrices Q_x and Q_u, we can balance the rate of convergence of the solutions with the cost of the control.

The solution to the LQR problem is given by a linear control law of the form

$$u = -Q_u^{-1} B^T P x,$$

where $P \in \mathbb{R}^{n \times n}$ is a positive definite, symmetric matrix that satisfies the equation

$$PA + A^T P - PBQ_u^{-1} B^T P + Q_x = 0. \tag{7.27}$$

Equation (7.27) is called the *algebraic Riccati equation*. If the system is reachable, it can be shown that there is a unique positive definite matrix P satisfying equation (7.27) that makes the closed loop system stable. The MATLAB command `lqr` returns the optimal controller given A, B, Q_x and Q_u.

One of the key questions in LQR design is how to choose the weights Q_x and Q_u. To guarantee that a solution exists, we must have $Q_x \succeq 0$ and $Q_u \succ 0$. In addition, there are certain "observability" conditions on Q_x that limit its choice. Here we assume $Q_x \succ 0$ to ensure that solutions to the algebraic Riccati equation always exist.

To choose specific values for the cost function weights Q_x and Q_u, we must use our knowledge of the system we are trying to control. A particularly simple choice is to use diagonal weights

$$Q_x = \begin{pmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{pmatrix}, \qquad Q_u = \begin{pmatrix} \rho_1 & & 0 \\ & \ddots & \\ 0 & & \rho_n \end{pmatrix}.$$

For this choice of Q_x and Q_u, the individual diagonal elements describe how much each state and input (squared) should contribute to the overall cost. Hence, we can take states that should remain small and attach higher weight values to them. Similarly, we can penalize an input versus the states and other inputs through choice of the corresponding input weight ρ.

Example 7.8 Vectored thrust aircraft
Consider the original dynamics of the system (3.29), written in state space form as

$$\frac{dz}{dt} = \begin{pmatrix} z_4 \\ z_5 \\ z_6 \\ -\frac{c}{m} z_4 \\ -g - \frac{c}{m} z_5 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m}\cos\theta\, F_1 - \frac{1}{m}\sin\theta\, F_2 \\ \frac{1}{m}\sin\theta\, F_1 + \frac{1}{m}\cos\theta\, F_2 \\ \frac{r}{J} F_1 \end{pmatrix}$$

(see also Example 6.4). The system parameters are $m = 4$ kg, $J = 0.0475$ kg m^2, $r = 0.25$ m, $g = 9.8$ m/s^2, $c = 0.05$ N s/m, which corresponds to a scaled model of the system. The equilibrium point for the system is given by $F_1 = 0$, $F_2 = mg$ and $z_e = (x_e, y_e, 0, 0, 0, 0)$. To derive the linearized model near an equilibrium point, we

compute the linearization according to equation (6.34):

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -g & -c/m & 0 & 0 \\ 0 & 0 & 0 & 0 & -c/m & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/m \\ r/J & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Letting $\xi = z - z_e$ and $v = F - F_e$, the linearized system is given by

$$\frac{d\xi}{dt} = A\xi + Bv, \qquad y = C\xi.$$

It can be verified that the system is reachable.

To compute a linear quadratic regulator for the system, we write the cost function as

$$J = \int_0^\infty (\xi^T Q_\xi \xi + v^T Q_v v)dt,$$

where $\xi = z - z_e$ and $v = F - F_e$ again represent the local coordinates around the desired equilibrium point (z_e, F_e). We begin with diagonal matrices for the state and input costs:

$$Q_\xi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad Q_v = \begin{pmatrix} \rho & 0 \\ 0 & \rho \end{pmatrix}.$$

This gives a control law of the form $v = -K\xi$, which can then be used to derive the control law in terms of the original variables:

$$F = v + F_e = -K(z - z_e) + F_e.$$

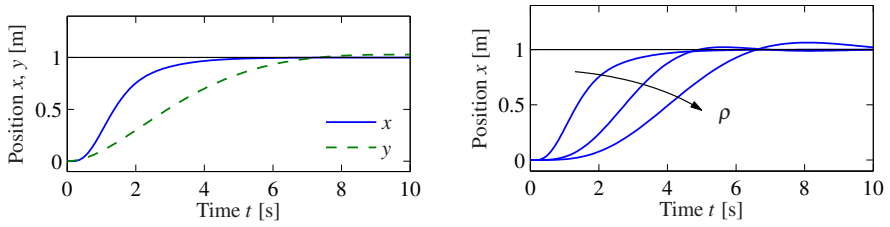
As computed in Example 6.4, the equilibrium points have $F_e = (0, mg)$ and $z_e = (x_e, y_e, 0, 0, 0, 0)$. The response of the controller to a step change in the desired position is shown in Figure 7.12a for $\rho = 1$. The response can be tuned by adjusting the weights in the LQR cost. Figure 7.12b shows the response in the x direction for different choices of the weight ρ.

∇

Linear quadratic regulators can also be designed for discrete-time systems, as illustrated by the following example.

Example 7.9 Web server control

Consider the web server example given in Section 4.4, where a discrete-time model



(a) Step response in x and y (b) Effect of control weight ρ

Figure 7.12: Step response for a vectored thrust aircraft. The plot in (a) shows the x and y positions of the aircraft when it is commanded to move 1 m in each direction. In (b) the x motion is shown for control weights $\rho = 1$, 10^2, 10^4. A higher weight of the input term in the cost function causes a more sluggish response.

for the system was given. We wish to design a control law that sets the server parameters so that the average server processor load is maintained at a desired level. Since other processes may be running on the server, the web server must adjust its parameters in response to changes in the load.

A block diagram for the control system is shown in Figure 7.13. We focus on the special case where we wish to control only the processor load using both the `KeepAlive` and `MaxClients` parameters. We also include a "disturbance" on the measured load that represents the use of the processing cycles by other processes running on the server. The system has the same basic structure as the generic control system in Figure 7.5, with the variation that the disturbance enters after the process dynamics.

The dynamics of the system are given by a set of difference equations of the form

$$x[k+1] = Ax[k] + Bu[k], \qquad y_{\text{cpu}}[k] = x_{\text{cpu}}[k] + d_{\text{cpu}}[k],$$

where $x = (x_{\text{cpu}}, x_{\text{mem}})$ is the state of the web server, $u = (u_{\text{ka}}, u_{\text{mc}})$ is the input, d_{cpu} is the processing load from other processes on the computer and y_{cpu} is the total processor load. The matrices $A \in \mathbb{R}^{2\times 2}$ and $B \in \mathbb{R}^{2\times 2}$ are described in Section 4.4.

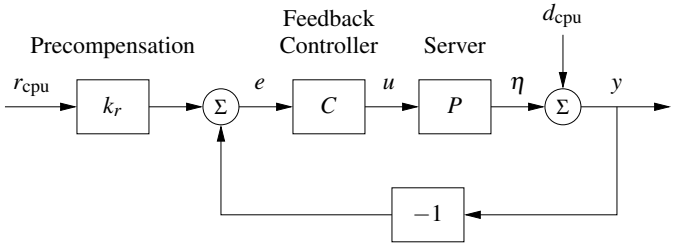


Figure 7.13: Feedback control of a web server. The controller sets the values of the web server parameters based on the difference between the nominal parameters (determined by $k_r r$) and the current load y_{cpu}. The disturbance v represents the load due to other processes running on the server. Note that the measurement is taken after the disturbance so that we measure the total load on the server.

We choose our controller to be a feedback controller of the form

$$u = -K \begin{pmatrix} y_{\text{cpu}} \\ x_{\text{mem}} \end{pmatrix} + k_r r_{\text{cpu}},$$

where r_{cpu} is the desired processor load. Note that we have used the measured processor load y_{cpu} instead of the CPU state x_{cpu} to ensure that we adjust the system operation based on the actual load. (This modification is necessary because of the nonstandard way in which the disturbance enters the process dynamics.)

The feedback gain matrix K can be chosen by any of the methods described in this chapter. Here we use a linear quadratic regulator, with the cost function given by

$$Q_x = \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix}, \qquad Q_u = \begin{pmatrix} 1/50^2 & 0 \\ 0 & 1/1000^2 \end{pmatrix}.$$

The cost function for the state Q_x is chosen so that we place more emphasis on the processor load versus the memory use. The cost function for the inputs Q_u is chosen so as to normalize the two inputs, with a `KeepAlive` timeout of 50 s having the same weight as a `MaxClients` value of 1000. These values are squared since the cost associated with the inputs is given by $u^T Q_u u$. Using the dynamics in Section 4.4 and the `dlqr` command in MATLAB, the resulting gains become

$$K = \begin{pmatrix} -22.3 & 10.1 \\ 382.7 & 77.7 \end{pmatrix}.$$

As in the case of a continuous-time control system, the reference gain k_r is chosen to yield the desired equilibrium point for the system. Setting $x[k+1] = x[k] = x_e$, the steady-state equilibrium point and output for a given reference input r are given by

$$x_e = (A - BK)x_e + Bk_r r, \qquad y_e = Cx_e.$$

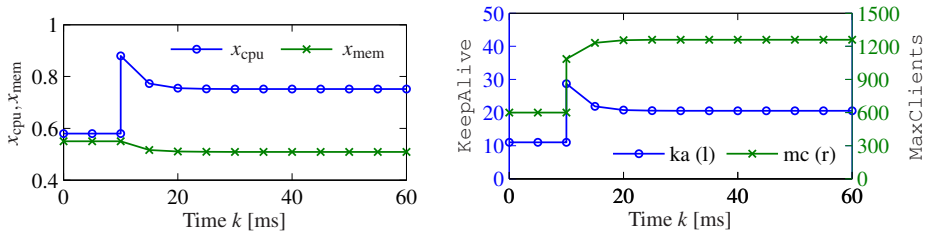
This is a matrix equation in which k_r is a column vector that sets the two inputs values based on the desired reference. If we take the desired output to be of the form $y_e = (r, 0)$, then we must solve

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = C(A - BK - I)^{-1} Bk_r.$$

Solving this equation for k_r, we obtain

$$k_r = \left(\left(C(A - BK - I)^{-1} B \right) \right)^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 49.3 \\ 539.5 \end{pmatrix}.$$

The dynamics of the closed loop system are illustrated in Figure 7.14. We apply a change in load of $d_{\text{cpu}} = 0.3$ at time $t = 10$ s, forcing the controller to adjust the operation of the server to attempt to maintain the desired load at 0.57. Note that both the `KeepAlive` and `MaxClients` parameters are adjusted. Although the load is decreased, it remains approximately 0.2 above the desired steady state. (Better results can be obtained using the techniques of the next section.) ∇



(a) System state (b) System inputs

Figure 7.14: Web server with LQR control. The plot in (a) shows the state of the system under a change in external load applied at $k = 10$ ms. The corresponding web server parameters (system inputs) are shown in (b). The controller is able to reduce the effect of the disturbance by approximately 40%.

7.4 Integral Action

Controllers based on state feedback achieve the correct steady-state response to command signals by careful calibration of the gain k_r. However, one of the primary uses of feedback is to allow good performance in the presence of uncertainty, and hence requiring that we have an *exact* model of the process is undesirable. An alternative to calibration is to make use of integral feedback, in which the controller uses an integrator to provide zero steady-state error. The basic concept of integral feedback was introduced in Section 1.6; here we provide a more complete description and analysis.

The basic approach in integral feedback is to create a state within the controller that computes the integral of the error signal, which is then used as a feedback term. We do this by augmenting the description of the system with a new state z:

$$\frac{d}{dt}\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} Ax + Bu \\ y - r \end{pmatrix} = \begin{pmatrix} Ax + Bu \\ Cx - r \end{pmatrix}. \tag{7.28}$$

The state z is seen to be the integral of the difference between the the actual output y and desired output r. Note that if we find a controller $u = \varphi(x, z)$ that stabilizes the system, then we will necessarily have $\dot{z} = 0$ in steady state and hence $y = r$ in steady state.

Given the augmented system, we design a state space controller in the usual fashion, with a control law of the form

$$u = -Kx - k_i z + k_r r, \tag{7.29}$$

where K is the usual state feedback term, k_i is the integral term and k_r is used to set the nominal input for the desired steady state. The resulting equilibrium point for the system is given as

$$x_e = -(A - BK)^{-1}B(k_r r - k_i z_e), \qquad Cx_e = r,$$

which comes from setting the right hand side of equation (7.28) to zero and substituting u from equation (7.29). Note that the value of z_e is not specified but rather

will automatically settle to the value that makes $\dot{z} = y - r = 0$, which implies that at equilibrium the output will equal the reference value. This holds independently of the specific values of A, B and K as long as the system is stable (which can be done through appropriate choice of K and k_i).

The final control law is given by

$$u = -Kx - k_i z + k_r r, \qquad \frac{dz}{dt} = y - r,$$

where we have now included the dynamics of the integrator as part of the specification of the controller. This type of control law is known as a *dynamic compensator* since it has its own internal dynamics. The following example illustrates the basic approach.

Example 7.10 Cruise control

Consider the cruise control example introduced in Section 1.5 and considered further in Example 6.11 (see also Section 4.1). The linearized dynamics of the process around an equilibrium point v_e, u_e are given by

$$\frac{dx}{dt} = ax - b_g\theta + bw, \qquad y = v = x + v_e,$$

where $x = v - v_e$, $w = u - u_e$, m is the mass of the car and θ is the angle of the road. The constant a depends on the throttle characteristic and is given in Example 6.11.

If we augment the system with an integrator, the process dynamics become

$$\frac{dx}{dt} = ax - b_g\theta + bw, \qquad \frac{dz}{dt} = y - v_r = v_e + x - v_r,$$

or, in state space form,

$$\frac{d}{dt}\begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} a & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x \\ z \end{pmatrix} + \begin{pmatrix} b \\ 0 \end{pmatrix} w + \begin{pmatrix} -b_g \\ 0 \end{pmatrix} \theta + \begin{pmatrix} 0 \\ v_e - v_r \end{pmatrix}.$$

Note that when the system is at equilibrium, we have that $\dot{z} = 0$, which implies that the vehicle speed $v = v_e + x$ should be equal to the desired reference speed v_r. Our controller will be of the form

$$\frac{dz}{dt} = y - v_r, \qquad w = -k_p x - k_i z + k_r v_r,$$

and the gains k_p, k_i and k_r will be chosen to stabilize the system and provide the correct input for the reference speed.

Assume that we wish to design the closed loop system to have the characteristic polynomial

$$\lambda(s) = s^2 + a_1 s + a_2.$$

Setting the disturbance $\theta = 0$, the characteristic polynomial of the closed loop system is given by

$$\det\big(sI - (A - BK)\big) = s^2 + (bk_p - a)s + bk_i,$$

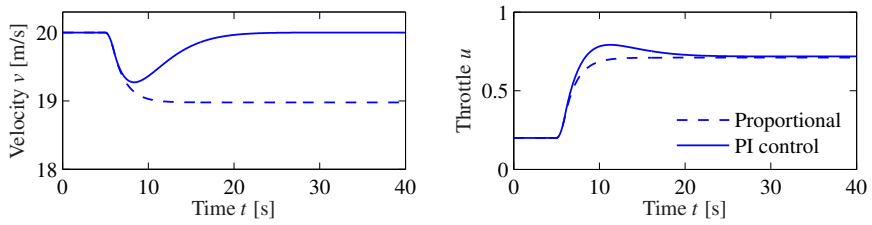


Figure 7.15: Velocity and throttle for a car with cruise control based on proportional (dashed) and PI control (solid). The PI controller is able to adjust the throttle to compensate for the effect of the hill and maintain the speed at the reference value of $v_r = 20$ m/s. The controller gains are $k_p = 0.5$ and $k_i = 0.1$.

and hence we set

$$k_p = \frac{a_1 + a}{b}, \qquad k_i = \frac{a_2}{b}, \qquad k_r = -1/\big(C(A - BK)^{-1}B\big) = \frac{a_1}{b}.$$

The resulting controller stabilizes the system and hence brings $\dot{z} = y - v_r$ to zero, resulting in perfect tracking. Notice that even if we have a small error in the values of the parameters defining the system, as long as the closed loop eigenvalues are still stable, then the tracking error will approach zero. Thus the exact calibration required in our previous approach (using k_r) is not needed here. Indeed, we can even choose $k_r = 0$ and let the feedback controller do all of the work. However, k_r does influence the transient response to command signals and setting it properly will generally give a more favorable response.

Integral feedback can also be used to compensate for constant disturbances. Figure 7.15 shows the results of a simulation in which the car encounters a hill with angle $\theta = 4^\circ$ at $t = 8$ s. The stability of the system is not affected by this external disturbance, and so we once again see that the car's velocity converges to the reference speed. This ability to handle constant disturbances is a general property of controllers with integral feedback (see Exercise 7.4). ∇

7.5 Further Reading

The importance of state models and state feedback was discussed in the seminal paper by Kalman [Kal60], where the state feedback gain was obtained by solving an optimization problem that minimized a quadratic loss function. The notions of reachability and observability (Chapter 8) are also due to Kalman [Kal61b] (see also [Gil63, KHN63]). Kalman defines controllability and reachability as the ability to reach the origin and an arbitrary state, respectively [KFA69]. We note that in most textbooks the term "controllability" is used instead of "reachability," but we prefer the latter term because it is more descriptive of the fundamental property of being able to reach arbitrary states. Most undergraduate textbooks on control contain material on state space systems, including, for example, Franklin, Powell and Emami-Naeini [FPEN05] and Ogata [Oga01]. Friedland's textbook [Fri04]

covers the material in the previous, current and next chapter in considerable detail, including the topic of optimal control.

Exercises

7.1 (Double integrator) Consider the double integrator. Find a piecewise constant control strategy that drives the system from the origin to the state $x = (1,1)$.

7.2 (Reachability from nonzero initial state) Extend the argument in Section 7.1 to show that if a system is reachable from an initial state of zero, it is reachable from a nonzero initial state.

7.3 (Unreachable systems) Consider the system shown in Figure 7.3. Write the dynamics of the two systems as

$$\frac{dx}{dt} = Ax + Bu, \qquad \frac{dz}{dt} = Az + Bu.$$

If x and z have the same initial condition, they will always have the same state regardless of the input that is applied. Show that this violates the definition of reachability and further show that the reachability matrix W_r is not full rank.

7.4 (Integral feedback for rejecting constant disturbances) Consider a linear system of the form

$$\frac{dx}{dt} = Ax + Bu + Fd, \qquad y = Cx$$

where u is a scalar and v is a disturbance that enters the system through a disturbance vector $F \in \mathbb{R}^n$. Assume that the matrix A is invertible and the zero frequency gain $CA^{-1}B$ is nonzero. Show that integral feedback can be used to compensate for a constant disturbance by giving zero steady-state output error even when $d \neq 0$.

7.5 (Rear-steered bicycle) A simple model for a bicycle was given by equation (4.5) in Section 4.2. A model for a bicycle with rear-wheel steering is obtained by reversing the sign of the velocity in the model. Determine the conditions under which this systems is reachable and explain any situations in which the system is not reachable.

7.6 (Characteristic polynomial for reachable canonical form) Show that the characteristic polynomial for a system in reachable canonical form is given by equation (7.7) and that

$$\frac{d^n z_k}{dt^n} + a_1 \frac{d^{n-1} z_k}{dt^{n-1}} + \cdots + a_{n-1} \frac{dz_k}{dt} + a_n z_k = \frac{d^{n-k} u}{dt^{n-k}},$$

where z_k is the kth state.

7.7 (Reachability matrix for reachable canonical form) Consider a system in reachable canonical form. Show that the inverse of the reachability matrix is given by

$$\tilde{W}_r^{-1} = \begin{pmatrix} 1 & a_1 & a_2 & \cdots & a_n \\ 0 & 1 & a_1 & \cdots & a_{n-1} \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & & & \ddots & a_1 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

7.8 (Non-maintainable equilibria) Consider the normalized model of a pendulum on a cart

$$\frac{d^2x}{dt^2} = u, \qquad \frac{d^2\theta}{dt^2} = -\theta + u,$$

where x is cart position and θ is pendulum angle. Can the angle $\theta = \theta_0$ for $\theta_0 \neq 0$ be maintained?

7.9 (Eigenvalue assignment for unreachable system) Consider the system

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} x,$$

with the control law

$$u = -k_1 x_1 - k_2 x_2 + k_r r.$$

Compute the rank of the reachability matrix for the system and show that eigenvalues of the system cannot be assigned to arbitrary values.

7.10 (Cayley–Hamilton theorem) Let $A \in \mathbb{R}^{n\times n}$ be a matrix with characteristic polynomial $\lambda(s) = \det(sI - A) = s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n$. Assume that the matrix A can be diagonalized and show that it satisfies

$$\lambda(A) = A^n + a_1 A^{n-1} + \cdots + a_{n-1}A + a_n I = 0,$$

Use the result to show that A^k, $k \geq n$, can be rewritten in terms of powers of A of order less than n.

7.11 (Dominant pairs)

7.12 (Motor drive) Consider the normalized model of the motor drive in Exercise 3.10. Using the following normalized parameters,

$$J_1 = 10/9, \qquad J_2 = 10, \qquad c = 0.1, \qquad k = 1, \qquad k_I = 1,$$

verify that the eigenvalues of the open loop system are $0, 0, -0.05 \pm i$. Design a state feedback that gives a closed loop system with eigenvalues -2, -1 and $-1 \pm i$. This choice implies that the oscillatory eigenvalues will be well damped and that the eigenvalues at the origin are replaced by eigenvalues on the negative real axis. Simulate the responses of the closed loop system to step changes in the command signal for θ_2 and a step change in a disturbance torque on the second rotor.

7.13 (Whipple bicycle model) Consider the Whipple bicycle model given by equation (4.7) in Section 4.2. Using the parameters from the companion web site, the model is unstable at the velocity $v = 5$ m/s and the open loop eigenvalues are -1.84, -14.29 and $1.30 \pm 4.60i$. Find the gains of a controller that stabilizes the bicycle and gives closed loop eigenvalues at -2, -10 and $-1 \pm i$. Simulate the response of the system to a step change in the steering reference of 0.002 rad.

7.14 (Atomic force microscope) Consider the model of an AFM in contact mode given in Example 6.9:

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -k_2/(m_1+m_2) & -c_2/(m_1+m_2) & 1/m_2 & 0 \\ 0 & 0 & 0 & \omega_3 \\ 0 & 0 & -\omega_3 & -2\zeta_3\omega_3 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \omega_3 \end{pmatrix} u,$$

$$y = \frac{m_2}{m_1+m_2} \begin{pmatrix} \frac{m_1 k_2}{m_1+m_2} & \frac{m_1 c_2}{m_1+m_2} & 1 & 0 \end{pmatrix} x.$$

Use the MATLAB script `afm_data.m` from the companion web site to generate the system matrices.

(a) Compute the reachability matrix of the system and numerically determine its rank. Scale the model by using milliseconds instead of seconds as time units. Repeat the calculation of the reachability matrix and its rank.

(b) Find a state feedback controller that gives a closed loop system with complex poles having damping ratio 0.707. Use the scaled model for the computations.

(c) Compute state feedback gains using linear quadratic control. Experiment by using different weights. Compute the gains for $q_1 = q_2 = 0, q_3 = q_4 = 1$ and $\rho_1 = 0.1$ and explain the result. Choose $q_1 = q_2 = q_3 = q_4 = 1$ and explore what happens to the feedback gains and closed loop eigenvalues when you change ρ_1. Use the scaled system for this computation.

7.15 Consider the second-order system

$$\frac{d^2y}{dt^2} + 0.5\frac{dy}{dt} + y = a\frac{du}{dt} + u.$$

Let the initial conditions be zero.

(a) Show that the initial slope of the unit step response is a. Discuss what it means when $a < 0$.

(b) Show that there are points on the unit step response that are invariant with a. Discuss qualitatively the effect of the parameter a on the solution.

(c) Simulate the system and explore the effect of a on the rise time and overshoot.

7.16 (Bryson's rule) Bryson and Ho [BH75] have suggested the following method for choosing the matrices Q_x and Q_u in equation (7.26). Start by choosing Q_x and Q_u as diagonal matrices whose elements are the inverses of the squares of

the maxima of the corresponding variables. Then modify the elements to obtain a compromise among response time, damping and control effort. Apply this method to the motor drive in Exercise 7.12. Assume that the largest values of the φ_1 and φ_2 are 1, the largest values of $\dot{\varphi}_1$ and $\dot{\varphi}_2$ are 2 and the largest control signal is 10. Simulate the closed loop system for $\varphi_2(0) = 1$ and all other states are initialized to 0. Explore the effects of different values of the diagonal elements for Q_x and Q_u.

Chapter Eight
Output Feedback

One may separate the problem of physical realization into two stages: computation of the "best approximation" $\hat{x}(t_1)$ of the state from knowledge of $y(t)$ for $t \leq t_1$ and computation of $u(t_1)$ given $\hat{x}(t_1)$.

R. E. Kalman, "Contributions to the Theory of Optimal Control," 1960 [Kal60].

In this chapter we show how to use output feedback to modify the dynamics of the system through the use of observers. We introduce the concept of observability and show that if a system is observable, it is possible to recover the state from measurements of the inputs and outputs to the system. We then show how to design a controller with feedback from the observer state. A general controller with two degrees of freedom is obtained by adding feedforward. We illustrate by oulining a controller for a nonlinear system which also employs gain scheduling.

8.1 Observability

In Section 7.2 of the previous chapter it was shown that it is possible to find a state feedback law that gives desired closed loop eigenvalues provided that the system is reachable and that all the states are measured by sensors. For many situations, it is highly unrealistic to assume that all the states are measured. In this section we investigate how the state can be estimated by using a mathematical model and a few measurements. It will be shown that computation of the states can be carried out by a dynamical system called an *observer*.

Definition of Observability

Consider a system described by a set of differential equations

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du, \tag{8.1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^p$ the input and $y \in \mathbb{R}^q$ the measured output. We wish to estimate the state of the system from its inputs and outputs, as illustrated in Figure 8.1. In some situations we will assume that there is only one measured signal, i.e., that the signal y is a scalar and that C is a (row) vector. This signal may be corrupted by noise w, although we shall start by considering the noise-free case. We write $\hat{x}$ for the state estimate given by the observer.

Definition 8.1 (Observability)**.** A linear system is *observable* if for every $T > 0$ it is possible to determine the state of the system $x(T)$ through measurements of $y(t)$ and $u(t)$ on the interval $[0, T]$.

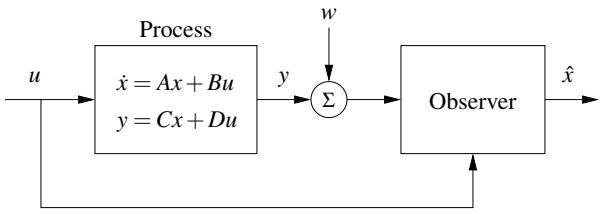


Figure 8.1: Block diagram for an observer. The observer uses the process measurement y (possibly corrupted by noise w) and the input u to estimate the current state of the process, denoted $\hat{x}$.

The definition above holds for nonlinear systems as well, and the results discussed here have extensions to the nonlinear case.

The problem of observability is one that has many important applications, even outside feedback systems. If a system is observable, then there are no "hidden" dynamics inside it; we can understand everything that is going on through observation (over time) of the inputs and outputs. As we shall see, the problem of observability is of significant practical interest because it will determine if a set of sensors is sufficient for controlling a system. Sensors combined with a mathematical model of the system can also be viewed as a "virtual sensor" that gives information about variables that are not measured directly. The process of reconciling signals from many sensors using mathematical models is also called *sensor fusion*.

Testing for Observability

When discussing reachability in the previous chapter, we neglected the output and focused on the state. Similarly, it is convenient here to initially neglect the input and focus on the autonomous system

$$\frac{dx}{dt} = Ax, \qquad y = Cx. \tag{8.2}$$

We wish to understand when it is possible to determine the state from observations of the output.

The output itself gives the projection of the state onto vectors that are rows of the matrix C. The observability problem can immediately be solved if the matrix C is invertible. If the matrix is not invertible, we can take derivatives of the output to obtain

$$\frac{dy}{dt} = C\frac{dx}{dt} = CAx.$$

From the derivative of the output we thus get the projection of the state on vectors

that are rows of the matrix CA. Proceeding in this way, we get at every time t

$$\begin{pmatrix} y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ y^{(n-1)}(t) \end{pmatrix} = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} x(t). \tag{8.3}$$

We thus find that the state at time t can be determined from the output and its derivatives at time t if the *observability matrix*

$$W_o = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} \tag{8.4}$$

has n independent rows. As in the case of reachability, it turns out that we need not consider any derivatives higher than $n-1$ (this is an application of the Cayley–Hamilton theorem [Exercise 7.10]).

The calculation can easily be extended to systems with inputs. The state is then given by a linear combination of inputs and outputs and their higher derivatives. The observability criterion is unchanged. We leave this case as an exercise for the reader.

In practice, differentiation of the output can give large errors when there is measurement noise, and therefore the method sketched above is not particularly practical. We will address this issue in more detail in the next section, but for now we have the following basic result.

Theorem 8.1 (Observability rank condition)**.** *A linear system of the form* (8.1) *is observable if and only if the observability matrix W_o is invertible (full rank).*

Proof. The sufficiency of the observability rank condition follows from the analysis above. To prove necessity, suppose that the system is observable but W_o is not invertible. Let $v \in \mathbb{R}^n$, $v \neq 0$, be a vector in the null space of W_o, so that $W_o v = 0$. If we let $x(0) = v$ be the initial condition for the system and choose $u = 0$, then the output is given by $y(t) = Ce^{At}v$. Since e^{At} can be written as a power series in A and since A^n and higher powers can be rewritten in terms of lower powers of A (by the Cayley–Hamilton theorem), it follows that $y(t)$ will be identically zero (the reader should fill in the missing steps. However, if both the input and output of the system are 0, then a valid estimate of the state is $\hat{x} = 0$ for all time, which is clearly incorrect since $x(0) = v \neq 0$. Hence by contradiction we must have that W_o is invertible if the system is observable. □

Example 8.1 Compartment model

Consider the two-compartment model in Figure 4.18a, but assume that only the

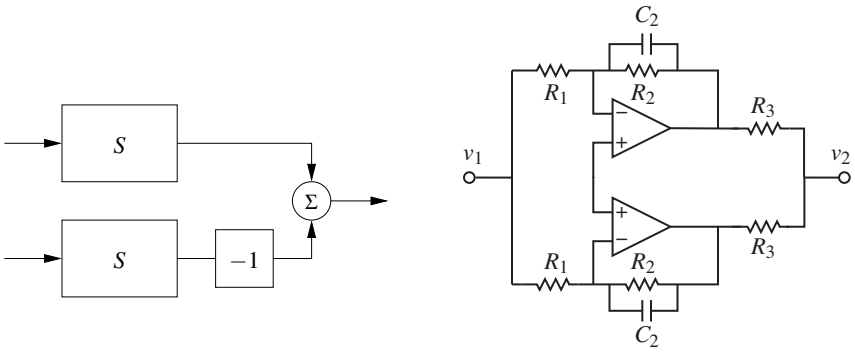


Figure 8.2: An unobservable system. Two identical subsystems have outputs that add together to form the overall system output. The individual states of the subsystem cannot be determined since the contributions of each to the output are not distinguishable. The circuit diagram on the right is an example of such a system.

concentration in the first compartment can be measured. The system is described by the linear system

$$\frac{dc}{dt} = \begin{pmatrix} -k_0 - k_1 & k_1 \\ k_2 & -k_2 \end{pmatrix} c + \begin{pmatrix} b_0 \\ 0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} c.$$

The first compartment represents the drug concentration in the blood plasma, and the second compartment the drug concentration in the tissue where it is active. To determine if it is possible to find the concentration in the tissue compartment from a measurement of blood plasma, we investigate the observability of the system by forming the observability matrix

$$W_o = \begin{pmatrix} C \\ CA \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -k_0 - k_1 & k_1 \end{pmatrix}.$$

The rows are linearly independent if $k_1 \neq 0$, and under this condition it is thus possible to determine the concentration of the drug in the active compartment from measurements of the drug concentration in the blood. ∇

It is useful to have an understanding of the mechanisms that make a system unobservable. Such a system is shown in Figure 8.2. The system is composed of two identical systems whose outputs are subtracted. It seems intuitively clear that it is not possible to deduce the states from the output since we cannot deduce the individual output contributions from the difference. This can also be seen formally (Exercise 8.2).

Observable Canonical Form

As in the case of reachability, certain canonical forms will be useful in studying observability. A linear single-input, single-output state space system is in *observ-*

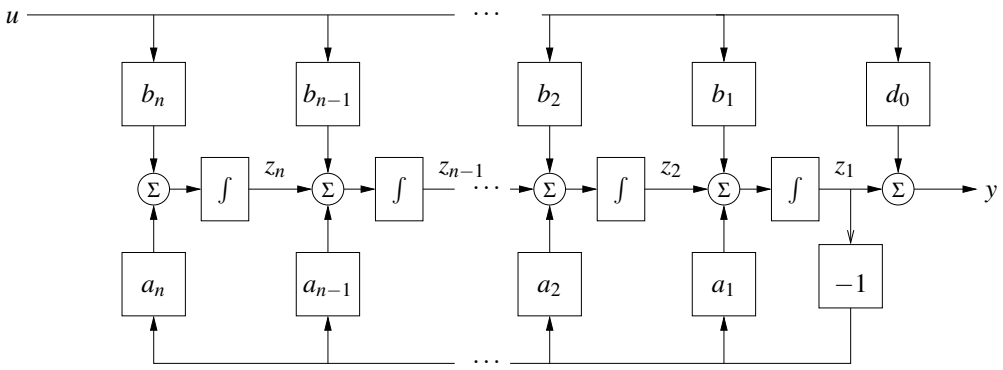


Figure 8.3: Block diagram of a system in observable canonical form. The states of the system are represented by individual integrators whose inputs are a weighted combination of the next integrator in the chain, the first state (rightmost integrator) and the system input. The output is a combination of the first state and the input.

able canonical form if its dynamics are given by

$$\frac{dz}{dt} = \begin{pmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & & 0 \\ \vdots & & & \ddots & \\ -a_{n-1} & 0 & 0 & & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{pmatrix} z + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{pmatrix} u,$$

$$y = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \end{pmatrix} z + d_0 u.$$

The definition can be extended to systems with many inputs; the only difference is that the vector multiplying u is replaced by a matrix.

Figure 8.3 is a block diagram for a system in observable canonical form. As in the case of reachable canonical form, we see that the coefficients in the system description appear directly in the block diagram. The characteristic polynomial for a system in observable canonical form is

$$\lambda(s) = s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n. \tag{8.5}$$

It is possible to reason about the observability of a system in observable canonical form by studying the block diagram. If the input u and the output y are available, the state z_1 can clearly be computed. Differentiating z_1, we obtain the input to the integrator that generates z_1, and we can now obtain $z_2 = \dot{z}_1 + a_1 z_1 - b_1 u$. Proceeding in this way, we can compute all states. The computation will, however, require that the signals be differentiated.

To check observability more formally, we compute the observability matrix for

a system in observable canonical form, which is given by

$$\tilde{W}_o = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -a_1 & 1 & 0 & \dots & 0 \\ -a_1^2 - a_2 & -a_1 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ * & * & & \dots & 1 \end{pmatrix},$$

where * represents an entry whose exact value is not important. The rows of this matrix are linearly independent (since it is lower triangular), and hence W_o is invertible. A straightforward but tedious calculation shows that the inverse of the observability matrix has a simple form given by

$$\tilde{W}_o^{-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & 1 \end{pmatrix}.$$

As in the case of reachability, it turns out that if a system is observable if and only if there always exists a transformation T that converts the system into observable canonical form. This is useful for proofs since it lets us assume that a system is in observable canonical form without any loss of generality. The observable canonical form may be poorly conditioned numerically.

8.2 State Estimation

Having defined the concept of observability, we now return to the question of how to construct an observer for a system. We will look for observers that can be represented as linear dynamical systems that take the inputs and outputs of the system we are observing and produces an estimate of the system's state. That is, we wish to construct a dynamical system of the form

$$\frac{d\hat{x}}{dt} = F\hat{x} + Gu + Hy,$$

where u and y are the input and output of the original system and $\hat{x} \in \mathbb{R}^n$ is an estimate of the state with the property that $\hat{x}(t) \to x(t)$ as $t \to \infty$.

The Observer

We consider the system in equation (8.1) with D set to zero to simplify the exposition:

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx. \tag{8.6}$$

We can attempt to determine the state simply by simulating the equations with the correct input. An estimate of the state is then given by

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu. \tag{8.7}$$

To find the properties of this estimate, introduce the estimation error $\tilde{x} = x - \hat{x}$. It follows from equations (8.6) and (8.7) that

$$\frac{d\tilde{x}}{dt} = A\tilde{x}.$$

If matrix A has all its eigenvalues in the left half-plane, the error $\tilde{x}$ will go to zero, and hence equation (8.7) is a dynamical system whose output converges to the state of the system (8.6). However, the convergence might be slower than desired.

The observer given by equation (8.7) uses only the process input u; the measured signal does not appear in the equation. We must also require that the system be stable, and essentially our estimator converges because the state of both the observer and the estimator are going to zero. This is not very useful in a control design context since we want to have our estimate converge quickly to a nonzero state so that we can make use of it in our controller. We will therefore attempt to modify the observer so that the output is used and its convergence properties can be designed to be fast relative to the system's dynamics. This version will also work for unstable systems.

Consider the observer

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}). \tag{8.8}$$

This can be considered as a generalization of equation (8.7). Feedback from the measured output is provided by adding the term $L(y - C\hat{x})$, which is proportional to the difference between the observed output and the output predicted by the observer. It follows from equations (8.6) and (8.8) that

$$\frac{d\tilde{x}}{dt} = (A - LC)\tilde{x}.$$

If the matrix L can be chosen in such a way that the matrix $A - LC$ has eigenvalues with negative real parts, the error $\tilde{x}$ will go to zero. The convergence rate is determined by an appropriate selection of the eigenvalues.

Notice the similarity between the problems of finding a state feedback and finding the observer. State feedback design by eigenvalue assignment is equivalent to finding a matrix K so that $A - BK$ has given eigenvalues. Designing an observer with prescribed eigenvalues is equivalent to finding a matrix L so that $A - LC$ has given eigenvalues. Since the eigenvalues of a matrix and its transpose are the same we can establish the following equivalences:

$$A \leftrightarrow A^T, \qquad B \leftrightarrow C^T, \qquad K \leftrightarrow L^T, \qquad W_r \leftrightarrow W_o^T.$$

The observer design problem is the *dual* of the state feedback design problem. Using the results of Theorem 7.3, we get the following theorem on observer design.

Theorem 8.2 (Observer design by eigenvalue assignment)**.** *Consider the system given by*

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx, \tag{8.9}$$

with one input and one output. Let $\lambda(s) = s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n$ *be the characteristic polynomial for A. If the system is observable, then the dynamical system*

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{8.10}$$

is an observer for the system, with L chosen as

$$L = W_o^{-1}\widetilde{W}_o \begin{pmatrix} p_1 - a_1 \\ p_2 - a_2 \\ \vdots \\ p_n - a_n \end{pmatrix} \tag{8.11}$$

and the matrices W_o *and* $\widetilde{W}_o$ *given by*

$$W_o = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}, \qquad \widetilde{W}_o = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & 1 & 0 & \cdots & 0 & 0 \\ a_2 & a_1 & 1 & & 0 & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ a_{n-2} & a_{n-3} & a_{n-4} & & 1 & 0 \\ a_{n-1} & a_{n-2} & a_{n-3} & \ldots & a_1 & 1 \end{pmatrix}^{-1}.$$

The resulting observer error $\tilde{x} = x - \hat{x}$ *is governed by a differential equation having the characteristic polynomial*

$$p(s) = s^n + p_1 s^{n-1} + \cdots + p_n.$$

The dynamical system (8.10) is called an *observer* for (the states of) the system (8.9) because it will generate an approximation of the states of the system from its inputs and outputs. This form of an observer is a much more useful form than the one given by pure differentiation in equation (8.3).

Example 8.2 Compartment model

Consider the compartment model in Example 8.1, which is characterized by the matrices

$$A = \begin{pmatrix} -k_0 - k_1 & k_1 \\ k_2 & -k_2 \end{pmatrix}, \qquad B = \begin{pmatrix} b_0 \\ 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 0 \end{pmatrix}.$$

The observability matrix was computed in Example 8.1, where we concluded that the system was observable if $k_1 \neq 0$. The dynamics matrix has the characteristic polynomial

$$\lambda(s) = \det \begin{pmatrix} s + k_0 + k_1 & -k_1 \\ -k_2 & s + k_2 \end{pmatrix} = s^2 + (k_0 + k_1 + k_2)s + k_0 k_2.$$

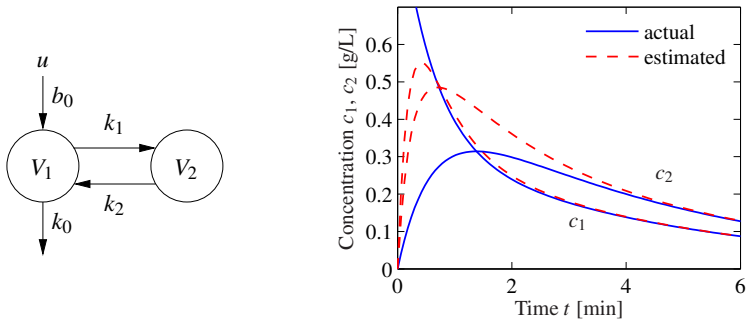


Figure 8.4: Observer for a two compartment system. A two compartment model is shown on the left. The observer measures the input concentration u and output concentration $y = c_1$ to determine the compartment concentrations, shown on the right. The true concentrations are shown by solid lines and the estimates generated by the observer by dashed lines.

Let the desired characteristic polynomial of the observer be $s^2 + p_1 s + p_2$, and equation (8.11) gives the observer gain

$$L = \begin{pmatrix} 1 & 0 \\ -k_0 - k_1 & k_1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ k_0 + k_1 + k_2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} p_1 - k_0 - k_1 - k_2 \\ p_2 - k_0 k_2 \end{pmatrix}$$
$$= \begin{pmatrix} p_1 - k_0 - k_1 - k_2 \\ (p_2 - p_1 k_2 + k_1 k_2 + k_2^2)/k_1 \end{pmatrix}.$$

Notice that the observability condition $k_1 \neq 0$ is essential. The behavior of the observer is illustrated by the simulation in Figure 8.4b. Notice how the observed concentrations approach the true concentrations. ∇

The observer is a dynamical system whose inputs are the process input u and the process output y. The rate of change of the estimate is composed of two terms. One term, $A\hat{x} + Bu$, is the rate of change computed from the model with $\hat{x}$ substituted for x. The other term, $L(y - \hat{y})$, is proportional to the difference $e = y - \hat{y}$ between measured output y and its estimate $\hat{y} = C\hat{x}$. The observer gain L is a matrix that tells how the error e is weighted and distributed among the states. The observer thus combines measurements with a dynamical model of the system. A block diagram of the observer is shown in Figure 8.5.

Computing the Observer Gain

For simple low-order problems it is convenient to introduce the elements of the observer gain L as unknown parameters and solve for the values required to give the desired characteristic polynomial, as illustrated in the following example.

Example 8.3 Vehicle steering
The normalized linear model for vehicle steering derived in Examples 6.12 and 7.4 gives the following state space model dynamics relating lateral path deviation y to

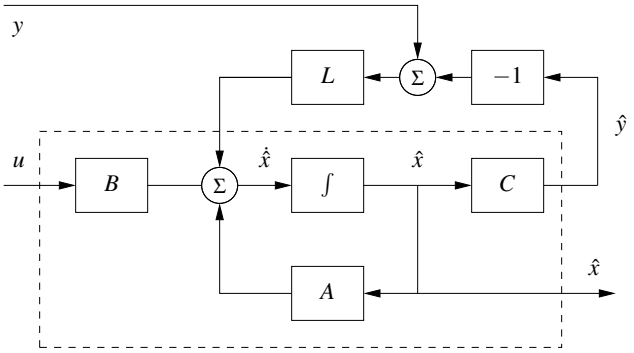


Figure 8.5: Block diagram of the observer. The observer takes the signals y and u as inputs and produces an estimate x. Notice that the observer contains a copy of the process model that is driven by $y-\hat{y}$ through the observer gain L.

steering angle u:

$$\frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} \gamma \\ 1 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} x. \tag{8.12}$$

Recall that the state x_1 represents the lateral path deviation and that x_2 represents the turning rate. We will now derive an observer that uses the system model to determine the turning rate from the measured path deviation.

The observability matrix is

$$W_o = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

i.e., the identity matrix. The system is thus observable, and the eigenvalue assignment problem can be solved. We have

$$A - LC = \begin{pmatrix} -l_1 & 1 \\ -l_2 & 0 \end{pmatrix},$$

which has the characteristic polynomial

$$\det(sI - A + LC) = \det \begin{pmatrix} s+l_1 & -1 \\ l_2 & s \end{pmatrix} = s^2 + l_1 s + l_2.$$

Assuming that we want to have an observer with the characteristic polynomial

$$s^2 + p_1 s + p_2 = s^2 + 2\zeta_o \omega_o s + \omega_o^2,$$

the observer gains should be chosen as

$$l_1 = p_1 = 2\zeta_o \omega_o, \qquad l_2 = p_2 = \omega_o^2.$$

The observer is then

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \hat{x} + \begin{pmatrix} \gamma \\ 1 \end{pmatrix} u + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} (y - \hat{x}_1).$$

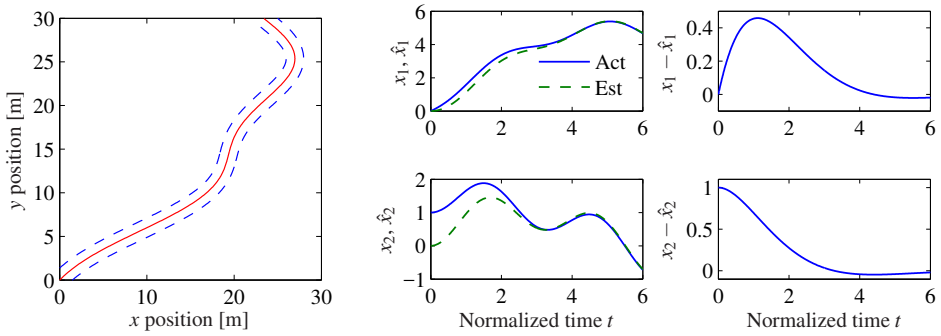


Figure 8.6: Simulation of an observer for a vehicle driving on a curvy road (left). The observer has an initial velocity error. The plots on the middle show the lateral deviation x_1, the lateral velocity x_2 by solid lines and their estimates $\hat{x}_1$ and $\hat{x}_2$ by dashed lines. The plots on the right show the estimation errors. The parameters used to design the estimator were $\omega_o = 1$ and $\zeta_o = 0.7$.

A simulation of the observer for a vehicle driving on a curvy road is simulated in Figure 8.6. The vehicle length is the time unit in the normalized model. The figure shows that the observer error settles in about 3 vehicle lengths. ∇

For systems of high order we have to use numerical calculations. The duality between the design of a state feedback and the design of an observer means that the computer algorithms for state feedback can also be used for the observer design; we simply use the transpose of the dynamics matrix and the output matrix. The MATLAB command `acker`, which essentially is a direct implementation of the calculations given in Theorem 8.2, can be used for systems with one output. The MATLAB command `place` can be used for systems with many outputs. It is also better conditioned numerically.

8.3 Control Using Estimated State

In this section we will consider a state space system of the form

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx. \tag{8.13}$$

Notice that we have assumed that there is no direct term in the system ($D = 0$). This is often a realistic assumption. The presence of a direct term in combination with a controller having proportional action creates an algebraic loop, which will be discussed in Section 9.4. The problem can be solved even if there is a direct term, but the calculations are more complicated.

We wish to design a feedback controller for the system where only the output is measured. As before, we will assume that u and y are scalars. We also assume that the system is reachable and observable. In Chapter 7 we found a feedback of the form

$$u = -Kx + k_r r$$

for the case that all states could be measured, and in Section 8.2 we developed an observer that can generate estimates of the state $\hat{x}$ based on inputs and outputs. In this section we will combine the ideas of these sections to find a feedback that gives desired closed loop eigenvalues for systems where only outputs are available for feedback.

If all states are not measurable, it seems reasonable to try the feedback

$$u = -K\hat{x} + k_r r, \tag{8.14}$$

where $\hat{x}$ is the output of an observer of the state, i.e.,

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}). \tag{8.15}$$

It is not clear that such a combination will have the desired effect. To explore this, note that since the system (8.13) and the observer (8.15) are both of state dimension n, the closed loop system has state dimension $2n$ with state $(x, \hat{x})$. The evolution of the states is described by equations (8.13)–(8.15). To analyze the closed loop system, we change coordinates and replace the estimated state variable $\hat{x}$ by the estimation error

$$\tilde{x} = x - \hat{x}. \tag{8.16}$$

Subtraction of equation (8.15) from equation (8.13) gives

$$\frac{d\tilde{x}}{dt} = Ax - A\hat{x} - L(Cx - C\hat{x}) = A\tilde{x} - LC\tilde{x} = (A - LC)\tilde{x}.$$

Returning to the process dynamics, introducing u from equation (8.14) into equation (8.13) and using equation (8.16) to eliminate $\hat{x}$ gives

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu = Ax - BK\hat{x} + Bk_r r = Ax - BK(x - \tilde{x}) + Bk_r r \\ &= (A - BK)x + BK\tilde{x} + Bk_r r.\end{aligned}$$

The closed loop system is thus governed by

$$\frac{d}{dt}\begin{pmatrix} x \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} A - BK & BK \\ 0 & A - LC \end{pmatrix}\begin{pmatrix} x \\ \tilde{x} \end{pmatrix} + \begin{pmatrix} Bk_r \\ 0 \end{pmatrix} r. \tag{8.17}$$

Notice that the state $\tilde{x}$, representing the observer error, is not affected by the reference signal r. This is desirable since we do not want the reference signal to generate observer errors.

Since the dynamics matrix is block diagonal, we find that the characteristic polynomial of the closed loop system is

$$\lambda(s) = \det(sI - A + BK)\det(sI - A + LC).$$

This polynomial is a product of two terms: the characteristic polynomial of the closed loop system obtained with state feedback and the characteristic polynomial of the observer error. The feedback (8.14) that was motivated heuristically thus provides a neat solution to the eigenvalue assignment problem. The result is summarized as follows.

Theorem 8.3 (Eigenvalue assignment by output feedback)**.** *Consider the system*

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx.$$

The controller described by

$$\begin{aligned} \frac{d\hat{x}}{dt} &= A\hat{x} + Bu + L(y - C\hat{x}) = (A - BK - LC)\hat{x} + Bk_r r + Ly, \\ u &= -K\hat{x} + k_r r \end{aligned}$$

gives a closed loop system with the characteristic polynomial

$$\lambda(s) = \det(sI - A + BK)\det(sI - A + LC).$$

This polynomial can be assigned arbitrary roots if the system is reachable and observable.

The controller has a strong intuitive appeal: it can be thought of as being composed of two parts: state feedback and an observer. The controller is now a dynamical systems with internal state dynamics generated by the observer. The control action makes use of feedback from the estimated states $\hat{x}$. The feedback gain K can be computed as if all state variables can be measured, and it depends on only A and B. The observer gain L depends on only A and C. The property that the eigenvalue assignment for output feedback can be separated into an eigenvalue assignment for a state feedback and an observer is called the *separation principle*.

A block diagram of the controller is shown in Figure 8.7. Notice that the controller contains a dynamical model of the plant. This is called the *internal model principle*: the controller contains a model of the process being controlled.

Example 8.4 Vehicle steering

Consider again the normalized linear model for vehicle steering in Example 7.4. The dynamics relating the steering angle u to the lateral path deviation y is given by the state space model (8.12). Combining the state feedback derived in Example 7.4 with the observer determined in Example 8.3, we find that the controller is given by

$$\begin{aligned} \frac{d\hat{x}}{dt} &= A\hat{x} + Bu + L(y - C\hat{x}) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \hat{x} + \begin{pmatrix} \gamma \\ 1 \end{pmatrix} u + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} (y - \hat{x}_1), \\ u &= -K\hat{x} + k_r r = k_1(r - \hat{x}_1) - k_2\hat{x}_2. \end{aligned}$$

Elimination of the variable u gives

$$\begin{aligned} \frac{d\hat{x}}{dt} &= (A - BK - LC)\hat{x} + Ly + Bk_r r \\ &= \begin{pmatrix} -l_1 - \gamma k_1 & 1 - \gamma k_2 \\ -k_1 - l_2 & -k_2 \end{pmatrix} \hat{x} + \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} y + \begin{pmatrix} \gamma \\ 1 \end{pmatrix} k_1 r, \end{aligned}$$

where we have set $k_r = k_1$ as described in Example 7.4. The controller is a dynamical system of second order, with two inputs y and r and one output u. Figure 8.8

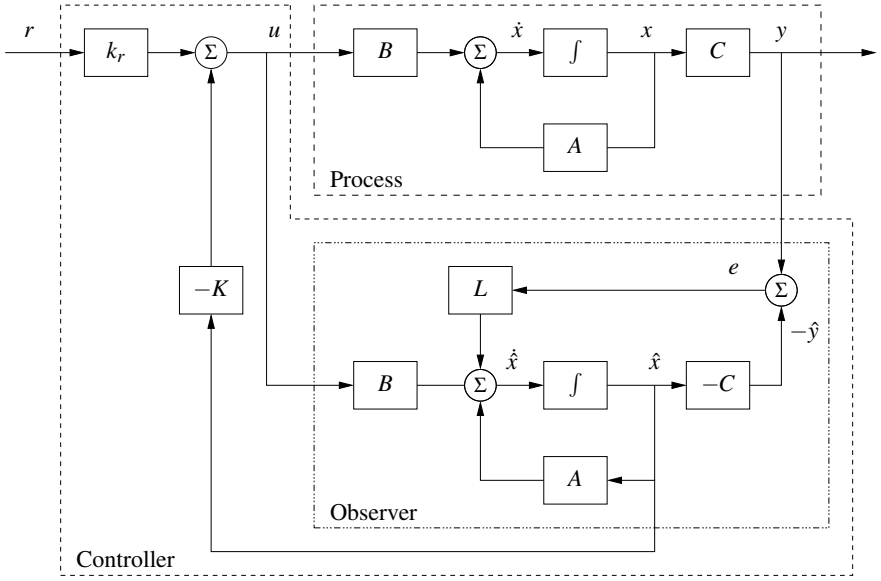


Figure 8.7: Block diagram of an observer-based control system. The observer uses the measured output y and the input u to construct an estimate of the state. This estimate is used by a state feedback controller to generate the corrective input. The controller consists of the observer and the state feedback; the observer is identical to that in Figure 8.5.

shows a simulation of the system when the vehicle is driven along a curvy road. Since we are using a normalized model, the length unit is the vehicle length and the time unit is the time it takes to travel one vehicle length. The estimator is initialized with all states equal to zero but the real system has an initial velocity of 0.5. The figures show that the estimates converge quickly to their true values. The vehicle tracks the desired path, which is in the middle of the road, but there are errors because the road is irregular. The tracking error can be improved by introducing feedforward (Section 8.5). ∇

Kalman's Decomposition of a Linear System

In this chapter and the previous one we have seen that two fundamental properties of a linear input/output system are reachability and observability. It turns out that these two properties can be used to classify the dynamics of a system. The key result is Kalman's decomposition theorem, which says that a linear system can be divided into four subsystems: Σ_{ro} which is reachable and observable, $\Sigma_{r\bar{o}}$ which is reachable but not observable, $\Sigma_{\bar{r}o}$ which is not reachable but is observable and $\Sigma_{\bar{r}\bar{o}}$ which is neither reachable nor observable.

We will first consider this in the special case of systems where the matrix A has distinct eigenvalues. In this case we can find a set of coordinates such that the A matrix is diagonal and, with some additional reordering of the states, the system

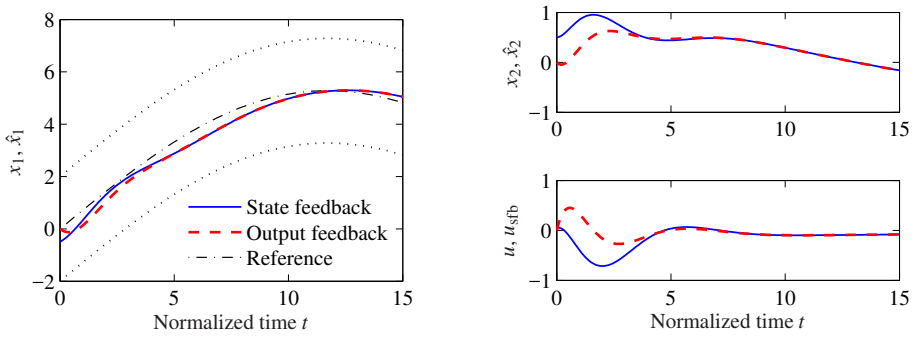


Figure 8.8: Simulation of a vehicle driving on a curvy road with a controller based on state feedback and an observer. The left plot shows the lane boundaries (dotted), the vehicle position (solid) and its estimate (dashed), the upper right plot shows the velocity (solid) and its estimate (dashed), and the lower right plot shows the control signal using state feedback (solid) and the control signal using the estimated state (dashed).

can be written as

$$\begin{aligned}\frac{dx}{dt} &= \begin{pmatrix} A_{ro} & 0 & 0 & 0 \\ 0 & A_{r\bar{o}} & 0 & 0 \\ 0 & 0 & A_{\bar{r}o} & 0 \\ 0 & 0 & 0 & A_{\bar{r}\bar{o}} \end{pmatrix} x + \begin{pmatrix} B_{ro} \\ B_{r\bar{o}} \\ 0 \\ 0 \end{pmatrix} u, \\ y &= \begin{pmatrix} C_{ro} & 0 & C_{\bar{r}o} & 0 \end{pmatrix} x + Du. \end{aligned} \tag{8.18}$$

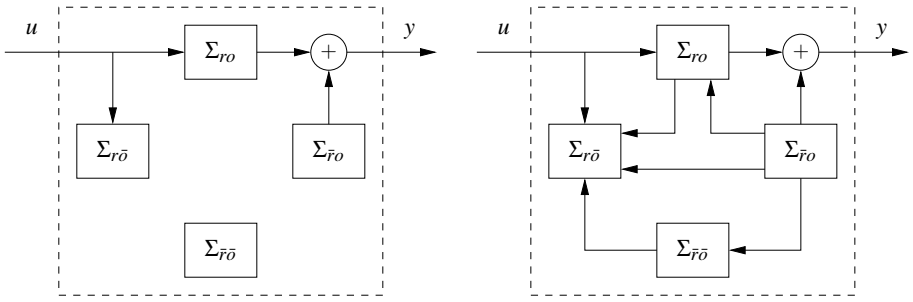
All states x_k such that $B_k \neq 0$ are reachable, and all states such that $C_k \neq 0$ are observable. If we set the initial state to zero (or equivalently look at the steady-state response if A is stable), the states given by $x_{\bar{r}o}$ and $x_{\bar{r}\bar{o}}$ will be zero and $x_{r\bar{o}}$ does not affect the output. Hence the output y can be determined from the system

$$\frac{dx_{ro}}{dt} = A_{ro}x_{ro} + B_{ro}u, \qquad y = C_{ro}x_{ro} + Du.$$

Thus from the input/output point of view, it is only the reachable and observable dynamics that matter. A block diagram of the system illustrating this property is given in Figure 8.9a.

The general case of the Kalman decomposition is more complicated and requires some additional linear algebra; see the original paper by Kalman, Ho and Narendra [KHN63]. The key result is that the state space can still be decomposed into four parts, but there will be additional coupling so that the equations have the form

$$\begin{aligned}\frac{dx}{dt} &= \begin{pmatrix} A_{ro} & 0 & * & 0 \\ * & A_{r\bar{o}} & * & * \\ 0 & 0 & A_{\bar{r}o} & 0 \\ 0 & 0 & * & A_{\bar{r}\bar{o}} \end{pmatrix} x + \begin{pmatrix} B_{ro} \\ B_{r\bar{o}} \\ 0 \\ 0 \end{pmatrix} u, \\ y &= \begin{pmatrix} C_{ro} & 0 & C_{\bar{r}o} & 0 \end{pmatrix} x, \end{aligned} \tag{8.19}$$



(a) Distinct eigenvalues (b) General case

Figure 8.9: Kalman's decomposition of a linear system. The decomposition in (a) is for a system with distinct eigenvalues and the one in (b) is the general case. The system is broken into four subsystems, representing the various combinations of reachable and observable states. The input/output relationship only depends on the subset of states that are both reachable and observable.

where $*$ denotes block matrices of appropriate dimensions. If $x_{\bar{r}o}(0) = 0$ then the input/output response of the system is given by

$$\frac{dx_{ro}}{dt} = A_{ro}x_{ro} + B_{ro}u, \qquad y = C_{ro}x_{ro} + Du, \tag{8.20}$$

which are the dynamics of the reachable and observable subsystem Σ_{ro}. A block diagram of the system is shown in Figure 8.9b.

The following example illustrates Kalman's decomposition.

Example 8.5 System and controller with feedback from observer states
Consider the system

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx.$$

The following controller, based on feedback from the observer state, was given in Theorem 8.3:

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}), \qquad u = -K\hat{x} + k_r r.$$

Introducing the states x and $\tilde{x} = x - \hat{x}$, the closed loop system can be written as

$$\frac{d}{dt}\begin{pmatrix} x \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} A - BK & BK \\ 0 & A - LC \end{pmatrix}\begin{pmatrix} x \\ \tilde{x} \end{pmatrix} + \begin{pmatrix} Bk_r \\ 0 \end{pmatrix} r, \qquad y = \begin{pmatrix} C & 0 \end{pmatrix}\begin{pmatrix} x \\ \tilde{x} \end{pmatrix},$$

which is a Kalman decomposition like the one shown in Figure 8.9b with only two subsystems Σ_{ro} and $\Sigma_{\bar{r}o}$. The subsystem Σ_{ro}, with state x, is reachable and observable, and the subsystem $\Sigma_{\bar{r}o}$, with state $\tilde{x}$, is not reachable but observable. It is natural that the state $\tilde{x}$ is not reachable from the reference signal r because it would not make sense to design a system where changes in the command signal could generate observer errors. The relationship between the reference r and the

output y is given by

$$\frac{dx}{dt} = (A - BK)x + Bk_r r, \qquad y = Cx,$$

which is the same relationship as for a system with full state feedback. ∇

Cancellation of poles and zeros was a controversy when systems were represented by transfer functions. It was manifested in rules for manipulating transfer functions: factors with roots in the right half plane can be canceled. Kalman's decompositon gives the clear insight that common factors in a transfer function indicates lack of reachability or observability.

8.4 Kalman Filtering

One of the principal uses of observers in practice is to estimate the state of a system in the presence of *noisy* measurements. We have not yet treated noise in our analysis, and a full treatment of stochastic dynamical systems is beyond the scope of this text. In this section, we present a brief introduction to the use of stochastic systems analysis for constructing observers. We work primarily in discrete time to avoid some of the complications associated with continuous-time random processes and to keep the mathematical prerequisites to a minimum. This section assumes basic knowledge of random variables and stochastic processes; see Kumar and Varaiya [KV86] or Åström [Åst06] for the required material.

Consider a discrete-time linear system with dynamics

$$x[k+1] = Ax[k] + Bu[k] + Fv[k], \qquad y[k] = Cx[k] + w[k], \tag{8.21}$$

where $v[k]$ and $w[k]$ are Gaussian white noise processes satisfying

$$\mathbb{E}(v[k]) = 0, \qquad \mathbb{E}(w[k]) = 0,$$

$$\mathbb{E}(v[k]v^T[j]) = \begin{cases} 0 & k \neq j \\ R_v & k = j, \end{cases} \qquad \mathbb{E}(w[k]w^T[j]) = \begin{cases} 0 & k \neq j \\ R_w & k = j, \end{cases} \tag{8.22}$$

$$\mathbb{E}(v[k]w^T[j]) = 0.$$

$\mathbb{E}(v[k])$ represents the expected value of $v[k]$ and $\mathbb{E}(v[k]v^T[j])$ is the correlation matrix. The matrices R_v and R_w are the covariance matrices for the process disturbance v and measurement noise w. We assume that the initial condition is also modeled as a Gaussian random variable with

$$\mathbb{E}(x[0]) = x_0, \qquad \mathbb{E}(x[0]x^T[0]) = P_0. \tag{8.23}$$

We would like to find an estimate $\hat{x}[k]$ that minimizes the mean square error $\mathbb{E}((x[k] - \hat{x}[k])(x[k] - \hat{x}[k])^T)$ given the measurements $\{y(\tau) : 0 \leq \tau \leq t\}$. We consider an observer in the same basic form as derived previously:

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] + L[k](y[k] - C\hat{x}[k]). \tag{8.24}$$

The following theorem summarizes the main result.

Theorem 8.4 (Kalman, 1961)**.** *Consider a random process $x[k]$ with dynamics given by equation* (8.21) *and noise processes and initial conditions described by equations* (8.22) *and* (8.23)*. The observer gain L that minimizes the mean square error is given by*

$$L[k] = AP[k]C^T(R_w + CP[k]C^T)^{-1},$$

where

$$\begin{aligned} P[k+1] &= (A-LC)P[k](A-LC)^T + FR_vF^T + LR_wL^T \\ P_0 &= \mathbb{E}(x[0]x^T[0]). \end{aligned} \tag{8.25}$$

Before we prove this result, we reflect on its form and function. First, note that the Kalman filter has the form of a *recursive* filter: given mean square error $P[k] = \mathbb{E}((x[k]-\hat{x}[k])(x[k]-\hat{x}[k])^T)$ at time k, we can compute how the estimate and error *change*. Thus we do not need to keep track of old values of the output. Furthermore, the Kalman filter gives the estimate $\hat{x}[k]$ *and* the error covariance $P[k]$, so we can see how reliable the estimate is. It can also be shown that the Kalman filter extracts the maximum possible information about output data. If we form the residual between the measured output and the estimated output,

$$e[k] = y[k] - C\hat{x}[k],$$

we can show that for the Kalman filter the correlation matrix is

$$R_e(j,k) = \mathbb{E}(e[j]e^T[k]) = W[k]\delta_{jk}, \qquad \delta_{jk} = \begin{cases} 1 & j = k \\ 0 & j \neq k. \end{cases}$$

In other words, the error is a white noise process, so there is no remaining dynamic information content in the error.

The Kalman filter is extremely versatile and can be used even if the process, noise or disturbances are time-varying. When the system is time-invariant and *if* $P[k]$ converges, then the observer gain is constant:

$$L = APC^T(R_w + CPC^T),$$

where P satisfies

$$P = APA^T + FR_vF^T - APC^T\left(R_w + CPC^T\right)^{-1}CPA^T.$$

We see that the optimal gain depends on both the process noise and the measurement noise, but in a nontrivial way. Like the use of LQR to choose state feedback gains, the Kalman filter permits a systematic derivation of the observer gains given a description of the noise processes. The solution for the constant gain case is solved by the `dlqe` command in MATLAB.

Proof of theorem. We wish to minimize the mean square of the error $\mathbb{E}((x[k] - \hat{x}[k])(x[k]-\hat{x}[k])^T)$. We will define this quantity as $P[k]$ and then show that it sat-

isfies the recursion given in equation (8.25). By definition,

$$\begin{aligned} P[k+1] &= \mathbb{E}((x[k+1]-\hat{x}[k+1])(x[k+1]-\hat{x}[k+1])^T) \\ &= (A-LC)P[k](A-LC)^T + FR_vF^T + LR_wL^T \\ &= AP[k]A^T + FR_vF^T - AP[k]C^TL^T - LCP[k]A^T \\ &\quad + L(R_w + CP[k]C^T)L^T. \end{aligned}$$

Letting $R_\varepsilon = (R_w + CP[k]C^T)$, we have

$$\begin{aligned} P[k+1] &= AP[k]A^T + FR_vF^T - AP[k]C^TL^T - LCP[k]A^T + LR_\varepsilon L^T \\ &= AP[k]A^T + FR_vF^T + \left(L - AP[k]C^TR_\varepsilon^{-1}\right)R_\varepsilon\left(L - AP[k]C^TR_\varepsilon^{-1}\right)^T \\ &\quad - AP[k]C^TR_\varepsilon^{-1}CP^T[k]A^T. \end{aligned}$$

To minimize this expression, we choose $L = AP[k]C^TR_\varepsilon^{-1}$, and the theorem is proved. □

The Kalman filter can also be applied to continuous-time stochastic processes. The mathematical derivation of this result requires more sophisticated tools, but the final form of the estimator is relatively straightforward.

Consider a continuous stochastic system

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu + Fv, & \mathbb{E}(v(s)v^T(t)) &= R_v(t)\delta(t-s), \\ y &= Cx + w, & \mathbb{E}(w(s)w^T(t)) &= R_w(t)\delta(t-s), \end{aligned}$$

where $\delta(\tau)$ is the unit impulse function. Assume that the disturbance v and noise w are zero mean and Gaussian (but not necessarily time-invariant):

$$\text{pdf}(v) = \frac{1}{\sqrt[n]{2\pi}\sqrt{\det R_v}} e^{-\frac{1}{2}v^TR_v^{-1}v}, \quad \text{pdf}(w) = \frac{1}{\sqrt[n]{2\pi}\sqrt{\det R_w}} e^{-\frac{1}{2}w^TR_w^{-1}w}.$$

We wish to find the estimate $\hat{x}(t)$ that minimizes the mean square error $\mathbb{E}((x(t)-\hat{x}(t))(x(t)-\hat{x}(t))^T)$ given $\{y(\tau) : 0 \le \tau \le t\}$.

Theorem 8.5 (Kalman–Bucy, 1961)**.** *The optimal estimator has the form of a linear observer*

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}),$$

where $L(t) = P(t)C^TR_w^{-1}$ *and* $P(t) = \mathbb{E}((x(t)-\hat{x}(t))(x(t)-\hat{x}(t))^T)$ *and satisfies*

$$\frac{dP}{dt} = AP + PA^T - PC^TR_w^{-1}(t)CP + FR_v(t)F^T, \quad P[0] = \mathbb{E}(x[0]x^T[0]).$$

As in the discrete case, when the system is time-invariant and if $P(t)$ converges, the observer gain is constant:

$$L = PC^TR_w^{-1} \qquad \text{where} \quad AP + PA^T - PC^TR_w^{-1}CP + FR_vF^T = 0.$$

The second equation is the *algebraic Riccati equation*.

Example 8.6 Vectored thrust aircraft

We consider the lateral dynamics of the system, consisting of the subsystems whose states are given by $z = (x, \theta, \dot{x}, \dot{\theta})$. The dynamics of the system can be obtained from Example 7.8 by extracting only the relevant states and outputs, giving

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -g & -c/m & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ r/J \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}.$$

To design a Kalman filter for the system, we must include a description of the process disturbances and the sensor noise. We thus augment the system to have the form

$$\frac{dz}{dt} = Az + Bu + Fv, \qquad y = Cz + w,$$

where F represents the structure of the disturbances (including the effects of nonlinearities that we have ignored in the linearization), v represents the disturbance source (modeled as zero mean, Gaussian white noise) and w represents that measurement noise (also zero mean, Gaussian and white).

For this example, we choose F as the identity matrix and choose disturbances v_i, $i = 1, \ldots, n$, to be independent random variables with covariance given by $R_{ii} = 0.1$, $R_{ij} = 0$, $i \neq j$. The sensor noise is a single random variable which we model as white noise having covariance $R_w = 10^{-4}$. Using the same parameters as before, the resulting Kalman gain is given by

$$L = \begin{pmatrix} 37.0 \\ -46.9 \\ 185 \\ -31.6 \end{pmatrix}.$$

The performance of the estimator is shown in Figure 8.10a. We see that while the estimator converges to the system state, it contains significant overshoot in the state estimate, which can lead to poor performance in a closed loop setting.

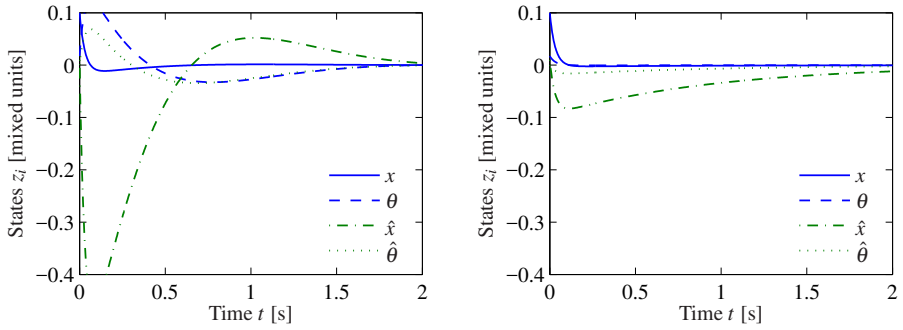
To improve the performance of the estimator, we explore the impact of adding a new output measurement. Suppose that instead of measuring just the output position x, we also measure the orientation of the aircraft θ. The output becomes

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} z + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix},$$

and if we assume that w_1 and w_2 are independent white noise sources each with covariance $R_{w_i} = 10^{-4}$, then the optimal estimator gain matrix becomes

$$L = \begin{pmatrix} 32.6 & -0.150 \\ -0.150 & 32.6 \\ 32.7 & -9.79 \\ -0.0033 & 31.6 \end{pmatrix}.$$

These gains provide good immunity to noise and high performance, as illustrated



(a) Position measurement only (b) Position and orientation

Figure 8.10: Kalman filter design for a vectored thrust aircraft. In the first design (a) only the lateral position of the aircraft is measured. Adding a direct measurement of the roll angle produces a much better observer (b). The initial condition for both simulations is $(0.1, 0.0175, 0.01, 0)$ and the controller gains are .

in Figure 8.10b. Note that the system trajectories are different in (a) and (b) since different estimates are used in each case (this difference is most noticeable in the θ trajectory. ∇

8.5 State Space Design

State estimators and state feedback are important components of a controller. In this section, we will add feedforward to arrive at a general controller structure that appears in many places in control theory and is the heart of most modern control systems. We will also briefly sketch how computers can be used to implement a controller based on output feedback.

Controller Architecture

In this chapter and the previous one we have emphasized feedback as a mechanism for minimizing tracking error; reference values were introduced simply by adding them to the state feedback through a gain k_r. A more sophisticated way of doing this is shown by the block diagram in Figure 8.11, where the controller consists of three parts: an observer that computes estimates of the states based on a model and measured process inputs and outputs, a state feedback, and a trajectory generator that generates the desired behavior of all states x_d and a feedforward signal u_{ff}. Under the ideal conditions of no disturbances and no modeling errors the signal u_{ff} generates the desired behavior x_d when applied to the process. The signals x_d and u_{ff} can be generated from the reference input r by making use of the model of the process dynamics.

To get some insight into the behavior of the system, we assume that there are no disturbances and that the system is in equilibrium with a constant reference signal

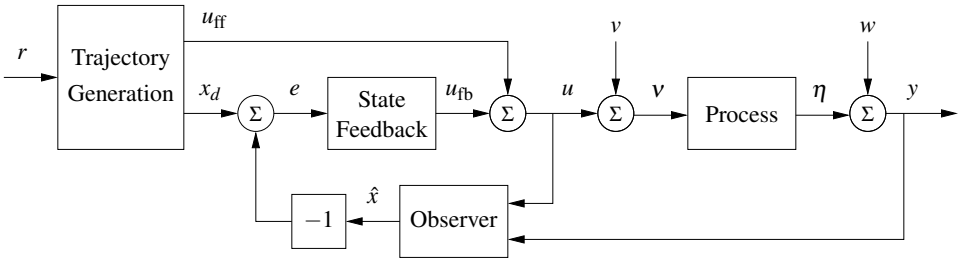


Figure 8.11: Block diagram of a controller based on a structure with two degrees of freedom that combines feedback and feedforward. The controller consists of a trajectory generator, state feedback and an observer. The trajectory generation subsystem computes a feedforward command $u_{\rm ff}$ along with the desired state x_d. The state feedback controller uses the estimated state and desired state to compute a corrective input $u_{\rm fb}$.

and with the observer state $\hat{x}$ equal to the process state x. When the reference signal is changed, the signals $u_{\rm ff}$ and x_d will change. The observer tracks the state perfectly because the initial state was correct. The estimated state $\hat{x}$ is thus equal to the desired state x_d, and the feedback signal $u_{\rm fb} = K(x_d - \hat{x})$ will also be zero. All action is thus created by the signals from the trajectory generator. If there are some disturbances or some modeling errors, the feedback signal will attempt to correct the situation.

This controller is said to have *two degrees of freedom* because the responses to command signals and disturbances are decoupled. Disturbance responses are governed by the observer and the state feedback, while the response to command signals is governed by the trajectory generator (feedforward).

Design of Feedforward and Gain Scheduling

We will now discuss design of controller with the architecture shown in Figure 8.11. For an analytic description we start with the full nonlinear dynamics of the process

$$\frac{dx}{dt} = f(x,u), \qquad y = h(x,u). \tag{8.26}$$

Assume that the trajectory generator is able to compute a desired trajectory $(x_d, u_{\rm ff})$ that satisfies the dynamics (8.26) and satisfies $r = h(x_d, u_{\rm ff})$. To design the controller, we construct the error system. Let $z = x - x_d$ and $v = u - u_{\rm ff}$ and compute the dynamics for the error:

$$\begin{aligned}\dot{z} &= \dot{x} - \dot{x}_d = f(x,u) - f(x_d, u_{\rm ff}) \\ &= f(z + x_d, v + u_{\rm ff}) - f(x_d, u_{\rm ff}) =: F(z, v, x_d(t), u_{\rm ff}(t)).\end{aligned}$$

In general, this system is time-varying. Note that z corresponds to $-e$ in Figure 8.11 due to the convention of using negative feedback in the block diagram.

For trajectory tracking, we can assume that e is small (if our controller is doing

a good job), and so we can linearize around $z = 0$:

$$\frac{dz}{dt} \approx A(t)z + B(t)v, \quad A(t) = \left.\frac{\partial F}{\partial z}\right|_{(x_d(t),u_{\text{ff}}(t))}, \quad B(t) = \left.\frac{\partial F}{\partial v}\right|_{(x_d(t),u_{\text{ff}}(t))}.$$

It is often the case that $A(t)$ and $B(t)$ depend only on x_d, in which case it is convenient to write $A(t) = A(x_d)$ and $B(t) = B(x_d)$.

Assume now that x_d and u_{ff} are either constant or slowly varying (with respect to the performance criterion). This allows us to consider just the (constant) linear system given by $(A(x_d), B(x_d))$. If we design a state feedback controller $K(x_d)$ for each x_d, then we can regulate the system using the feedback

$$v = -K(x_d)z.$$

Substituting back the definitions of z and v, our controller becomes

$$u = -K(x_d)(x - x_d) + u_{\text{ff}}.$$

This form of controller is called a *gain scheduled* linear controller with *feedforward* u_{ff}.

Finally, we consider the observer. The full nonlinear dynamics can be used for the prediction portion of the observer and the linearized system for the correction term:

$$\frac{d\hat{x}}{dt} = f(\hat{x}, u) + L(\hat{x})(y - h(\hat{x}, u)),$$

where $L(\hat{x})$ is the observer gain obtained by linearizing the system around the currently estimated state. This form of the observer is known as an *extended Kalman filter* and has proved to be a very effective means of estimating the state of a nonlinear system.

There are many ways to generate the feedforward signal, and there are also many different ways to compute the feedback gain K and the observer gain L. Note that once again the internal model principle applies: the controller contains a model of the system to be controlled through the observer. When a nonlinear system is modeled using differential algebraic equation as is done in Modelica, is is possible to generate the inverse model automatically.

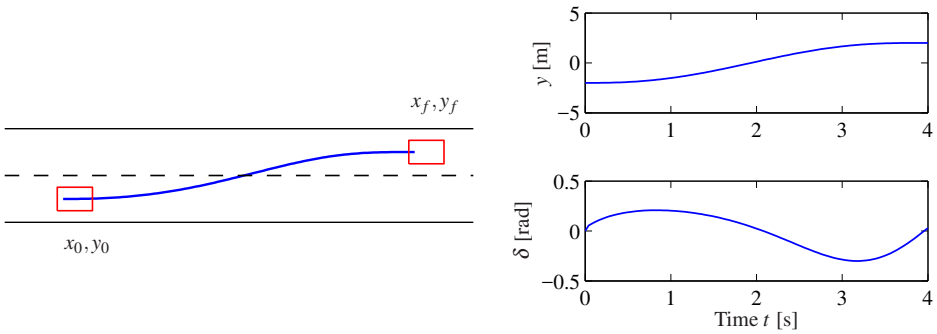
Example 8.7 Vehicle steering

To illustrate how we can use a two degree-of-freedom design to improve the performance of the system, consider the problem of steering a car to change lanes on a road, as illustrated in Figure 8.12a.

We use the non-normalized form of the dynamics, which were derived in Example 3.10. Using the center of the rear wheels as the reference ($\alpha = 0$), the dynamics can be written as

$$\frac{dx}{dt} = \cos\theta v, \qquad \frac{dy}{dt} = \sin\theta v, \qquad \frac{d\theta}{dt} = \frac{v}{b}\tan\delta,$$

where v is the forward velocity of the vehicle, θ is the heading angle and δ is the steering angle. To generate a trajectory for the system, we note that we can solve



(a) Overhead view (b) Position and steering

Figure 8.12: Trajectory generation for changing lanes. We wish to change from the left lane to the right lane over a distance of 30 m in 4 s. The planned trajectory in the xy plane is shown in (a) and the lateral position y and the steering angle δ over the maneuver time interval are shown in (b).

for the states and inputs of the system given $x(t)$, $y(t)$ by solving the following sets of equations:

$$\begin{aligned} \dot{x} &= v\cos\theta, & \ddot{x} &= \dot{v}\cos\theta - v\dot{\theta}\sin\theta, \\ \dot{y} &= v\sin\theta, & \ddot{y} &= \dot{v}\sin\theta + v\dot{\theta}\cos\theta, \\ \dot{\theta} &= (v/b)\tan\delta. \end{aligned} \tag{8.27}$$

This set of five equations has five unknowns (θ, $\dot{\theta}$, v, $\dot{v}$ and δ) that can be solved using trigonometry and linear algebra given the path variables $x(t)$, $y(t)$ and their time derivatives. It follows that we can compute a feasible state trajectory for the system given any path $x(t)$, $y(t)$. f

To find a trajectory from an initial state (x_0, y_0, θ_0) to a final state (x_f, y_f, θ_f) at a time T, we look for a path $x(t), y(t)$ that satisfies

$$\begin{aligned} x(0) &= x_0, & x(T) &= x_f, \\ y(0) &= y_0, & y(T) &= y_f, \\ \dot{x}(0)\sin\theta_0 - \dot{y}(0)\cos\theta_0 &= 0, & \dot{x}(T)\sin\theta_f - \dot{y}(T)\cos\theta_f &= 0, \\ \dot{y}(0)\sin\theta_0 + \dot{x}(0)\cos\theta_0 &= v_0, & \dot{y}(T)\sin\theta_f + \dot{x}(T)\cos\theta_f &= v_f. \end{aligned} \tag{8.28}$$

One such trajectory can be found by choosing $x(t)$ and $y(t)$ to have the form

$$x_d(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3, \qquad y_d(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3.$$

Substituting these equations into equation (8.28), we are left with a set of linear equations that can be solved for α_i, β_i, $i = 0, 1, 2, 3$. This gives a feasible trajectory for the system by using equation (8.27) to solve for θ_d, v_d and δ_d.

Figure 8.12b shows a sample trajectory generated by a set of higher-order equations that also set the initial and final steering angle to zero. Notice that the feedforward input is different from zero, allowing the controller to command a steering

angle that executes the turn in the absence of errors. ∇

Differential Flatness

The concept of differential flatness that we exploited in the previous example is a fairly general one and can be applied to many interesting trajectory generation problems. A nonlinear system (8.26) is *differentially flat* if there exists a flat output z such that the state x and the input u can be expressed as functions of the flat output z and a finite number of its derivatives

$$x = \beta(z, \dot{z}, \ldots, z^{(q)}), \qquad u = \gamma(z, \dot{z}, \ldots, z^{(q)}). \tag{8.29}$$

The number of flat outputs is always equal to the number of system inputs. The kinematic car is differentially flat with the position of the rear wheels as the flat output. For the linear system in (7.6), which is in reachable canonical form, we have

$$z_1 = z_n^{(n-1)},\ z_2 = z_n^{(n-2)}, \cdots,\ z_n = z_n,\ u = z_n^{(n)} + a_1 z_n^{(n-1)} + a_2 z_n^{(n-2)} + \cdots + a_n z_n,$$

and the n:th component z_n of the state vector is is thus a flat output.

Note that no ODEs need to be integrated in order to compute the feasible trajectories for a differentially flat system (unlike optimal control methods, which often involve parameterizing the *input* and then solving the ODEs). The practical implication is that nominal trajectories and inputs that satisfy the equations of motion for a differentially flat system can be computed efficiently.

Computer Implementation

The controllers obtained so far have been described by ordinary differential equations. They can be implemented directly using analog components, whether electronic circuits, hydraulic valves or other physical devices. Since in modern engineering applications most controllers are implemented using computers, we will briefly discuss how this can be done.

A computer-controlled system typically operates periodically: every cycle, signals from the sensors are sampled and converted to digital form by the A/D converter, the control signal is computed and the resulting output is converted to analog form for the actuators, as shown in Figure 8.13. To illustrate the main principles of how to implement feedback in this environment, we consider the controller described by equations (8.14) and (8.15), i.e.,

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}), \qquad u = -K\hat{x} + k_r r.$$

The second equation consists only of additions and multiplications and can thus be implemented directly on a computer. The first equation can be implemented by approximating the derivative by a difference

$$\frac{d\hat{x}}{dt} \approx \frac{\hat{x}(t_{k+1}) - \hat{x}(t_k)}{h} = A\hat{x}(t_k) + Bu(t_k) + L\big(y(t_k) - C\hat{x}(t_k)\big),$$

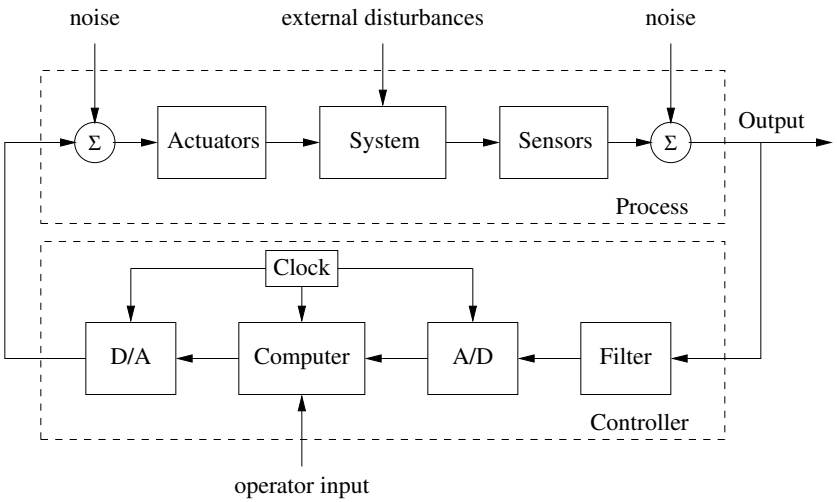


Figure 8.13: Components of a computer-controlled system. The controller consists of analog-to-digital (A/D) and digital-to-analog (D/A) converters, as well as a computer that implements the control algorithm. A system clock controls the operation of the controller, synchronizing the A/D, D/A and computing processes. The operator input is also fed to the computer as an external input.

where t_k are the sampling instants and $h = t_{k+1} - t_k$ is the sampling period. Rewriting the equation to isolate $\hat{x}(t_{k+1})$, we get the difference equation

$$\hat{x}(t_{k+1}) = \hat{x}(t_k) + h\big(A\hat{x}(t_k) + Bu(t_k) + L\big(y(t_k) - C\hat{x}(t_k)\big)\big). \tag{8.30}$$

The calculation of the estimated state at time t_{k+1} requires only addition and multiplication and can easily be done by a computer. A section of pseudocode for the program that performs this calculation is

```
% Control algorithm - main loop
r = adin(ch1)                          % read reference
y = adin(ch2)                          % get process output
u = K*(xd - xhat) + uff                % compute control variable
daout(ch1, u)                          % set analog output
xhat = xhat + h*(A*x+B*u+L*(y-C*x))    % update state estimate
```

The program runs periodically at a fixed rate h. Notice that the number of computations between reading the analog input and setting the analog output has been minimized by updating the state after the analog output has been set. The program has an array of states `xhat` that represents the state estimate. The choice of sampling period requires some care.

There are more sophisticated ways of approximating a differential equation by a difference equation. If the control signal is constant between the sampling instants, it is possible to obtain exact equations; see [ÅW97].

There are several practical issues that also must be dealt with. For example, it is necessary to filter measured signals before they are sampled so that the filtered signal has little frequency content above $f_s/2$, where f_s is the sampling frequency.

This avoids a phenomena known as *aliasing*. If controllers with integral action are used, it is also necessary to provide protection so that the integral does not become too large when the actuator saturates. This issue, called *integrator windup*, is studied in more detail in Chapter 11. Care must also be taken so that parameter changes do not cause disturbances.

8.6 Further Reading

The notion of observability is due to Kalman [Kal61b] and, combined with the dual notion of reachability, it was a major stepping stone toward establishing state space control theory beginning in the 1960s. The observer first appeared as the Kalman filter, in the paper by Kalman [Kal61a] on the discrete-time case and Kalman and Bucy [KB61] on the continuous-time case. Kalman also conjectured that the controller for output feedback could be obtained by combining a state feedback with an observer; see the quote in the beginning of this chapter. This result was formally proved by Josep and Tou [JT61] and Gunckel and Franklin [GF71]. The combined result is known as the linear quadratic Gaussian control theory; a compact treatment is given in the books by Anderson and Moore [AM90] and Åström [Åst06]. Much later it was shown that solutions to robust control problems also had a similar structure but with different ways of computing observer and state feedback gains [DGKF89]. The general controller structure discussed in Section 8.5, which combines feedback and feedforward, was described by Horowitz in 1963 [Hor63]. The particular form in Figure 8.11, which appeared in [ÅW97], which also treats digital implementation of the controller. The hypothesis that motion control in humans is based on a combination of feedback and feedforward was proposed by Ito in 1970 [Ito70]. Differentially flat systems were originally studied by Fliess et al. [FLMR92].

Exercises

8.1 (Coordinate transformations) Consider a system under a coordinate transformation $z = Tx$, where $T \in \mathbb{R}^{n\times n}$ is an invertible matrix. Show that the observability matrix for the transformed system is given by $\widetilde{W}_o = W_o T^{-1}$ and hence observability is independent of the choice of coordinates.

8.2 Show that the system depicted in Figure 8.2 is not observable.

8.3 (Observable canonical form) Show that if a system is observable, then there exists a change of coordinates $z = Tx$ that puts the transformed system into observable canonical form.

8.4 (Bicycle dynamics) The linearized model for a bicycle is given in equation (4.5), which has the form

$$J\frac{d^2\varphi}{dt^2} - \frac{Dv_0}{b}\frac{d\delta}{dt} = mgh\varphi + \frac{mv_0^2 h}{b}\delta,$$

where φ is the tilt of the bicycle and δ is the steering angle. Give conditions under which the system is observable and explain any special situations where it loses observability.

8.5 (Integral action) The model (8.1) assumes that the input $u = 0$ corresponds to $x = 0$. In practice, it is very difficult to know the value of the control signal that gives a precise value of the state or the output because this would require a perfectly calibrated system. One way to avoid this assumption is to assume that the model is given by

$$\frac{dx}{dt} = Ax + B(u + u_0), \qquad y = Cx + Du,$$

where u_0 is an unknown constant that can be modeled as $du_0/dt = 0$. Consider u_0 as an additional state variable and derive a controller based on feedback from the observed state. Show that the controller has integral action and that it does not require a perfectly calibrated system.

8.6 (Vectored thrust aircraft) The lateral dynamics of the vectored thrust aircraft example described in Example 7.8 can be obtained by considering the motion described by the states $z = (x, \theta, \dot{x}, \dot{\theta})$. Construct an estimator for these dynamics by setting the eigenvalues of the observer into a *Butterworth pattern* with $\lambda_{\text{bw}} = -3.83 \pm 9.24i$, $-9.24 \pm 3.83i$. Using this estimator combined with the state space controller computed in Example 7.8, plot the step response of the closed loop system.

8.7 (Uniqueness of observers) Show that the design of an observer by eigenvalue assignment is unique for single-output systems. Construct examples that show that the problem is not necessarily unique for systems with many outputs.

8.8 (Observers using differentiation) Consider the linear system (8.2), and assume that the observability matrix W_o is invertible. Show that

$$\hat{x} = W_o^{-1} \begin{pmatrix} y & \dot{y} & \ddot{y} & \cdots & y^{(n-1)} \end{pmatrix}^T$$

is an observer. Show that it has the advantage of giving the state instantaneously but that it also has some severe practical drawbacks.

8.9 (Observer for Teorell's compartment model) Teorell's compartment model, shown in Figure 4.17, has the following state space representation:

$$\frac{dx}{dt} = \begin{pmatrix} -k_1 & 0 & 0 & 0 & 0 \\ k_1 & -k_2 - k_4 & 0 & k_3 & 0 \\ 0 & k_4 & 0 & 0 & 0 \\ 0 & k_2 & 0 & -k_3 - k_5 & 0 \\ 0 & 0 & 0 & k_5 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} u,$$

where representative parameters are $k_1 = 0.02$, $k_2 = 0.1$, $k_3 = 0.05$, $k_4 = k_5 = 0.005$. The concentration of a drug that is active in compartment 5 is measured in

the bloodstream (compartment 2). Determine the compartments that are observable from measurement of concentration in the bloodstream and design an estimator for these concentrations base on eigenvalue assignment. Choose the closed loop eigenvalues -0.03, -0.05 and -0.1. Simulate the system when the input is a pulse injection.

8.10 (Observer design for motor drive) Consider the normalized model of the motor drive in Exercise 3.10 where the open loop system has the eigenvalues $0, 0, -0.05 \pm i$. A state feedback that gave a closed loop system with eigenvalues in -2, -1 and $-1 \pm i$ was designed in Exercise 7.12. Design an observer for the system that has eigenvalues -4, -2 and $-2 \pm 2i$. Combine the observer with the state feedback from Exercise 7.12 to obtain an output feedback and simulate the complete system.

8.11 (Feedforward design for motor drive) Consider the normalized model of the motor drive in Exercise 3.10. Design the dynamics of the block labeled "trajectory generation" in Figure 8.11 so that the dynamics relating the output η to the reference signal r has the dynamics

$$\frac{d^3 y_m}{dt^3} + a_{m1}\frac{d^2 y_m}{dt^2} + a_{m2}\frac{dy_m}{dt} + a_{m3} y_m = a_{m3} r, \tag{8.31}$$

with parameters $a_{m1} = 2.5\omega_m$, $a_{m2} = 2.5\omega_m^2$ and $a_{m3} = \omega_m^3$. Discuss how the largest value of the feedforward signal for a unit step in the command signal depends on ω_m.

8.12 (Whipple bicycle model) Consider the Whipple bicycle model given by equation (4.7) in Section 4.2. A state feedback for the system was designed in Exercise 7.13. Design an observer and an output feedback for the system.

8.13 (Discrete-time random walk) Suppose that we wish to estimate the position of a particle that is undergoing a random walk in one dimension (i.e., along a line). We model the position of the particle as

$$x[k+1] = x[k] + u[k],$$

where x is the position of the particle and u is a white noise processes with $E\{u[i]\} = 0$ and $E\{u[i]\,u[j]\} = R_u \delta(i-j)$. We assume that we can measure x subject to additive, zero-mean, Gaussian white noise with covariance 1.

(a) Compute the expected value and covariance of the particle as a function of k.

(b) Construct a Kalman filter to estimate the position of the particle given the noisy measurements of its position. Compute the steady-state expected value and covariance of the error of your estimate.

(c) Suppose that $E\{u[0]\} = \mu \neq 0$ but is otherwise unchanged. How would your answers to parts (a) and (b) change?

8.14 (Kalman decomposition) Consider a linear system characterized by the matrices

$$A = \begin{pmatrix} -2 & 1 & -1 & 2 \\ 1 & -3 & 0 & 2 \\ 1 & 1 & -4 & 2 \\ 0 & 1 & -1 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 1 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 & -1 & 0 \end{pmatrix}, \quad D = 0.$$

Construct a Kalman decomposition for the system. (Hint: Try to diagonalize.)

Chapter Nine
Transfer Functions

The typical regulator system can frequently be described, in essentials, by differential equations of no more than perhaps the second, third or fourth order. . . . In contrast, the order of the set of differential equations describing the typical negative feedback amplifier used in telephony is likely to be very much greater. As a matter of idle curiosity, I once counted to find out what the order of the set of equations in an amplifier I had just designed would have been, if I had worked with the differential equations directly. It turned out to be 55.

Hendrik Bode, 1960 [Bod60].

This chapter introduces the concept of the *transfer function*, which is a compact description of the input/output relation for a linear time invariant system. To simplify the writing we will often drop time invariant in the text. Combining transfer functions with block diagrams gives a powerful method for dealing with complex linear systems. The relationship between transfer functions and other descriptions of system dynamics is also discussed.

9.1 Frequency Domain Modeling

Figure 9.1 is a block diagram for a typical control system, consisting of a process to be controlled and a controller that combines feedback and feedforward. We saw in the previous two chapters how to analyze and design such systems using state space descriptions of the blocks. As mentioned in Chapter 3, an alternative approach is to focus on the input/output characteristics of the system. Since it is the inputs and outputs that are used to connect the systems, one could expect that this

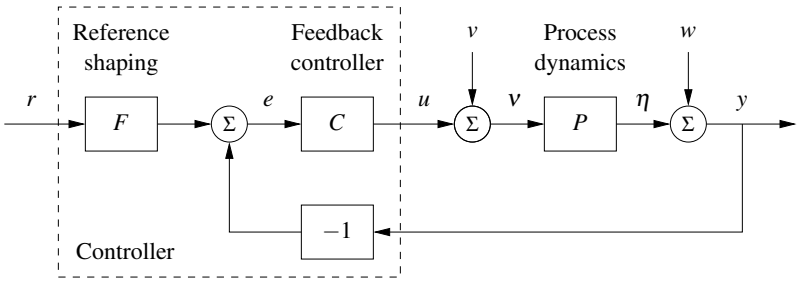


Figure 9.1: A block diagram for a feedback control system. The reference signal r is fed through a reference shaping block, which generates a signal which is compared with the output y to form the error e. The control signal u is generated by the controller which has the error as the input. The load disturbance v and the measurement noise w are external signals.

point of view would allow an understanding of the overall behavior of the system. Transfer functions are the main tool in implementing this point of view for linear systems.

The basic idea of the transfer function comes from looking at the frequency response of a system. Suppose that we have an input signal that is periodic. Then we can decompose this signal into the sum of a set of sines and cosines,

$$u(t) = \sum_{k=0}^{\infty} a_k \sin(k\omega t) + b_k \cos(k\omega t),$$

where ω is the fundamental frequency of the periodic input. As we saw in Section 6.3, the input $u(t)$ generates corresponding sine and cosine outputs (in steady state), with possibly shifted magnitude and phase. The gain and phase at each frequency are determined by the frequency response given in equation (6.24):

$$G(s) = C(sI - A)^{-1}B + D, \tag{9.1}$$

where we set $s = i(k\omega)$ for each $k = 1, \ldots, \infty$ and $i = \sqrt{-1}$. If we know the steady-state frequency response $G(s)$, we can thus compute the response to any (periodic) signal using superposition. The transfer function generalizes this notion to allow a broader class of input signals besides periodic ones. As we shall see in the next section, the transfer function represents the response of the system to an *exponential input*, $u = e^{st}$. It turns out that the form of the transfer function is precisely the same as that of equation (9.1). This should not be surprising since we derived equation (9.1) by writing sinusoids as sums of complex exponentials. The transfer function can also be introduced as is the ratio of the Laplace transforms of output and input when the state is zero, although one does not have to understand the details of Laplace transforms in order to make use of transfer functions.

Modeling a system through its response to sinusoidal and exponential signals is known as *frequency domain modeling*. This terminology stems from the fact that we represent the dynamics of the system in terms of the generalized frequency s rather than the time domain variable t. The transfer function provides a complete representation of a linear system in the frequency domain.

The power of transfer functions is that they provide a particularly convenient representation in manipulating and analyzing complex linear feedback systems. As we shall see, there are graphical representations of transfer functions (Bode and Nyquist plots) that capture interesting properties of the underlying dynamics. Transfer functions also make it possible to express the changes in a system because of modeling error, which is essential when considering sensitivity to process variations of the sort discussed in Chapter 13. More specifically, using transfer functions, it is possible to analyze what happens when dynamic models are approximated by static models or when high-order models are approximated by low-order models. One consequence is that we can introduce concepts that express the degree of stability of a system.

While many of the concepts for state space modeling and analysis apply directly to nonlinear systems, frequency domain analysis applies primarily to linear

systems. The notions of gain and phase can, however, be generalized to nonlinear systems and, in particular, propagation of sinusoidal signals through a nonlinear system can approximately be captured by an analog of the frequency response called the describing function. These extensions of frequency response will be discussed in Section 10.5.

9.2 Determining the Transfer Function

As we have seen in previous chapters, the input/output dynamics of a linear system have two components: the initial condition response and the forced response, which depends on the system input. The forced response can be completely characterized by the transfer function. In Section 2.1 we briefly introduced the transfer function as the response to an exponential function e^{st} . In this section we will compute transfer functions for general linear systems and we will show that the transfer function is indeed a complete characterization of the input/output behavior of a linear system.

Transmission of Exponential Signals

To formally compute the transfer function of a system, we will make use of a special type of signal, called an *exponential signal,* of the form e^{st}, where $s = \sigma + i\omega$ is a complex number. Exponential signals play an important role in linear systems. They appear in the solution of differential equations and in the impulse response of linear systems, and many signals can be represented as exponentials or sums of exponentials. For example, a constant signal is simply $e^{\alpha t}$ with $\alpha = 0$. Damped sine and cosine signals can be represented by

$$e^{(\sigma+i\omega)t} = e^{\sigma t}e^{i\omega t} = e^{\sigma t}(\cos\omega t + i\sin\omega t),$$

where $\sigma < 0$ determines the decay rate. Figure 9.2 gives examples of signals that can be represented by complex exponentials; many other signals can be represented by linear combinations of these signals.

As in the case of the sinusoidal signals we considered in Section 6.3, we will allow complex-valued signals in the derivation that follows, although in practice we always add together combinations of signals that result in real-valued functions.

In Section 2.1 we showed that the transfer function $G(s)$ of a linear time-invariant system can be obtained by computing a particular solution of the form $y_p(t) = G(s)e^{st}$ when the input is $u(t) = e^{st}$, where s is a complex number. For example, the system

$$\frac{d^n y}{dt^n} + a_1\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_n y = b_0\frac{d^m u}{dt^m} + b_1\frac{d^{m-1}u}{dt^{m-1}} + \cdots + b_m u,$$

has the transfer function

$$G(s) = \frac{b_0 s^m + b_1 s^{m-2} + \cdots + b_m}{s^n + a_1 s^{n-1} + \cdots + a_n};$$

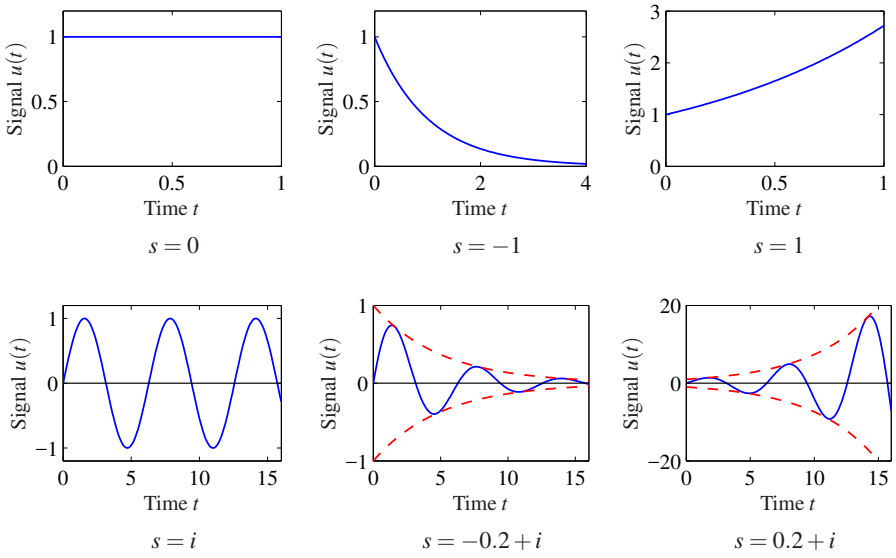


Figure 9.2: Examples of exponential signals. The top row corresponds to exponential signals with a real exponent, and the bottom row corresponds to those with complex exponents. The dashed line in the last two cases denotes the bounding envelope for the oscillatory signals. In each case, if the real part of the exponent is negative then the signal decays, while if the real part is positive then it grows.

see equations (2.2) and (2.7).

To find the transfer function for the state space system

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx + Du, \tag{9.2}$$

we let the input be the exponential signal $u(t) = e^{st}$ and assume that $s \notin \lambda(A)$. The state is then given by

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Be^{s\tau}\,d\tau = e^{At}x(0) + e^{At}(sI - A)^{-1}\left(e^{(sI-A)t} - I\right)B.$$

The output y of equation (9.2) then becomes

$$\begin{aligned} y(t) &= Cx(t) + Du(t) \\ &= \underbrace{Ce^{At}x(0)}_{\text{initial state response}} + \underbrace{\left(C(sI-A)^{-1}B + D\right)e^{st} - Ce^{At}(sI-A)^{-1}B}_{\text{input response}} \\ &= \underbrace{Ce^{At}\left(x(0) - (sI-A)^{-1}B\right)}_{\text{transient response}} + \underbrace{\left(C(sI-A)^{-1}B + D\right)e^{st}}_{\text{pure exponential response}}. \end{aligned} \tag{9.3}$$

and the *transfer function* from u to y of the system (9.2) is the coefficient of the

term e^{st}, hence

$$G(s) = C(sI - A)^{-1}B + D. \tag{9.4}$$

Compare this with the definition of frequency response given by equations (6.23) and (6.24).

An important point in the derivation of the transfer function is the fact that we have restricted s so that $s \neq \lambda_j(A)$, the eigenvalues of A. At those values of s, we see that the response (9.3) of the system is singular (since $sI - A$ then is not invertible). The transfer function can, however, be extended to all values of s by analytic continuation.

To give some insight we will now discuss the equation (9.3). We first notice that the output $y(t)$ can be separated in terms in two ways with different interpretations. The response of the system to initial conditions is $Ce^{At}x(0)$. Recall that e^{At} can be written in terms of the eigenvalues of A (using the Jordan form in the case of repeated eigenvalues), and hence the transient response is a linear combination of terms of the form $p_j(t)e^{\lambda_j t}$, where λ_j are eigenvalues of A and $p_j(t)$ is a polynomial whose degree is less than the multiplicity of the eigenvalue, see Exercise 9.2.

The response to the input $u(t) = e^{st}$ contains a mixture of terms $p_j(t)e^{\lambda_j t}$ and the exponential function

$$y_p(t) = \big(C(sI - A)^{-1}B + D\big)e^{st}, \tag{9.5}$$

which is a *particular solution* to the differential equation (9.2). We call this solution the *pure exponential solution* because had only one exponential e^{st}. It follows from equation (9.3) that the output $y(t)$ is equal to the pure exponential solution $y_p(t)$ if the initial condition is chosen as

$$x(0) = (sI - A)^{-1}B. \tag{9.6}$$

If the system (9.2) is aymptotically stable, then $e^{At} \to 0$ as $t \to \infty$. If in addition the input $u(t)$ is a constant $u(t) = e^{0 \cdot t}$ or a sinusoid $u(t) = e^{i\omega}$ the response then goes to a constant or sinusoidal *steady state solution*, see equation (6.23).

To simplify manipulation of the equations describing linear time invariant systems, we introduce $\mathscr{E}$ as the class of time functions of the form $X(s)e^{st}$, where the parameters s is a complex variable and $X(s)$ is a complex function (vector valued if f is a vector). It follows from equations (9.3, 9.4) that if a system with transfer function $G(s)$ has the input $u \in \mathscr{E}$ then there is a particular solution such that the output is $y = Gu$. This solution is the actual response of the system if the initial condition is chosen as equation (9.6). Since the transfer function of a system is given by the pure exponential response, we can derive transfer functions using exponential signals, and we will use the notation

$$y = Gu, \quad u, y \in \mathscr{E}, \tag{9.7}$$

but we must always be aware of the meaning of this notation.

Example 9.1 Damped oscillator

Consider the response of a damped linear oscillator, whose state space dynamics

were studied in Section 7.3:

$$\frac{dx}{dt} = \begin{pmatrix} 0 & \omega_0 \\ -\omega_0 & -2\zeta\omega_0 \end{pmatrix} x + \begin{pmatrix} 0 \\ k\omega_0 \end{pmatrix} u, \qquad y = \begin{pmatrix} 1 & 0 \end{pmatrix} x. \tag{9.8}$$

This system is asymptotically stable if $\zeta > 0$, and so we can look at the steady-state response to an input $u = e^{st}$,

$$\begin{aligned} G_{yu}(s) = C(sI-A)^{-1}B &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} s & -\omega_0 \\ \omega_0 & s+2\zeta\omega_0 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ k\omega_0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \end{pmatrix} \left(\frac{1}{s^2+2\zeta\omega_0 s+\omega_0^2} \begin{pmatrix} s+2\zeta\omega_0 & -\omega_0 \\ \omega_0 & s \end{pmatrix} \right) \begin{pmatrix} 0 \\ k\omega_0 \end{pmatrix} \\ &= \frac{k\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2}. \end{aligned} \tag{9.9}$$

The steady-state response to a step input is obtained by setting $s = 0$ which gives

$$u = 1 \qquad \Longrightarrow \qquad y = G_{yu}(0)u = k.$$

If we wish to compute the steady-state response to a sinusoid, we write

$$\begin{aligned} u = \sin\omega t &= \frac{1}{2}\left(ie^{-i\omega t} - ie^{i\omega t}\right), \\ y &= \frac{1}{2}\left(iG_{yu}(-i\omega)e^{-i\omega t} - iG_{yu}(i\omega)e^{i\omega t}\right). \end{aligned}$$

We can now write $G(i\omega)$ in terms of its magnitude and phase,

$$G(i\omega) = \frac{k\omega_0^2}{-\omega^2 + (2\zeta\omega_0\omega)i + \omega_0^2} = Me^{i\theta},$$

where the magnitude (or gain) M and phase θ are given by

$$M = \frac{k\omega_0^2}{\sqrt{(\omega_0^2-\omega^2)^2 + (2\zeta\omega_0\omega)^2}}, \qquad \frac{\sin\theta}{\cos\theta} = \frac{-2\zeta\omega_0\omega}{\omega_0^2-\omega^2}.$$

We can also make use of the fact that $G(-i\omega)$ is given by its complex conjugate $G^*(i\omega)$, and it follows that $G(-i\omega) = Me^{-i\theta}$. Substituting these expressions into our output equation, we obtain

$$\begin{aligned} y &= \frac{1}{2}\left(i(Me^{-i\theta})e^{-i\omega t} - i(Me^{i\theta})e^{i\omega t}\right) \\ &= M \cdot \frac{1}{2}\left(ie^{-i(\omega t+\theta)} - ie^{i(\omega t+\theta)}\right) = M\sin(\omega t+\theta). \end{aligned}$$

The responses to other signals can be computed by writing the input as an appropriate combination of exponential responses and using linearity. ∇

Example 9.2 Electrical circuit elements

Modeling of electrical circuits is a common use of transfer functions. Consider, for example, a resistor modeled by Ohm's law $V = IR$, where V is the voltage across

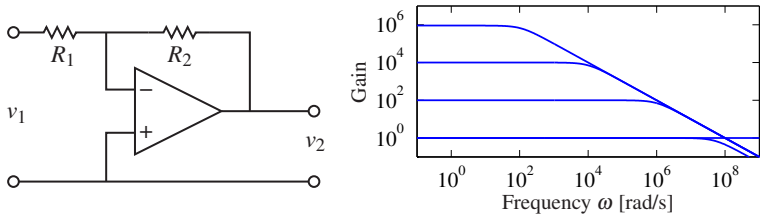


Figure 9.3: Stable amplifier based on negative feedback around an operational amplifier. The block diagram on the left shows a typical amplifier with low-frequency gain R_2/R_1. If we model the dynamic response of the op amp as $G(s) = ak/(s+a)$, then the gain falls off at frequency $\omega = aR_1k/R_2$, as shown in the gain curves on the right. The frequency response is computed for $k = 10^7$, $a = 10$ rad/s, $R_2 = 10^6$ Ω, and $R_1 = 1$, 10^2, 10^4 and 10^6 Ω.

the resistor, I is the current through the resistor and R is the resistance value. If we consider current to be the input and voltage to be the output, the resistor has the transfer function $Z(s) = R$. $Z(s)$ is also called the *impedance* of the circuit element.

Next we consider an inductor whose input/output characteristic is given by

$$L\frac{dI}{dt} = V.$$

Letting the current be $I(t) = e^{st}$, we find that the voltage is $V(t) = Lse^{st}$ and the transfer function of an inductor is thus $Z(s) = Ls$. A capacitor is characterized by

$$C\frac{dV}{dt} = I,$$

and a similar analysis gives a transfer function from current to voltage of $Z(s) = 1/(Cs)$. Using transfer functions, complex electrical circuits can be analyzed algebraically by using the complex impedance $Z(s)$ just as one would use the resistance value in a resistor network. ∇

Example 9.3 Operational amplifier circuit

To further illustrate the use of exponential signals, we consider the operational amplifier circuit described in Section 4.3 and reproduced in Figure 9.3 (left). The model in Section 4.3 is a simplification because the linear behavior of the amplifier is modeled as a constant gain. In reality there are significant dynamics in the amplifier, and the static model $v_{\text{out}} = -kv$ (equation (4.10)) should therefore be replaced by a dynamic model $v_{\text{out}} = -Gv$. A simple transfer function is

$$G(s) = \frac{ak}{s+a}. \tag{9.10}$$

These dynamics correspond to a first-order system with time constant $1/a$. The parameter k is called the *open loop gain*, and the product ak is called the *gain-bandwidth product*; typical values for these parameters are $k = 10^7$ and $ak = 10^7$–10^9 rad/s.

If the input v_1 is an exponential signal e^{st}, then there are solutions where all signals in the circuit are exponentials, $v, v_1, v_2 \in \mathcal{E}$, since all of the elements of the

circuit are modeled as being linear. The equations describing the system can then be manipulated algebraically.

Assuming that the current into the amplifier is zero, as is done in Section 4.3, the current through the resistors R_1 and R_2 are the same, hence

$$\frac{v_1 - v}{R_1} = \frac{v - v_2}{R_2}, \quad \text{or} \quad (R_1 + R_2)v = R_2 v_1 + R_1 v_2$$

Combining the above equation with the open loop dynamics of the operational amplifier (9.10), which can be written as $v_2 = -Gv$ in the simplified notation (9.7), then give the following model for the closed loop system:

$$(R_1 + R_2)v = R_2 v_1 + R_1 v_2, \qquad v_2 = -Gv, \qquad v, v_1, v_2 \in \mathscr{E}. \tag{9.11}$$

Eliminating v between these equations gives

$$v_2 = \frac{-R_2 G(s)}{R_1 + R_2 + R_1 G(s)} v_1 = \frac{-R_2 ak}{R_1 ak + (R_1 + R_2)(s + a)} v_1.$$

and the transfer function of the closed loop system is

$$G_{v_2 v_1} = \frac{-R_2 ak}{R_1 ak + (R_1 + R_2)(s + a)}. \tag{9.12}$$

The low-frequency gain is obtained by setting $s = 0$, hence

$$G_{v_2 v_1}(0) = \frac{-kR_2}{(k+1)R_1 + R_2} \approx -\frac{R_2}{R_1},$$

which is the result given by equation (4.11) in Section 4.3. The bandwidth of the amplifier circuit is

$$\omega_b = a\frac{R_1(k+1) + R_2}{R_1 + R_2} \approx a\frac{R_1 k}{R_2} \quad \text{for} \quad k \gg 1,$$

where the approximation holds for $R_2/R_1 \gg 1$. The gain of the closed loop system drops off at high frequencies as $R_2 k/(\omega(R_1 + R_2))$. The frequency response of the transfer function is shown in Figure 9.3 (right) for $k = 10^7$, $a = 10$ rad/s, $R_2 = 10^6\ \Omega$ and $R_1 = 1$, 10^2, 10^4 and $10^6\ \Omega$.

Note that in solving this example, we bypassed explicitly writing the signals as $v = Ve^{st}$ and instead worked directly with V, assuming it was an exponential. This shortcut is handy in solving problems of this sort and when manipulating block diagrams. A comparison with Section 4.3, where we make the same calculation when $G(s)$ is a constant, shows analysis of systems using transfer functions is as easy as using static systems. The calculations are the same if the resistances R_1 and R_2 are replaced by impedances, as discussed in Example 9.2. ∇

Although we have focused thus far on ordinary differential equations, transfer functions can also be used for other types of linear systems. We illustrate this by time delays and systems described by a partial differential equation.

Example 9.4 Time Delay
Time delays appear in many systems: typical examples are delays in nerve propagation, communication and mass transport. A system with a time delay has the input/output relation

$$y(t) = u(t-\tau). \tag{9.13}$$

To obtain the corresponding transfer function we let the input be $u(t) = e^{st}$, the output is then

$$y(t) = u(t-\tau) = e^{s(t-\tau)} = e^{-s\tau}e^{st} = e^{-s\tau}u(t).$$

and we find that the transfer function of a time delay is thus $G(s) = e^{-s\tau}$, which is not a rational function. ∇

Example 9.5 Heat propagation
Consider the problem of one-dimensional heat propagation in a semi-infinite metal rod. Assume that the input is the temperature at one end and that the output is the temperature at a point along the rod. Let $\theta(x,t)$ be the temperature at position x and time t. With a proper choice of length scales and units, heat propagation is described by the partial differential equation

$$\frac{\partial\theta}{\partial t} = \frac{\partial^2\theta}{\partial^2 x}, \qquad y(t) = \theta(1,t) \tag{9.14}$$

and the point of interest can be assumed to have $x = 1$. The boundary condition for the partial differential equation is

$$\theta(0,t) = u(t).$$

To determine the transfer function we choose the input as $u(t) = e^{st}$. Assume that there is a solution to the partial differential equation of the form $\theta(x,t) = \psi(x)e^{st}$ and insert this into equation (9.14) to obtain

$$s\psi(x) = \frac{d^2\psi}{dx^2},$$

with boundary condition $\psi(0) = 1$. This ordinary differential equation (with independent variable x) has the solution

$$\psi(x) = Ae^{x\sqrt{s}} + Be^{-x\sqrt{s}}.$$

Since the temperature of the rod is bounded we have $A = 0$, the boundary condition gives $B = 1$, and the solution is then

$$y(t) = \theta(1,t) = \psi(1)e^{st} = e^{-\sqrt{s}}e^{st} = e^{-\sqrt{s}}u(t).$$

The system thus has the transfer function $G(s) = e^{-\sqrt{s}}$. As in the case of a time delay, the transfer function is not a rational function but is analytic except at $s = 0$. ∇

Transfer function for some common linear time-invariant systems are given in Table 9.1.

Table 9.1: Transfer functions for some common linear time-invariant systems.

Type	System	Transfer Function
Integrator	$\dot{y} = u$	$\frac{1}{s}$
Differentiator	$y = \dot{u}$	s
First-order system	$\dot{y} + ay = u$	$\frac{1}{s+a}$
Double integrator	$\ddot{y} = u$	$\frac{1}{s^2}$
Damped oscillator	$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y = u$	$\frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$
State space systm	$\dot{x} = Ax + Bu, y = Cx + Du$	$C(sI - A)^{-1}B + D$
PID controller	$y = k_p u + k_d \dot{u} + k_i \int u$	$k_p + k_d s + \frac{k_i}{s}$
Time delay	$y(t) = u(t - \tau)$	$e^{-\tau s}$

Laplace Transforms

Traditionally, Laplace transforms were used to compute responses of linear systems to different stimuli. Today we can easily generate the responses using computers. Laplace transforms were also used to derive transfer functions which we illustrate in this section. Only a few elementary properties are needed for basic control applications. Students who are not familiar with them can safely skip this section. A good reference for the mathematical material in this section is the classic book by Widder [Wid41].

Consider a function $f(t)$, $f : \mathbb{R}^+ \to \mathbb{R}$, that is integrable and grows no faster than $e^{s_0 t}$ for some finite $s_0 \in \mathbb{R}$ and large t. The Laplace transform maps f to a function $F = \mathscr{L} f : \mathbb{C} \to \mathbb{C}$ of a complex variable. It is defined by

$$F(s) = \int_0^\infty e^{-st} f(t)\, dt, \quad \operatorname{Re} s > s_0. \tag{9.15}$$

The transform has some properties that makes it well suited to deal with linear systems. First we observe that the transform is linear because

$$\begin{aligned} \mathscr{L}(af + bg) &= \int_0^\infty e^{-st}(af(t) + bg(t))\, dt \\ &= a\int_0^\infty e^{-st} f(t)\, dt + b\int_0^\infty e^{-st} g(t)\, dt = a\mathscr{L} f + b\mathscr{L} g. \end{aligned} \tag{9.16}$$

Next we will calculate the Laplace transform of the integral of a function. Using

integration by parts, we get

$$\mathscr{L}\int_0^t f(\tau)\,d\tau = \int_0^\infty \left(e^{-st}\int_0^t f(\tau)\,d\tau\right)dt$$
$$= -\frac{e^{-st}}{s}\int_0^t f(\tau)\,d\tau\Big|_0^\infty + \int_0^\infty \frac{e^{-s\tau}}{s} f(\tau)\,d\tau = \frac{1}{s}\int_0^\infty e^{-s\tau} f(\tau)\,d\tau,$$

hence

$$\mathscr{L}\int_0^t f(\tau)\,d\tau = \frac{1}{s}\mathscr{L}f = \frac{1}{s}F(s). \tag{9.17}$$

Integration of a time function thus corresponds to division of the corresponding Laplace transform by s.

Since integration corresponds to division by s, we can expect that differentiation corresponds to multiplication by s. This is not quite true as we will see by calculating the Laplace transform of the derivative of a function. We have

$$\mathscr{L}\frac{df}{dt} = \int_0^\infty e^{-st} f'(t)\,dt = e^{-st} f(t)\Big|_0^\infty + s\int_0^\infty e^{-st} f(t)\,dt = -f(0) + s\mathscr{L}f,$$

where the second equality is obtained using integration by parts. We thus obtain

$$\mathscr{L}\frac{df}{dt} = s\mathscr{L}f - f(0) = sF(s) - f(0). \tag{9.18}$$

Notice that the appearance of the initial value $f(0)$ of the function. The formula (9.18) is particularly simple if the initial conditions are zero, because if $f(0) = 0$ it follows that *differentiation of a function corresponds to multiplication of the transform by s*, compare with the differentiation of exponential signals.

Using Laplace transforms the transfer function for a linear time invariant system can be defined as *the ratio of the transform of the input and the output, when the transforms are computed under the assumption that all initial conditions are zero*. We will now illustrate how Laplace transforms can be used to compute transfer functions.

Example 9.6 Transfer function of state space model
Consider the state space system described by equation (9.2). Taking Laplace transforms give

$$sX(s) - x(0) = AX(s) + BU(s) \qquad Y(s) = CX(s) + DU(s).$$

Elimination of $X(s)$ gives

$$X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s). \tag{9.19}$$

When the initial condition $x(0)$ is zero we have

$$X(s) = (sI - A)^{-1}BU(s), \qquad Y(s) = \Big(C(sI - A)^{-1}B + D\Big)U(s).$$

and the transfer function is $G(s) = C(sI - A)^{-1}B + D$, compare with equation (9.4).
∇

Example 9.7 Transfer function and impulse response
Consider a linear time-invariant system with zero initial state. We saw in Section 6.3 that the relation between the input u and the output y is given by the convolution integral

$$y(t) = \int_0^\infty h(t-\tau)u(\tau)\,d\tau,$$

where $h(t)$ is the impulse response for the system (assumed causal). Taking the Laplace transform of this expression gives

$$\begin{aligned} Y(s) &= \int_0^\infty e^{-st}y(t)\,dt = \int_0^\infty e^{-st}\int_0^\infty h(t-\tau)u(\tau)\,d\tau\,dt \\ &= \int_0^\infty \int_0^t e^{-s(t-\tau)}e^{-s\tau}h(t-\tau)u(\tau)\,d\tau\,dt \\ &= \int_0^\infty e^{-s\tau}u(\tau)\,d\tau \int_0^\infty e^{-st}h(t)\,dt = H(s)U(s). \end{aligned}$$

Thus, the input/output response is given by $Y(s) = H(s)U(s)$, where H, U and Y are the Laplace transforms of h, u and y.

The system theoretic interpretation is that the Laplace transform of the output of a linear system is a product of two terms, the Laplace transform of the input $U(s)$ and the Laplace transform of the impulse response of the system $H(s)$. A mathematical interpretation is that the Laplace transform of a convolution is the product of the transforms of the functions that are convolved. The fact that the formula $Y(s) = H(s)U(s)$ is much simpler than a convolution is one reason why Laplace transforms have become popular in engineering. ∇

Coordinate Changes

The matrices A, B and C in equation (9.2) depend on the choice of coordinate system for the states. Since the transfer function relates input to outputs, it should be invariant to coordinate changes in the state space. To show this, consider the model (9.2) and introduce new coordinates z by the transformation $z = Tx$, where T is a nonsingular matrix. The system is then described by

$$\begin{aligned} \frac{dz}{dt} &= T(Ax+Bu) = TAT^{-1}z + TBu =: \tilde{A}z + \tilde{B}u, \\ y &= Cx + Du = CT^{-1}z + Du =: \tilde{C}z + Du. \end{aligned}$$

This system has the same form as equation (9.2), but the matrices A, B and C are different:

$$\tilde{A} = TAT^{-1}, \qquad \tilde{B} = TB, \qquad \tilde{C} = CT^{-1}. \tag{9.20}$$

Computing the transfer function of the transformed model, we get

$$\begin{aligned} \tilde{G}(s) &= \tilde{C}(sI-\tilde{A})^{-1}\tilde{B} + \tilde{D} = CT^{-1}(sI - TAT^{-1})^{-1}TB + D \\ &= C\big(T^{-1}(sI - TAT^{-1})T\big)^{-1}B + D = C(sI-A)^{-1}B + D = G(s), \end{aligned}$$

which is identical to the transfer function (9.4) computed from the system description (9.2). The transfer function is thus invariant to changes of the coordinates in the state space.

9.3 Gain Poles and Zeros

The transfer function has many useful interpretations and the features of a transfer function are often associated with important system properties. Three of the most important features are the gain and the locations of the poles and zeros.

Gain

The *zero frequency gain* of a system is given by the magnitude of the transfer function at $s = 0$. It represents the ratio of the steady-state value of the output with respect to a step input (which can be represented as $u = e^{st}$ with $s = 0$). For a state space system, we computed the zero frequency gain in equation (6.22):

$$G(0) = D - CA^{-1}B.$$

Consider a stable system modeled by the linear differential equation

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots + a_n y = b_0 \frac{d^m u}{dt^m} + b_1 \frac{d^{m-1} u}{dt^{m-1}} + \cdots + b_m u.$$

If we assume that the input and output of the system are constants y_0 and u_0, then we find that $a_n y_0 = b_m u_0$, and the zero frequency gain is

$$G(0) = \frac{y_0}{u_0} = \frac{b_m}{a_n}. \tag{9.21}$$

Recall that in Section 6.3 equation (6.24) we introduced the notion of gain and phase for the steady-state response $G(i\omega)$ to sinusoidal signals.

Poles and Zeros

Next consider a linear system with the rational transfer function

$$G(s) = \frac{b(s)}{a(s)}.$$

The roots of the polynomial $a(s)$ are called the *poles* of the system, and the roots of $b(s)$ are called the *zeros* of the system. A pole p corresponds to a *mode* of the system with corresponding modal solution e^{pt}. The unforced motion of the system after an arbitrary excitation is a weighted sum of modes.

Zeros have a different interpretation. Since the pure exponential output corresponding to the input $u(t) = e^{st}$ with $a(s) \neq 0$ is $G(s)e^{st}$, it follows that the pure exponential output is zero if $b(s) = 0$. Zeros of the transfer function thus block transmission of the corresponding exponential signals.

For a state space system with transfer function $G(s) = C(sI - A)^{-1}B + D$, the poles of the transfer function are the eigenvalues of the matrix A in the state space model. One easy way to see this is to notice that the value of $G(s)$ is unbounded when s is an eigenvalue of a system since this is precisely the set of points where the characteristic polynomial $\lambda(s) = \det(sI - A) = 0$ (and hence $sI - A$ is non-invertible). It follows that the poles of a state space system depend only on the matrix A, which represents the intrinsic dynamics of the system. We say that a transfer function is stable if all of its poles have negative real part.

To find the zeros of a state space system, we observe that the zeros are complex numbers s such that the input $u(t) = U_0 e^{st}$ gives zero output. Inserting the pure exponential response $x(t) = X_0 e^{st}$ and setting $y(t) = 0$ in equation (9.2) gives

$$se^{st}x_0 = AX_0e^{st} + BU_0e^{st} \qquad 0 = Ce^{st}X_0 + De^{st}U_0,$$

which can be written as

$$\begin{pmatrix} A - sI & B \\ C & D \end{pmatrix} \begin{pmatrix} X_0 \\ U_0 \end{pmatrix} e^{st} = 0.$$

This equation has a solution with nonzero X_0, U_0 only if the matrix on the left does not have full column rank. The zeros are thus the values s such that the matrix

$$\begin{pmatrix} A - sI & B \\ C & D \end{pmatrix} \tag{9.22}$$

loses rank.

Since the zeros depend on A, B, C and D, they therefore depend on how the inputs and outputs are coupled to the states. Notice in particular that if the matrix B has full row rank, then the matrix in equation (9.22) has n linearly independent rows for all values of s. Similarly there are n linearly independent columns if the matrix C has full column rank. This implies that systems where the matrix B or C is square and full rank do not have zeros. In particular it means that a system has no zeros if it is fully actuated (each state can be controlled independently) or if the full state is measured.

A convenient way to view the poles and zeros of a transfer function is through a *pole zero diagram*, as shown in Figure 9.4. In this diagram, each pole is marked with a cross, and each zero with a circle. If there are multiple poles or zeros at a fixed location, these are often indicated with overlapping crosses or circles (or other annotations). Poles in the left half-plane correspond to stable modes of the system, and poles in the right half-plane correspond to unstable modes. We thus call a pole in the left-half plane a *stable pole* and a pole in the right-half plane an *unstable pole*. A similar terminology is used for zeros, even though the zeros do not directly relate to stability or instability of the system. Notice that the gain must also be given to have a complete description of the transfer function.

Example 9.8 Balance system

Consider the dynamics for a balance system, shown in Figure 9.5. The transfer function for a balance system can be derived directly from the second-order equa-

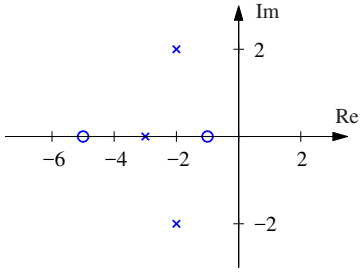


Figure 9.4: A pole zero diagram for a transfer function with zeros at -5 and -1 and poles at -3 and $-2\pm 2j$. The circles represent the locations of the zeros, and the crosses the locations of the poles. A complete characterization requires we also specify the gain of the system.

tions, given in Example 3.1:

$$M_t\frac{d^2p}{dt^2} - ml\frac{d^2\theta}{dt^2}\cos\theta + c\frac{dp}{dt} + ml\sin\theta\left(\frac{d\theta}{dt}\right)^2 = F,$$
$$-ml\cos\theta\frac{d^2p}{dt^2} + J_t\frac{d^2\theta}{dt^2} + \gamma\frac{d\theta}{dt} - mgl\sin\theta = 0.$$

If we assume that θ and $\dot{\theta}$ are small, we can approximate this nonlinear system by a set of linear second-order differential equations,

$$M_t\frac{d^2p}{dt^2} - ml\frac{d^2\theta}{dt^2} + c\frac{dp}{dt} = F,$$
$$-ml\frac{d^2p}{dt^2} + J_t\frac{d^2\theta}{dt^2} + \gamma\frac{d\theta}{dt} - mgl\theta = 0.$$

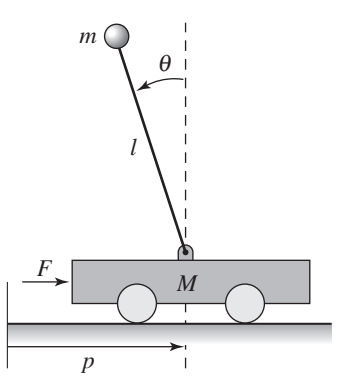
If we let F be an exponential signal, the resulting response satisfies

$$M_ts^2\,p - mls^2\,\theta + cs\,p = F,$$
$$J_ts^2\,\theta - mls^2\,p + \gamma s\,\theta - mgl\,\theta = 0,$$

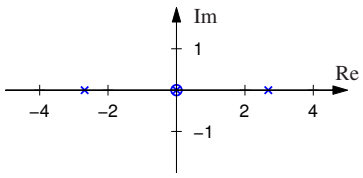
where all signals are exponential signals. The resulting transfer functions for the position of the cart and the orientation of the pendulum are given by solving for p and θ in terms of F to obtain

$$H_{\theta F} = \frac{mls}{(M_tJ_t - m^2l^2)s^3 + (\gamma M_t + cJ_t)s^2 + (c\gamma - M_tmgl)s - mglc},$$
$$H_{pF} = \frac{J_ts^2 + \gamma s - mgl}{(M_tJ_t - m^2l^2)s^4 + (\gamma M_t + cJ_t)s^3 + (c\gamma - M_tmgl)s^2 - mglcs},$$

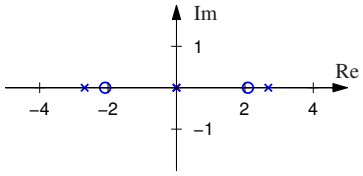
where each of the coefficients is positive. The pole zero diagrams for these two transfer functions are shown in Figure 9.5 using the parameters from Example 7.7.



(a) Cart–pendulum system



(b) Pole zero diagram for $H_{\theta F}$



(c) Pole zero diagram for H_{pF}

Figure 9.5: Poles and zeros for a balance system. The balance system (a) can be modeled around its vertical equilibrium point by a fourth order linear system. The poles and zeros for the transfer functions $H_{\theta F}$ and H_{pF} are shown in (b) and (c), respectively.

If we assume the damping is small and set $c = 0$ and $\gamma = 0$, we obtain

$$H_{\theta F} = \frac{ml}{(M_t J_t - m^2 l^2)s^2 - M_t mgl},$$
$$H_{pF} = \frac{J_t s^2 - mgl}{s^2\big((M_t J_t - m^2 l^2)s^2 - M_t mgl\big)}.$$

This gives nonzero poles and zeros at

$$p = \pm\sqrt{\frac{mglM_t}{M_t J_t - m^2 l^2}} \approx \pm 2.68, \qquad z = \pm\sqrt{\frac{mgl}{J_t}} \approx \pm 2.09.$$

We see that these are quite close to the pole and zero locations in Figure 9.5. ∇

Pole/Zero Cancellations

Because transfer functions are often polynomials in s, it can sometimes happen that the numerator and denominator have a common factor, which can be canceled. Sometimes these cancellations are simply algebraic simplifications, but in other situations they can mask potential fragilities in the model. In particular, if a pole/zero cancellation occurs because terms in separate blocks that just happen to coincide, the cancellation may not occur if one of the systems is slightly perturbed. In some situations this can result in severe differences between the expected behavior and the actual behavior.

Consider the block diagram in Figure 9.8 with $F = 1$ (no feedforward compen-

sation) and let C and P be given by

$$C(s) = \frac{n_c(s)}{d_c(s)}, \qquad P(s) = \frac{n_p(s)}{d_p(s)}.$$

The transfer function from r to e is then given by

$$G_{er}(s) = \frac{1}{1+PC} = \frac{d_c(s)d_p(s)}{d_c(s)d_p(s)+n_c(s)n_p(s)}.$$

If there are common factors in the numerator and denominator polynomials, then these terms can be factored out and eliminated from both the numerator and denominator. For example, if the controller has a zero at $s = -a$ and the process has a pole at $s = -a$, then we will have

$$G_{er}(s) = \frac{(s+a)d_c(s)d'_p(s)}{(s+a)d_c(s)d'_p(s)+(s+a)n'_c(s)n_p(s)} = \frac{d_c(s)d'_p(s)}{d_c(s)d'_p(s)+n'_c(s)n_p(s)},$$

where $n'_c(s)$ and $d'_p(s)$ represent the relevant polynomials with the term $s+a$ factored out and it does not appear in the transfer function G_{er}.

Suppose instead that we compute the transfer function from v to e, which represents the effect of a disturbance on the error between the reference and the output. This transfer function is given by

$$G_{ev}(s) = -\frac{d_c(s)n_p(s)}{(s+a)d_c(s)d'_p(s)+(s+a)n'_c(s)n_p(s)}.$$

Notice that if $a < 0$, then the pole is in the right half-plane and the transfer function G_{ev} is *unstable*. Hence, even though the transfer function from r to e appears to be okay (assuming a perfect pole/zero cancellation), the transfer function from v to e can exhibit unbounded behavior. This unwanted behavior is typical of an *unstable pole/zero cancellation*.

It turns out that the cancellation of a pole with a zero can also be understood in terms of the state space representation of the systems. Reachability or observability is lost when there are cancellations of poles and zeros (Exercise 9.11). A consequence is that the transfer function represents the dynamics only in the reachable and observable subsystem (see Section 8.3).

Another property of the transfer function is that it corresponds to the portion of the state space dynamics that is both reachable and observable. In particular, if we make use of the Kalman decomposition (Section 8.3), then the transfer function depends only on the dynamics in the reachable and observable subspace Σ_{ro} (Exercise 9.7).

Example 9.9 Cruise control

The input/output response from throttle to velocity for the linearized model for a car has the transfer function $G(s) = b/(s+a)$. A simple (but not necessarily good) way to design a PI controller is to choose the parameters of the PI controller as $k_i = ak_p$. The controller zero at $s = -k_i/k_p = -a$ then cancels the process pole at

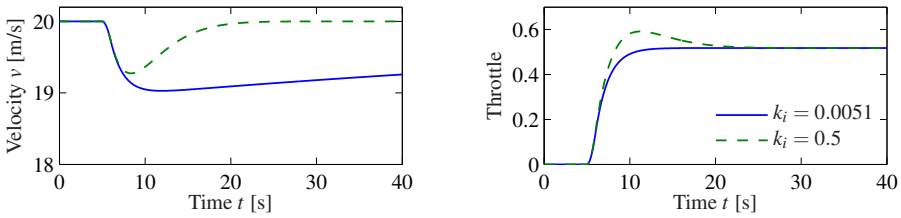


Figure 9.6: Car with PI cruise control encountering a sloping road. The velocity error is shown on the left and the throttle is shown on the right. Results with a PI controller with $k_p = 0.5$ and $k_i = 0.0051$, where the process pole $s = -0.0101$, is shown by solid lines, and a controller with $k_p = 0.5$ and $k_i = 0.5$ is shown by dashed lines. Compare with Figure 4.3b.

$s = -a$. The transfer function from reference to velocity is $G_{vr}(s) = bk_p/(s+bk_p)$, and control design is then simply a matter of choosing the gain k_p. The closed loop system dynamics are of first order with the time constant $1/bk_p$.

Figure 9.6 shows the velocity error when the car encounters an increase in the road slope. A comparison with the controller used in Figure 4.3b (reproduced in dashed curves) shows that the controller based on pole/zero cancellation has very poor performance. The velocity error is larger, and it takes a long time to settle.

Notice that the control signal remains practically constant after $t = 15$ even if the error is large after that time. To understand what happens we will analyze the system. The parameters of the system are $a = 0.0101$ and $b = 1.32$, and the controller parameters are $k_p = 0.5$ and $k_i = 0.0051$. The closed loop time constant is $1/(bk_p) = 2.5$ s, and we would expect that the error would settle in about 10 s (4 time constants). The transfer functions from road slope to velocity and control signals are

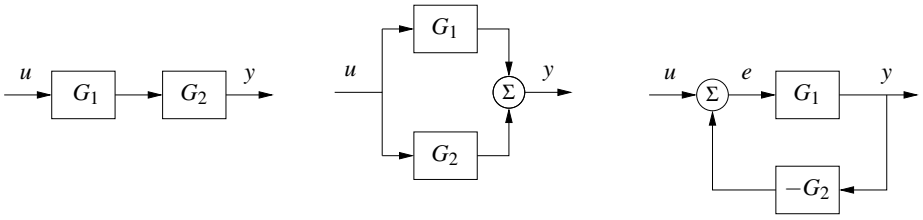
$$G_{v\theta}(s) = \frac{b_g s}{(s+a)(s+bk_p)}, \qquad G_{u\theta}(s) = \frac{bk_p}{s+bk_p}.$$

Notice that the canceled mode $s = a = -0.0101$ appears in $G_{v\theta}$ but not in $G_{u\theta}$. The reason why the control signal remains constant is that the controller has a zero at $s = -0.0101$, which cancels the slowly decaying process mode. Notice that the error would diverge if the canceled pole was unstable. ∇

The lesson we can learn from this example is that it is a bad idea to try to cancel unstable or slow process poles. A more detailed discussion of pole/zero cancellations is given in Section 13.4.

9.4 Block Diagrams and Transfer Functions

The combination of block diagrams and transfer functions is a powerful way to represent control systems. Transfer functions relating different signals in the system can be derived by purely algebraic manipulations of the transfer functions of the blocks using *block diagram algebra*. To show how this can be done, we will



(a) $G_{yu} = G_2 G_1$ (b) $G_{yu} = G_1 + G_2$ (c) $G_{yu} = \dfrac{G_1}{1 + G_1 G_2}$

Figure 9.7: Interconnections of linear systems. Series (a), parallel (b) and feedback (c) connections are shown. The transfer functions for the composite systems can be derived by algebraic manipulations assuming exponential functions for all signals.

begin with simple combinations of systems. We will assume that all signals are exponential signals $\mathscr{E}$ and we will use the compact notation $y = Gu$ for the output $y \in \mathscr{E}$ of a linear time-invariant system with the input $u \in \mathscr{E}$ and the transfer function G, see equation (9.7) and recall its interpretation.

Consider a system that is a cascade combination of systems with the transfer functions $G_1(s)$ and $G_2(s)$, as shown in Figure 9.7a. Let the input of the system be $u \in \mathscr{E}$. The output of the first block is then $G_1 u \in \mathscr{E}$, which is also the input to the second system. The output of the second system is then

$$y = G_2(G_1 u) = (G_2 G_1)u, \qquad u, y \in \mathscr{E}. \tag{9.23}$$

Let $u = e^{st}$ be the input to the system, y be the pure exponential output, and e be the pure exponential part of the intermediate signal given by the sum of u and the output of the second block. The transfer function of the series connection is thus $G = G_2 G_1$, i.e., the product of the transfer functions. The order of the individual transfer functions is due to the fact that we place the input signal on the right-hand side of this expression, hence we first multiply by G_1 and then by G_2. Unfortunately, this has the opposite ordering from the diagrams that we use, where we typically have the signal flow from left to right, so one needs to be careful. The ordering is important if either G_1 or G_2 is a vector-valued transfer function, as we shall see in some examples.

Consider next a parallel connection of systems with the transfer functions G_1 and G_2, as shown in Figure 9.7b and assume that all signals are exponential signals. The outputs of the first and second systems are simply $G_1 u$ and $G_2 u$ and the output of the parallel conncection is

$$y = G_1 u + G_2 u = (G_1 + G_2)u, \qquad u, y \in \mathscr{E}.$$

The transfer function for a parallel connection is thus $G = G_1 + G_2$.

Finally, consider a feedback connection of systems with the transfer functions G_1 and G_2, as shown in Figure 9.7c. Writing the relations between the signals for the different blocks and the summation unit, we find

$$y = G_1 e, \qquad e = u - G_2 y, \qquad y, e, u \in \mathscr{E}. \tag{9.24}$$

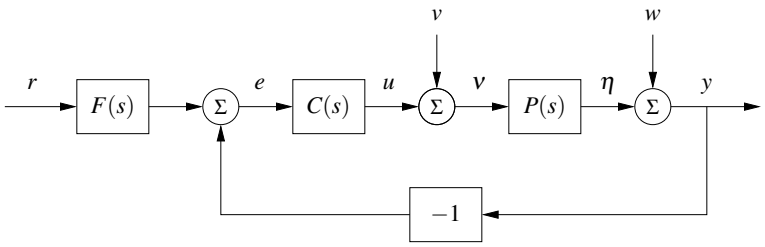


Figure 9.8: Block diagram of a feedback system. The inputs to the system are the reference signal r, the process disturbance v and the measurement noise w. The remaining signals in the system can all be chosen as possible outputs, and transfer functions can be used to relate the system inputs to the other labeled signals.

Elimination of e gives

$$y = G_1(u - G_2 y) \quad \Longrightarrow \quad (1 + G_1 G_2) y = G_1 u \quad \Longrightarrow \quad y = \frac{G_1}{1 + G_1 G_2} u.$$

The transfer function of the feedback connection is thus

$$G = \frac{G_1}{1 + G_1 G_2}. \tag{9.25}$$

These three basic interconnections can be used as the basis for computing transfer functions for more complicated systems.

Control System Transfer Functions

Consider the system in Figure 9.8, which was given at the beginning of the chapter. The system has three blocks representing a process P, a feedback controller C and a feedforward controller F. Together, C and F define the *control law* for the system. There are three external signals: the reference (or command signal) r, the load disturbance v and the measurement noise w. A typical problem is to find out how the error e is related to the signals r, v and w.

To derive the transfer functions we are interested in we assume that all signals are exponential signals $\mathscr{E}$ and we write the relations between the signals for each block in the block transfer function. Assume for example that we are interested in the control error e. The summation point and the block $F(s)$ gives

$$e = Fr - y, \qquad e, r, y \in \mathscr{E}.$$

The signal y is the sum of w and η, where η is the output of the process $P(s)$: .

$$y = n + \eta, \qquad \eta = P(v + u), \qquad u = Ce, \qquad y, n, \eta, v, u, e \in \mathscr{E}.$$

Combining these equations gives

$$\begin{aligned} e &= Fr - y = Fr - (n + \eta) = Fr - \big(n + P(d + u)\big) \\ &= Fr - \big(n + P(d + Ce)\big), \end{aligned}$$

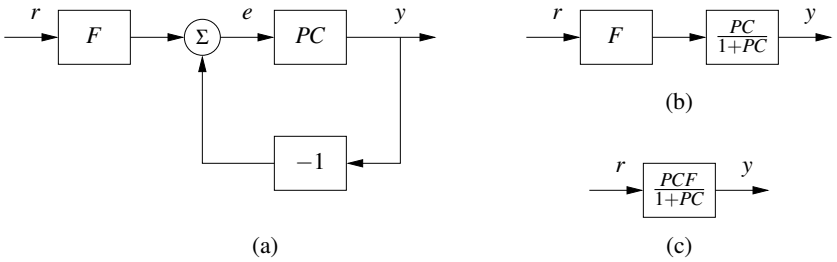


Figure 9.9: Example of block diagram algebra. The results from multiplying the process and controller transfer functions (from Figure 9.8) are shown in (a). Replacing the feedback loop with its transfer function equivalent yields (b), and finally multiplying the two remaining blocks gives the reference to output representation in (c).

and hence

$$e = Fr - w - Pv - PCe, \qquad e, r, w, v \in \mathcal{E}.$$

Finally, solving this equation for e gives

$$e = \frac{F}{1+PC} r - \frac{1}{1+PC} w - \frac{P}{1+PC} v = G_{er} r + G_{ew} w + G_{ev} v, \tag{9.26}$$

and the error is thus the sum of three terms, depending on the reference r, the measurement noise w and the load disturbance v. The functions

$$G_{er} = \frac{F}{1+PC}, \qquad G_{ew} = \frac{-1}{1+PC}, \qquad G_{ev} = \frac{-P}{1+PC} \tag{9.27}$$

are transfer functions from reference r, noise w and disturbance v to the error e.

We can also derive transfer functions by manipulating the block diagrams directly, as illustrated in Figure 9.9. Suppose we wish to compute the transfer function between the reference r and the output y. We begin by combining the process and controller blocks in Figure 9.8 to obtain the diagram in Figure 9.9a. We can now eliminate the feedback loop using the algebra for a feedback interconnection (Figure 9.9b) and then use the series interconnection rule to obtain

$$G_{yr} = \frac{PCF}{1+PC}. \tag{9.28}$$

Similar manipulations can be used to obtain the other transfer functions (Exercise 9.8).

The above analysis illustrates an effective way to manipulate the equations to obtain the relations between inputs and outputs in a feedback system. The general idea is to start with the variable of interest and to trace variables backwards around the feedback loop. With some practice, equations (9.26) and (9.27) can be written directly by inspection of the block diagram. Notice, for example, that all terms in equation (9.27) have the same denominators and that the numerators are the blocks that one passes through when going directly from input to output (ignoring the feedback). This type of rule can be used to compute transfer functions by inspection, although for systems with multiple feedback loops it can be tricky

to compute them without writing down the algebra explicitly.

Next we will investigate a system with a controller based on state feedback and an observer shown in Figure 8.7. The process model is

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx,$$

and the controller (8.14) is given by

$$u = -K\hat{x} + k_r r, \tag{9.29}$$

where $\hat{x}$ is the output of a state observer (8.15) given by

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}), \qquad u = -K\hat{x} + k_r r. \tag{9.30}$$

The controller is a system with one input u and two outputs the reference r and the measured signal y. Using transfer functions and exponential signals it can be represented as

$$u = G_{ur} r + G_{uy} y, \qquad u, r, y \in \mathcal{E}. \tag{9.31}$$

The transfer function, G_{uy}, from y to u describes the feedback action and, $G_{ur}(s)$, from r to u describes the feedforward action. We call these *open loop* transfer functions because they represent the relationships between the signals without considering the dynamics of the process (e.g., removing $P(s)$ from the system description or cutting the loop at the process input or output). To derive the controller transfer functions we rewrite equation (9.30) as

$$\frac{d\hat{x}}{dt} = (A - BK - LC)\hat{x} + Bk_r r + Ly, \qquad u = -K\hat{x} + k_r r.$$

Let x, r, y be exponential signals, the above equation gives

$$(sI - (A - BK - LC))\hat{x} = Bk_r + Ly, \qquad u = -K\hat{x} + k_r r, \qquad x, r, y \in \mathcal{E},$$

and we find that the controller transfer functions in equation (9.31) are

$$G_{ur} = k_r - K(sI - BK - KC)^{-1} Bk_r, \qquad G_{uy} = -K(sI - BK - LC)^{-1} L \tag{9.32}$$

We illustrated with an example.

Example 9.10 Vehicle steering

Consider the linearized model for vehicle steering introduced in Example 6.12. In Examples 7.4 and 8.3 we designed a state feedback controller and state estimator for the system. A block diagram for the resulting control system is given in Figure 9.10. Note that we have split the estimator into two components, $G_{\hat{x}u}(s)$ and $G_{\hat{x}y}(s)$, corresponding to its inputs u and y. Using the expressions for A, B, C and L from Example 8.3, it follows from equation (9.32)

$$G_{uy}(s) = -\frac{s(k_1 l_1 + k_2 l_2) + k_1 l_2}{s^2 + s(\gamma k_1 + k_2 + l_1) + k_1 + l_2 + k_2 l_1 - \gamma k_2 l_2} = \frac{-KG_{\hat{x}y}(s)}{1 + KG_{\hat{x}u}(s)},$$

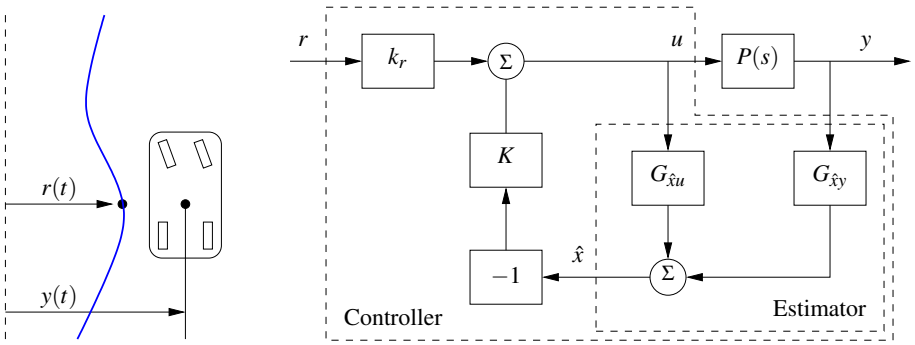


Figure 9.10: Block diagram for a steering control system. The control system is designed to maintain the lateral position of the vehicle along a reference curve (left). The structure of the control system is shown on the right as a block diagram of transfer functions. The estimator consists of two components that compute the estimated state $\hat{x}$ from the combination of the input u and output y of the process. The estimated state is fed through a state feedback controller and combined with a reference gain to obtain the commanded steering angle u.

and

$$G_{ur}(s) = \frac{k_r(s^2 + l_1 s + l_2)}{s^2 + s(\gamma k_1 + k_2 + l_1) + k_1 + l_2 + k_2 l_1 - \gamma k_2 l_2} = \frac{k_r}{1 + K G_{\hat{x}u}(s)},$$

where k_1 and k_2 are the state feedback gains and k_r is the reference gain. The last equalities are obtained applying block diagram algebra to Figure 9.10.

To compute the full closed loop dynamics, we begin by deriving the transfer function for the process $P(s)$. We can compute this directly from the state space description, which was given in Example 6.12. Using that description, we have

$$P(s) = G_{yu}(s) = C(sI - A)^{-1}B + D = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} s & -1 \\ 0 & s \end{pmatrix}^{-1} \begin{pmatrix} \gamma \\ 1 \end{pmatrix} = \frac{\gamma s + 1}{s^2}.$$

The transfer function for the full closed loop system between the input r and the output y is then given by

$$G_{yr} = \frac{P(s)G_{ur}(s)}{1 - P(s)G_{uy}(s)} = \frac{k_r(\gamma s + 1)}{s^2 + (k_1\gamma + k_2)s + k_1}.$$

Note that the observer gains l_1 and l_2 do not appear in this equation. This is because we are considering steady-state analysis and, in steady state, the estimated state exactly tracks the state of the system assuming perfect models. We will return to this example in Chapter 13 to study the robustness of this particular approach.

∇

9.5 The Bode Plot

The frequency response of a linear system can be computed from its transfer function by setting $s = i\omega$, corresponding to a complex exponential

$$u(t) = e^{i\omega t} = \cos(\omega t) + i\sin(\omega t).$$

The resulting output has the form

$$y(t) = G(i\omega)e^{i\omega t} = Me^{i(\omega t+\varphi)} = M\cos(\omega t+\varphi) + iM\sin(\omega t+\varphi),$$

where M and φ are the gain and phase of G:

$$M = |G(i\omega)|, \qquad \varphi = \arctan\frac{\operatorname{Im} G(i\omega)}{\operatorname{Re} G(i\omega)}.$$

The phase of G is also called the *argument* of G, a term that comes from the theory of complex variables.

It follows from linearity that the response to a single sinusoid (sin or cos) is amplified by M and phase-shifted by φ. It will often be convenient to represent the phase in degrees rather than radians. We will use the notation $\angle G(i\omega)$ for the phase in degrees and $\arg G(i\omega)$ for the phase in radians. In addition, while we always take $\arg G(i\omega)$ to be in the range $(-\pi, \pi]$, we will take $\angle G(i\omega)$ to be continuous, so that it can take on values outside the range of -180° to 180°.

The frequency response $G(i\omega)$ can thus be represented by two curves: the gain curve and the phase curve. The *gain curve* gives $|G(i\omega)|$ as a function of frequency ω, and the *phase curve* gives $\angle G(i\omega)$. One particularly useful way of drawing these curves is to use a log/log scale for the gain plot and a log/linear scale for the phase plot. This type of plot is called a *Bode plot* and is shown in Figure 9.11.

Sketching and Interpreting Bode Plots

Part of the popularity of Bode plots is that they are easy to sketch and interpret. Since the frequency scale is logarithmic, they cover the behavior of a linear system over a wide frequency range.

Consider a transfer function that is a rational function of the form

$$G(s) = \frac{b_1(s)b_2(s)}{a_1(s)a_2(s)}.$$

We have

$$\log|G(s)| = \log|b_1(s)| + \log|b_2(s)| - \log|a_1(s)| - \log|a_2(s)|,$$

and hence we can compute the gain curve by simply adding and subtracting gains corresponding to terms in the numerator and denominator. Similarly,

$$\angle G(s) = \angle b_1(s) + \angle b_2(s) - \angle a_1(s) - \angle a_2(s),$$

and so the phase curve can be determined in an analogous fashion. Since a poly-

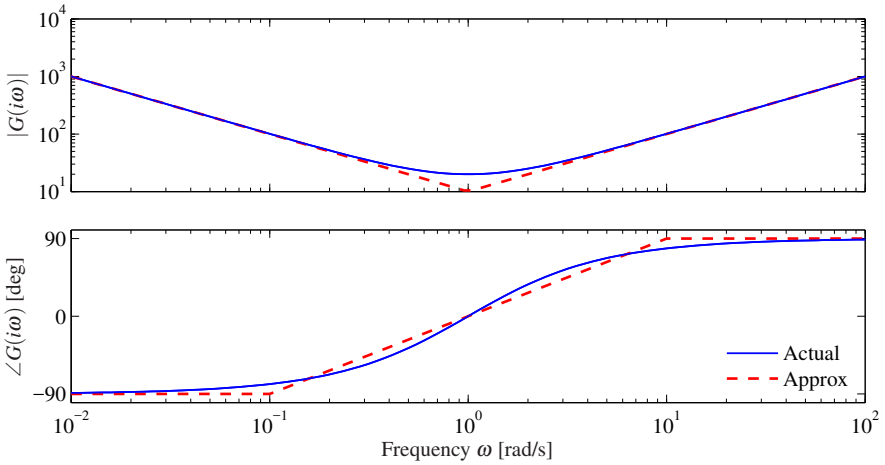


Figure 9.11: Bode plot of the transfer function $C(s) = 20 + \frac{10}{s} + 10s = 10\frac{(s+1)^2}{s}$ corresponding to an ideal PID controller. The top plot is the gain curve and the bottom plot is the phase curve. The dashed lines show straight-line approximations of the gain curve and the corresponding phase curve.

nomial can be written as a product of terms of the type

$$k, \quad s, \quad s+a, \quad s^2 + 2\zeta\omega_0 s + \omega_0^2,$$

it suffices to be able to sketch Bode diagrams for these terms. The Bode plot of a complex system is then obtained by adding the gains and phases of the terms.

The function $G(s) = s^k$ is a simple transfer function, with the important special cases of $k = 1$ corresponding to a differentiator and $k = -1$ to an integrator. The gain and phase of the term are given by

$$\log|G(i\omega)| = k\log\omega, \quad \angle G(i\omega) = 90k.$$

The gain curve is thus a straight line with slope k, and the phase curve is a constant at $90^\circ \times k$. The case when $k = 1$ corresponds to a differentiator and has slope 1 with phase 90°. The case when $k = -1$ corresponds to an integrator and has slope -1 with phase -90°. Bode plots of the various powers of k are shown in Figure 9.12.

Consider next the transfer function of a first-order system, given by

$$G(s) = \frac{a}{s+a}, \qquad a > 0.$$

We have

$$|G(s)| = \frac{|a|}{|s+a|}, \qquad \angle G(s) = \angle(a) - \angle(s+a),$$

and hence

$$\log|G(i\omega)| = \log a - \frac{1}{2}\log(\omega^2 + a^2), \qquad \angle G(i\omega) = -\frac{180}{\pi}\arctan\frac{\omega}{a}.$$

The Bode plot is shown in Figure 9.13a, with the magnitude normalized by the

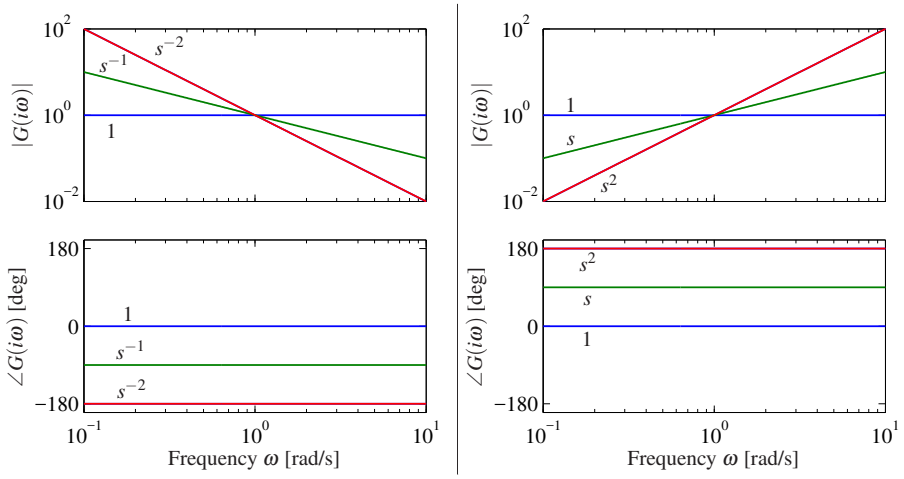


Figure 9.12: Bode plots of the transfer functions $G(s) = s^k$ for $k = -2, -1, 0, 1, 2$. On a log-log scale, the gain curve is a straight line with slope k. The phase curves for the transfer functions are constants, with phase equal to $90^\circ \times k$

.

zero frequency gain. Both the gain curve and the phase curve can be approximated by the following straight lines

$$\log|G(i\omega)| \approx \begin{cases} 0 & \text{if } \omega < a \\ \log a - \log \omega & \text{if } \omega > a, \end{cases}$$

$$\angle G(i\omega) \approx \begin{cases} 0 & \text{if } \omega < a/10 \\ -45 - 45(\log \omega - \log a) & a/10 < \omega < 10a \\ -90 & \text{if } \omega > 10a, \end{cases}$$

which intersect at $\omega = a$. The approximate gain curve consists of a horizontal line up to frequency $\omega = a$, called the *breakpoint* or *corner frequency*, after which the curve is a line of slope -1 (on a log-log scale). The phase curve is zero up to frequency $a/10$ and then decreases linearly by 45°/decade up to frequency $10a$, at which point it remains constant at 90°. Notice that a first-order system behaves like a constant for low frequencies and like an integrator for high frequencies; compare with the Bode plot in Figure 9.12.

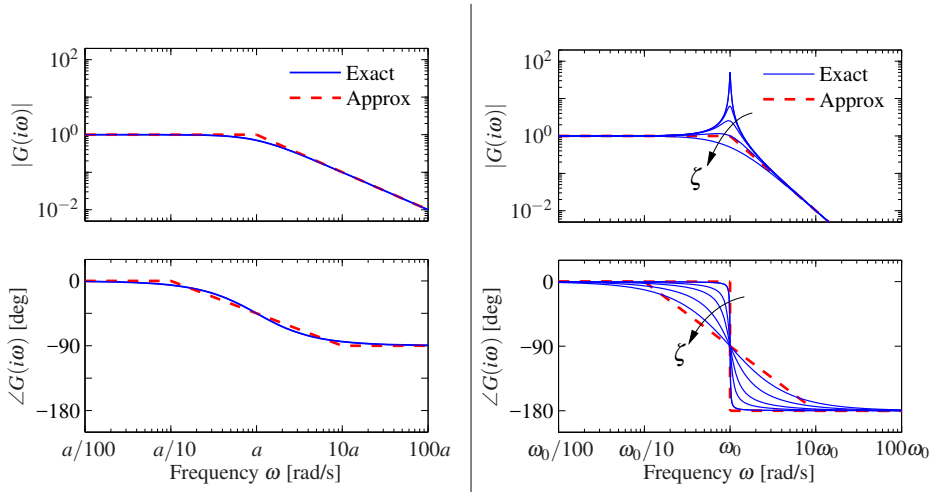
Finally, consider the transfer function for a second-order system,

$$G(s) = \frac{\omega_0^2}{s^2 + 2\omega_0\zeta s + \omega_0^2},$$

for which we have

$$\log|G(i\omega)| = 2\log\omega_0 - \frac{1}{2}\log\left(\omega^4 + 2\omega_0^2\omega^2(2\zeta^2 - 1) + \omega_0^4\right),$$

$$\angle G(i\omega) = -\frac{180}{\pi}\arctan\frac{2\zeta\omega_0\omega}{\omega_0^2 - \omega^2},$$



(a) First-order system (b) Second-order system

Figure 9.13: Bode plots for first- and second-order systems. (a) The first-order system $G(s) = a/(s+a)$ can be approximated by asymptotic curves (dashed) in both the gain and the frequency, with the breakpoint in the gain curve at $\omega = a$ and the phase decreasing by 90° over a factor of 100 in frequency. (b) The second-order system $G(s) = \omega_0^2/(s^2 + 2\zeta\omega_0 s + \omega_0^2)$ has a peak at frequency a and then a slope of -2 beyond the peak; the phase decreases from 0° to −180°. The height of the peak and the rate of change of phase depending on the damping ratio ζ ($\zeta = 0.02$, 0.1, 0.2, 0.5 and 1.0 shown).

which intersect at $\omega = \omega_0$. The gain curve has an asymptote with zero slope for $\omega \ll \omega_0$. For large values of ω the gain curve has an asymptote with slope -2. The largest gain $Q = \max_\omega |G(i\omega)| \approx 1/(2\zeta)$, called the *Q-value*, is obtained for $\omega \approx \omega_0$. The phase is zero for low frequencies and approaches 180° for large frequencies. The curves can be approximated with the following piecewise linear expressions

$$\log|G(i\omega)| \approx \begin{cases} 0 & \text{if } \omega \ll \omega_0 \\ 2\log\omega_0 - 2\log\omega & \text{if } \omega \gg \omega_0, \end{cases}$$

$$\angle G(i\omega) \approx \begin{cases} 0 & \text{if } \omega \ll \omega_0 \\ -180 & \text{if } \omega \gg \omega_0. \end{cases}$$

The Bode plot is shown in Figure 9.13b. Note that the asymptotic approximation is poor near $\omega = \omega_0$ and that the Bode plot depends strongly on ζ near this frequency.

Given the Bode plots of the basic functions, we can now sketch the frequency response for a more general system. The following example illustrates the basic idea.

Example 9.11 Asymptotic approximation for a transfer function
Consider the transfer function given by

$$G(s) = \frac{k(s+b)}{(s+a)(s^2 + 2\zeta\omega_0 s + \omega_0^2)}, \qquad a \ll b \ll \omega_0.$$

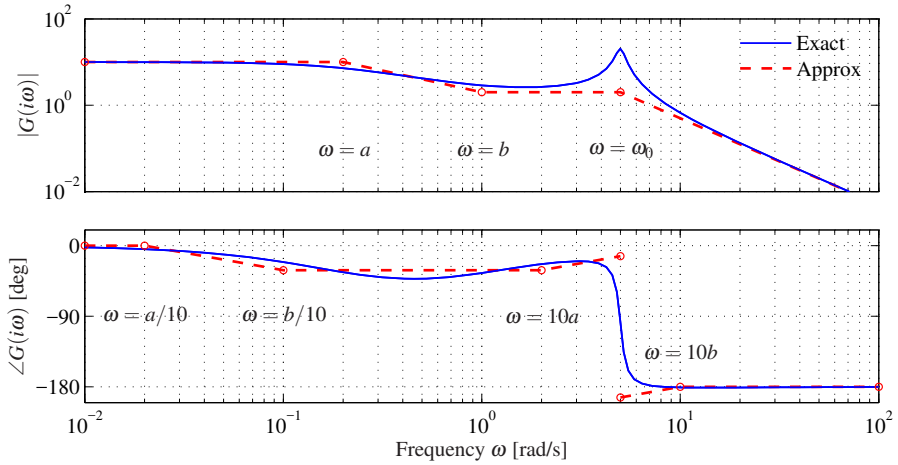


Figure 9.14: Asymptotic approximation to a Bode plot. The thin line is the Bode plot for the transfer function $G(s) = k(s+b)/(s+a)(s^2+2\zeta\omega_0 s+\omega_0^2)$, where $a \ll b \ll \omega_0$. Each segment in the gain and phase curves represents a separate portion of the approximation, where either a pole or a zero begins to have effect. Each segment of the approximation is a straight line between these points at a slope given by the rules for computing the effects of poles and zeros.

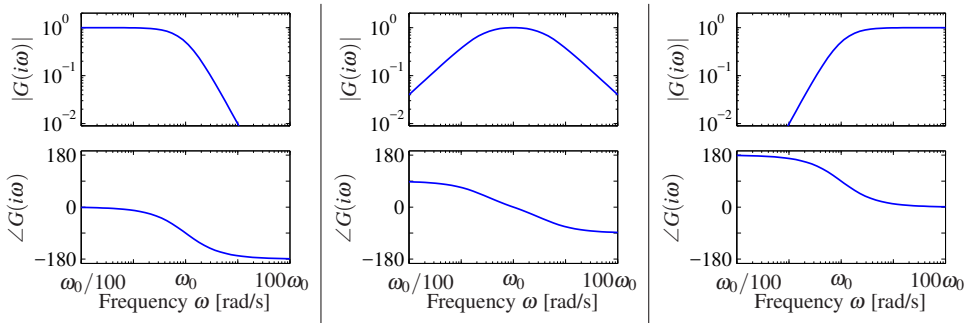
The Bode plot for this transfer function appears in Figure 9.14, with the complete transfer function shown as a solid curve and the asymptotic approximation shown as a dashed curve.

We begin with the gain curve. At low frequency, the magnitude is given by

$$G(0) = \frac{kb}{a\omega_0^2}.$$

When we reach $\omega = a$, the effect of the pole begins and the gain decreases with slope -1. At $\omega = b$, the zero comes into play and we increase the slope by 1, leaving the asymptote with net slope 0. This slope is used until the effect of the second-order pole is seen at $\omega = \omega_0$, at which point the asymptote changes to slope -2. We see that the gain curve is fairly accurate except in the region of the peak due to the second-order pole (indicating that for this case ζ is reasonably small).

The phase curve is more complicated since the effect of the phase stretches out much further. The effect of the pole begins at $\omega = a/10$, at which point we change from phase 0 to a slope of -45°/decade. The zero begins to affect the phase at $\omega = b/10$, producing a flat section in the phase. At $\omega = 10a$ the phase contributions from the pole end, and we are left with a slope of +45°/decade (from the zero). At the location of the second-order pole, $s \approx i\omega_0$, we get a jump in phase of -180°. Finally, at $\omega = 10b$ the phase contributions of the zero end, and we are left with a phase of -180 degrees. We see that the straight-line approximation for the phase is not as accurate as it was for the gain curve, but it does capture the basic features of the phase changes as a function of frequency. ∇



$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

(a) Low-pass filter

$$G(s) = \frac{2\zeta\omega_0 s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

(b) Band-pass filter

$$G(s) = \frac{s^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

(c) High-pass filter

Figure 9.15: Bode plots for low-pass, band-pass and high-pass filters. The top plots are the gain curves and the bottom plots are the phase curves. Each system passes frequencies in a different range and attenuates frequencies outside of that range.

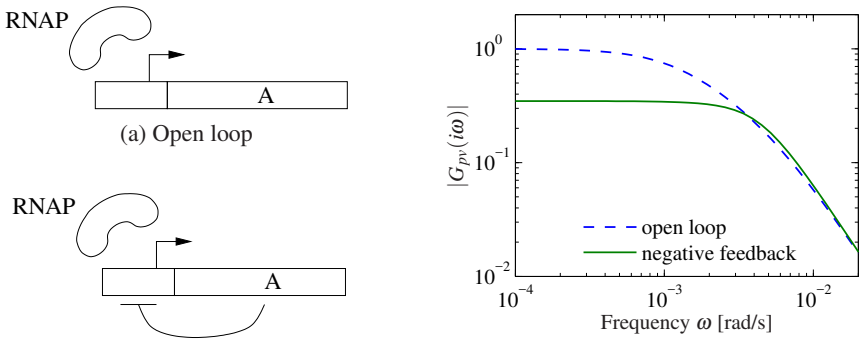
The Bode plot gives a quick overview of a system. Since any signal can be decomposed into a sum of sinusoids, it is possible to visualize the behavior of a system for different frequency ranges. The system can be viewed as a filter that can change the amplitude (and phase) of the input signals according to the frequency response. For example, if there are frequency ranges where the gain curve has constant slope and the phase is close to zero, the action of the system for signals with these frequencies can be interpreted as a pure gain. Similarly, for frequencies where the slope is +1 and the phase close to 90°, the action of the system can be interpreted as a differentiator, as shown in Figure 9.12.

Three common types of frequency responses are shown in Figure 9.15. The system in Figure 9.15a is called a *low-pass filter* because the gain is constant for low frequencies and drops for high frequencies. Notice that the phase is zero for low frequencies and $-180°$ for high frequencies. The systems in Figure 9.15b and c are called a *band-pass filter* and *high-pass filter* for similar reasons.

To illustrate how different system behaviors can be read from the Bode plots we consider the band-pass filter in Figure 9.15b. For frequencies around $\omega = \omega_0$, the signal is passed through with no change in gain. However, for frequencies well below or well above ω_0, the signal is attenuated. The phase of the signal is also affected by the filter, as shown in the phase curve. For frequencies below $\omega_0/100$ there is a phase lead of 90°, and for frequencies above $100\omega_0$ there is a phase lag of 90°. These actions correspond to differentiation and integration of the signal in these frequency ranges.

Example 9.12 Transcriptional regulation

Consider a genetic circuit consisting of a single gene. We wish to study the response of the protein concentration to fluctuations in the mRNA dynamics. We consider two cases: a *constitutive promoter* (no regulation) and self-repression



(a) Open loop

(b) Negative feedback

(c) Frequency response

Figure 9.16: Noise attenuation in a genetic circuit. The open loop system (a) consists of a constitutive promoter, while the closed loop circuit (b) is self-regulated with negative feedback (repressor). The frequency response for each circuit is shown in (c).

(negative feedback), illustrated in Figure 9.16. The dynamics of the system are given by

$$\frac{dm}{dt} = \alpha(p) - \gamma m - v, \qquad \frac{dp}{dt} = \beta m - \delta p,$$

where v is a disturbance term that affects mRNA transcription.

For the case of no feedback we have $\alpha(p) = \alpha_0$, and when $v = 0$ the system has an equilibrium point at $m_e = \alpha_0/\gamma$, $p_e = \beta\alpha_0/(\delta\gamma)$. The open loop transfer function from v to p is given by

$$G_{pv}^{\rm ol}(s) = \frac{-\beta}{(s+\gamma)(s+\delta)}.$$

For the case of negative regulation, we have

$$\alpha(p) = \frac{\alpha_1}{1+kp^n} + \alpha_0,$$

and the equilibrium points satisfy

$$m_e = \frac{\delta}{\beta} p_e, \qquad \frac{\alpha}{1+kp_e^n} + \alpha_0 = \gamma m_e = \frac{\gamma\delta}{\beta} p_e.$$

The resulting transfer function is given by

$$G_{pv}^{\rm cl}(s) = \frac{\beta}{(s+\gamma)(s+\delta)+\beta\sigma}, \qquad \sigma = \frac{n\alpha_1 k p_e^{n-1}}{(1+kp_e^n)^2}.$$

Figure 9.16c shows the frequency response for the two circuits. We see that the feedback circuit attenuates the response of the system to disturbances with low-frequency content but slightly amplifies disturbances at high frequency (compared to the open loop system). Notice that these curves are very similar to the frequency response curves for the op amp shown in Figure 9.3. ∇

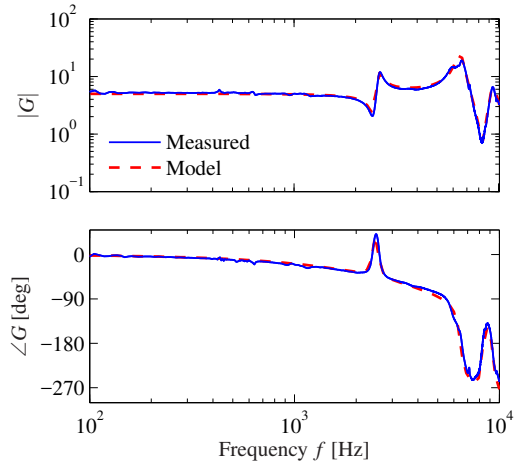


Figure 9.17: Frequency response of a preloaded piezoelectric drive for an atomic force microscope. The Bode plot shows the response of the measured transfer function (solid) and the fitted transfer function (dashed).

Transfer Functions from Experiments

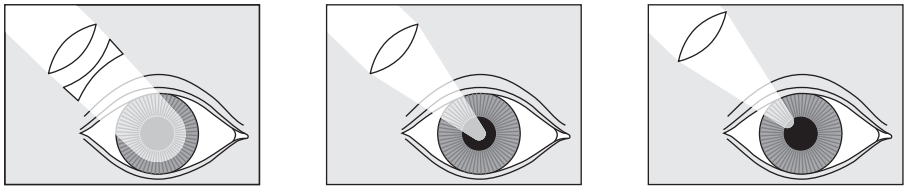
The transfer function of a system provides a summary of the input/output response and is very useful for analysis and design. We can often build an input/output model for a given application by directly measuring the frequency response and fitting a transfer function to it. To do so, we perturb the input to the system using a sinusoidal signal at a fixed frequency. When steady state is reached, the amplitude ratio and the phase lag give the frequency response for the excitation frequency. The complete frequency response is obtained by sweeping over a range of frequencies.

By using correlation techniques it is possible to determine the frequency response very accurately, and an analytic transfer function can be obtained from the frequency response by curve fitting. The success of this approach has led to instruments and software that automate this process, called *spectrum analyzers*. We illustrate the basic concept through two examples.

Example 9.13 Atomic force microscope

To illustrate the utility of spectrum analysis, we consider the dynamics of the atomic force microscope, described in Section 4.5. Experimental determination of the frequency response is particularly attractive for this system because its dynamics are very fast and hence experiments can be done quickly. A typical example is given in Figure 9.17, which shows an experimentally determined frequency response (solid line). In this case the frequency response was obtained in less than a second. The transfer function

$$G(s) = \frac{k\omega_2^2\omega_3^2\omega_5^2(s^2+2\zeta_1\omega_1 s+\omega_1^2)(s^2+2\zeta_4\omega_4 s+\omega_4^2)e^{-s\tau}}{\omega_1^2\omega_4^2(s^2+2\zeta_2\omega_2 s+\omega_2^2)(s^2+2\zeta_3\omega_3 s+\omega_3^2)(s^2+2\zeta_5\omega_5 s+\omega_5^2)},$$



(a) Closed loop (b) Open loop (c) High gain

Figure 9.18: Light stimulation of the eye. In (a) the light beam is so large that it always covers the whole pupil, giving closed loop dynamics. In (b) the light is focused into a beam which is so narrow that it is not influenced by the pupil opening, giving open loop dynamics. In (c) the light beam is focused on the edge of the pupil opening, which has the effect of increasing the gain of the system since small changes in the pupil opening have a large effect on the amount of light entering the eye. From Stark [Sta68].

with $\omega_k = 2\pi f_k$ and $f_1 = 2.42$ kHz, $\zeta_1 = 0.03$, $f_2 = 2.55$ kHz, $\zeta_2 = 0.03$, $f_3 = 6.45$ kHz, $\zeta_3 = 0.042$, $f_4 = 8.25$ kHz, $\zeta_4 = 0.025$, $f_5 = 9.3$ kHz, $\zeta_5 = 0.032$, $\tau = 10^{-4}$ s and $k = 5$, was fitted to the data (dashed line). The frequencies ω_1 and ω_4 associated with the zeros are located where the gain curve has minima, and the frequencies ω_2, ω_3 and ω_5 associated with the poles are located where the gain curve has local maxima. The relative damping ratios are adjusted to give a good fit to maxima and minima. When a good fit to the gain curve is obtained, the time delay is adjusted to give a good fit to the phase curve. The piezo drive is preloaded, and a simple model of its dynamics is derived in Exercise 4.7. The pole at 2.55 kHz corresponds to the trampoline mode derived in the exercise; the other resonances are higher modes.

∇

Example 9.14 Pupillary light reflex dynamics

The human eye is an organ that is easily accessible for experiments. It has a control system that adjusts the pupil opening to regulate the light intensity at the retina.

This control system was explored extensively by Stark in the 1960s [Sta68]. To determine the dynamics, light intensity on the eye was varied sinusoidally and the pupil opening was measured. A fundamental difficulty is that the closed loop system is insensitive to internal system parameters, so analysis of a closed loop system thus gives little information about the internal properties of the system. Stark used a clever experimental technique that allowed him to investigate both open and closed loop dynamics. He excited the system by varying the intensity of a light beam focused on the eye and measured pupil area, as illustrated in Figure 9.18. By using a wide light beam that covers the whole pupil, the measurement gives the closed loop dynamics. The open loop dynamics were obtained by using a narrow beam, which is small enough that it is not influenced by the pupil opening. The result of one experiment for determining open loop dynamics is given in Figure 9.19. Fitting a transfer function to the gain curve gives a good fit for $G(s) = 0.17/(1+0.08s)^3$. This curve gives a poor fit to the phase curve as shown by the dashed curve in Figure 9.19. The fit to the phase curve is improved by

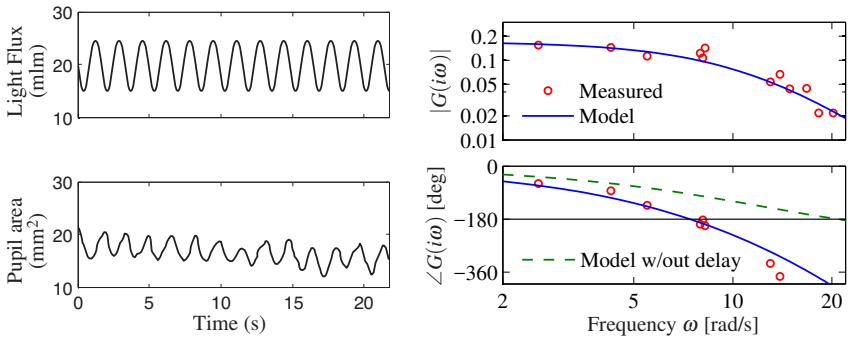


Figure 9.19: Sample curves from an open loop frequency response of the eye (left) and a Bode plot for the open loop dynamics (right). The solid curve shows a fit of the data using a third-order transfer function with 0.2 s time delay. The dashed curve in the Bode plot is the phase of the system without time delay, showing that the delay is needed to properly capture the phase. (Figure redrawn from the data of Stark [Sta68].)

adding a 0.2 s time delay, which leaves the gain curve unchanged while substantially modifying the phase curve. The final fit gives the model

$$G(s) = \frac{0.17}{(1+0.08s)^3} e^{-0.2s}.$$

The Bode plot of this is shown with solid curves in Figure 9.19. Modeling of the pupillary reflex from first principles is discussed in detail in [KS01]. ∇

Notice that for both the AFM drive and pupillary dynamics it is not easy to derive appropriate models from first principles. In practice, it is often fruitful to use a combination of analytical modeling and experimental identification of parameters. Experimental determination of frequency response is less attractive for systems with slow dynamics because the experiment takes a long time.

9.6 Further Reading

The idea of characterizing a linear system by its steady-state response to sinusoids was introduced by Fourier in his investigation of heat conduction in solids [Fou07]. Much later, it was used by the electrical engineer Steinmetz who introduced the $i\omega$ method for analyzing electrical circuits. Transfer functions were introduced via the Laplace transform by Gardner Barnes [GB42], who also used them to calculate the response of linear systems. The Laplace transform was very important in the early phase of control because it made it possible to find transients via tables (see, e.g., [JNP47]). Combined with block diagrams, transfer functions and Laplace transforms provided powerful techniques for dealing with complex systems. Calculation of responses based on Laplace transforms is less important today, when responses of linear systems can easily be generated using computers. There are many excellent books on the use of Laplace transforms and transfer functions for

modeling and analysis of linear input/output systems. Traditional texts on control such as [DB04], [FPEN05] and [Oga01] are representative examples. Pole/zero cancellation was one of the mysteries of early control theory. It is clear that common factors can be canceled in a rational function, but cancellations have system theoretical consequences that were not clearly understood until Kalman's decomposition of a linear system was introduced [KHN63]. In the following chapters, we will use transfer functions extensively to analyze stability and to describe model uncertainty.

Exercises

9.1 Let $G(s)$ be the transfer function for a linear system. Show that if we apply an input $u(t) = A\sin(\omega t)$, then the steady-state output is given by $y(t) = |G(i\omega)|A\sin(\omega t + \arg G(i\omega))$. (Hint: Start by showing that the real part of a complex number is a linear operation and then use this fact.)

9.2 Consider the system

$$\frac{dx}{dt} = ax + u.$$

Compute the exponential response of the system and use this to derive the transfer function from u to x. Show that when $s = a$, a pole of the transfer function, the response to the exponential input $u(t) = e^{st}$ is $x(t) = e^{at}x(0) + te^{at}$.

9.3 (Inverted pendulum) A model for an inverted pendulum was introduced in Example 3.2. Neglecting damping and linearizing the pendulum around the upright position gives a linear system characterized by the matrices

$$A = \begin{pmatrix} 0 & 1 \\ mgl/J_t & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1/J_t \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad D = 0.$$

Determine the transfer function of the system.

9.4 (Solutions corresponding to poles and zeros) Consider the differential equation

$$\frac{d^n y}{dt^n} + a_1\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_n y = b_1\frac{d^{n-1}u}{dt^{n-1}} + b_2\frac{d^{n-2}u}{dt^{n-2}} + \cdots + b_n u.$$

(a) Let λ be a root of the characteristic polynomial

$$s^n + a_1 s^{n-1} + \cdots + a_n = 0.$$

Show that if $u(t) = 0$, the differential equation has the solution $y(t) = e^{\lambda t}$.

(b) Let κ be a zero of the polynomial

$$b(s) = b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_n.$$

Show that if the input is $u(t) = e^{\kappa t}$, then there is a solution to the differential equation that is identically zero.

9.5 (Operational amplifier) Consider the operational amplifier described in Section 4.3 and analyzed in Example 9.3. A PI controller can be constructed using an op amp by replacing the resistor R_2 with a resistor and capacitor in series, as shown in Figure 4.10. The resulting transfer function of the circuit is given by

$$G(s) = -\left(R_2 + \frac{1}{Cs}\right) \cdot \left(\frac{kCs}{\big((k+1)R_1C + R_2C\big)s + 1}\right),$$

where k is the gain of the op amp, R_1 and R_2 are the resistances in the compensation network and C is the capacitance.

(a) Sketch the Bode plot for the system under the assumption that $k \gg R_2 > R_1$. You should label the key features in your plot, including the gain and phase at low frequency, the slopes of the gain curve, the frequencies at which the gain changes slope, etc.

(b) Suppose now that we include some dynamics in the amplifier, as outlined in Example 8.1. This would involve replacing the gain k with the transfer function

$$H(s) = \frac{k}{1 + sT}.$$

Compute the resulting transfer function for the system (i.e., replace k with $H(s)$) and find the poles and zeros assuming the following parameter values

$$\frac{R_2}{R_1} = 100, \qquad k = 10^6, \qquad R_2C = 1, \qquad T = 0.01.$$

(c) Sketch the Bode plot for the transfer function in part (b) using straight line approximations and compare this to the exact plot of the transfer function (using MATLAB). Make sure to label the important features in your plot.

9.6 (Transfer function for state space system) Consider the linear state space system

$$\frac{dx}{dt} = Ax + Bu, \qquad y = Cx.$$

(a) Show that the transfer function is

$$G(s) = \frac{b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_n}{s^n + a_1 s^{n-1} + \cdots + a_n},$$

where

$$b_1 = CB, \quad b_2 = CAB + a_1 CB, \quad \ldots, \quad b_n = CA^{n-1}B + a_1 CA^{n-2}B + \cdots + a_{n-1}CB$$

and $\lambda(s) = s^n + a_1 s^{n-1} + \cdots + a_n$ is the characteristic polynomial for A.

(b) Compute the transfer function for a linear system in reachable canonical form and show that it matches the transfer function given above.

9.7 (Kalman decomposition) Show that the transfer function of a system depends only on the dynamics in the reachable and observable subspace of the Kalman decomposition. (Hint: Consider the representation given by equation (8.19).)

9.8 Using block diagram algebra, show that the transfer functions from v to y and w to y in Figure 9.8 are given by

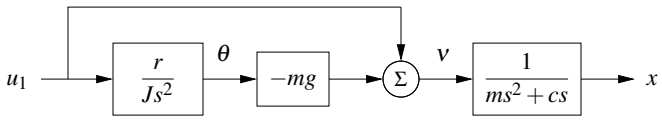
$$G_{yv} = \frac{P}{1+PC} \qquad G_{yw} = \frac{1}{1+PC}.$$

9.9 (Bode plot for a simple zero) Show that the Bode plot for transfer function $G(s) = (s+a)/a$ can be approximated by

$$\log|G(i\omega)| \approx \begin{cases} 0 & \text{if } \omega < a \\ \log\omega - \log a & \text{if } \omega > a, \end{cases}$$

$$\angle G(i\omega) \approx \begin{cases} 0 & \text{if } \omega < a/10 \\ 45 + 45(\log\omega - \log a) & a/10 < \omega < 10a \\ 90 & \text{if } \omega > 10a. \end{cases}$$

9.10 (Vectored thrust aircraft) Consider the lateral dynamics of a vectored thrust aircraft as described in Example 3.11. Show that the dynamics can be described using the following block diagram:



Use this block diagram to compute the transfer functions from u_1 to θ and x and show that they satisfy

$$H_{\theta u_1} = \frac{r}{Js^2}, \qquad H_{xu_1} = \frac{Js^2 - mgr}{Js^2(ms^2+cs)}.$$

9.11 (Common poles) Consider a closed loop system of the form of Figure 9.8, with $F = 1$ and P and C having a pole/zero cancellation. Show that if each system is written in state space form, the resulting closed loop system is not reachable and not observable.

9.12 (Congestion control) Consider the congestion control model described in Section 4.4. Let w represent the individual window size for a set of N identical sources, q represent the end-to-end probability of a dropped packet, b represent the number of packets in the router's buffer and p represent the probability that a packet is dropped by the router. We write $\bar{w} = Nw$ to represent the total number of packets being received from all N sources. Show that the linearized model can be described

by the transfer functions

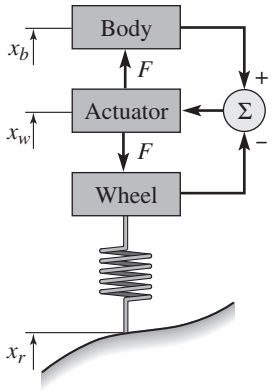
$$G_{b\bar{w}}(s) = \frac{e^{-\tau_f s}}{\tau_e s + e^{-\tau_f s}}, \qquad G_{\bar{w}q}(s) = -\frac{N}{q_e(\tau_e s + q_e w_e)}, \qquad G_{pb}(s) = \rho,$$

where (w_e, b_e) is the equilibrium point for the system, τ_e is the steady-state round-trip time and τ_f is the forward propagation time.

9.13 (Inverted pendulum with PD control) Consider the normalized inverted pendulum system, whose transfer function is given by $P(s) = 1/(s^2 - 1)$ (Exercise 9.3). A proportional-derivative control law for this system has transfer function $C(s) = k_p + k_d s$ (see Table 9.1). Suppose that we choose $C(s) = \alpha(s - 1)$. Compute the closed loop dynamics and show that the system has good tracking of reference signals but does not have good disturbance rejection properties.

9.14 (Vehicle suspension [HB90]) Active and passive damping are used in cars to give a smooth ride on a bumpy road. A schematic diagram of a car with a damping system in shown in the figure below.

(Porter Class I race car driven by Todd Cuffaro)



This model is called a *quarter car model*, and the car is approximated with two masses, one representing one fourth of the car body and the other a wheel. The actuator exerts a force F between the wheel and the body based on feedback from the distance between the body and the center of the wheel (the *rattle space*).

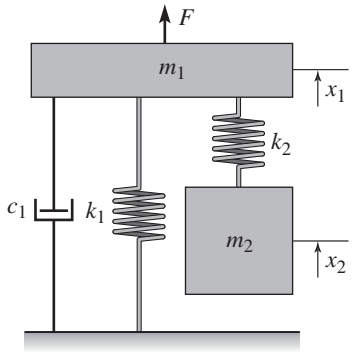
Let x_b, x_w and x_r represent the heights of body, wheel and road measured from their equilibria. A simple model of the system is given by Newton's equations for the body and the wheel,

$$m_b \ddot{x}_b = F, \qquad m_w \ddot{x}_w = -F + k_t(x_r - x_w),$$

where m_b is a quarter of the body mass, m_w is the effective mass of the wheel including brakes and part of the suspension system (the *unsprung mass*) and k_t is the tire stiffness. For a conventional damper consisting of a spring and a damper, we have $F = k(x_w - x_b) + c(\dot{x}_w - \dot{x}_b)$. For an active damper the force F can be more general and can also depend on riding conditions. Rider comfort can be characterized by the transfer function G_{ax_r} from road height x_r to body acceleration $a = \ddot{x}_b$. Show that this transfer function has the property $G_{ax_r}(i\omega_t) = k_t/m_b$,

where $\omega_t = \sqrt{k_t/m_w}$ (the *tire hop frequency*). The equation implies that there are fundamental limitations to the comfort that can be achieved with any damper.

9.15 (Vibration absorber) Damping vibrations is a common engineering problem. A schematic diagram of a damper is shown below:



The disturbing vibration is a sinusoidal force acting on mass m_1, and the damper consists of the mass m_2 and the spring k_2. Show that the transfer function from disturbance force to height x_1 of the mass m_1 is

$$G_{x_1F} = \frac{m_2 s^2 + k_2}{m_1 m_2 s^4 + m_2 c_1 s^3 + (m_1 k_2 + m_2(k_1 + k_2))s^2 + k_2 c_1 s + k_1 k_2}.$$

How should the mass m_2 and the stiffness k_2 be chosen to eliminate a sinusoidal oscillation with frequency ω_0. (More details are vibration absorbers is given in the classic text by Den Hartog [DH85, pp. 87–93].)

Chapter Ten
Frequency Domain Analysis

Mr. Black proposed a negative feedback repeater and proved by tests that it possessed the advantages which he had predicted for it. In particular, its gain was constant to a high degree, and it was linear enough so that spurious signals caused by the interaction of the various channels could be kept within permissible limits. For best results the feedback factor $\mu\beta$ had to be numerically much larger than unity. The possibility of stability with a feedback factor larger than unity was puzzling.

Harry Nyquist, "The Regeneration Theory," 1956 [Nyq56].

In this chapter we study how the stability and robustness of closed loop systems can be determined by investigating how sinusoidal signals of different frequencies propagate around the feedback loop. This technique allows us to reason about the *closed loop behavior* of a system through the frequency domain properties of the *open loop* transfer function. The Nyquist stability theorem is a key result that provides a way to analyze stability and introduce measures of degrees of stability.

10.1 The Loop Transfer Function

Understanding how the behavior of a closed loop system is influenced by the properties of its open loop is tricky. Indeed, as the quote from Nyquist above illustrates, the behavior of feedback systems can often be puzzling. However, using the mathematical framework of transfer functions provides an elegant way to reason about such systems, which we call *loop analysis*.

The basic idea of loop analysis is to trace how a sinusoidal signal propagates in the feedback loop and explore the resulting stability by investigating if the propagated signal grows or decays. This is easy to do because the transmission of sinusoidal signals through a linear dynamical system is characterized by the frequency response of the system. The key result is the Nyquist stability theorem, which provides a great deal of insight regarding the stability of a system. Unlike proving stability with Lyapunov functions, studied in Chapter 5, the Nyquist criterion allows us to determine more than just whether a system is stable or unstable. It provides a measure of the degree of stability through the definition of stability margins. The Nyquist theorem also indicates how an unstable system should be changed to make it stable, which we shall study in detail in Chapters 11–13.

Consider the system in Figure 10.1a. The traditional way to determine if the closed loop system is stable is to investigate if the closed loop characteristic polynomial has all its roots in the left half-plane. If the process and the controller have rational transfer functions $P(s) = n_p(s)/d_p(s)$ and $C(s) = n_c(s)/d_c(s)$, then the

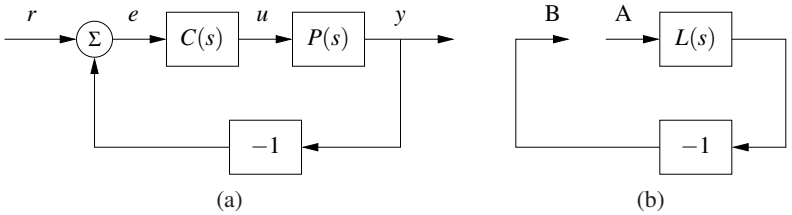


Figure 10.1: The loop transfer function. The stability of the feedback system (a) can be determined by tracing signals around the loop. Letting $L = PC$ represent the loop transfer function, we break the loop in (b) and ask whether a signal injected at the point A has the same magnitude and phase when it reaches point B.

closed loop system has the transfer function

$$G_{yr}(s) = \frac{PC}{1+PC} = \frac{n_p(s)n_c(s)}{d_p(s)d_c(s)+n_p(s)n_c(s)},$$

and the characteristic polynomial is

$$\lambda(s) = d_p(s)d_c(s)+n_p(s)n_c(s).$$

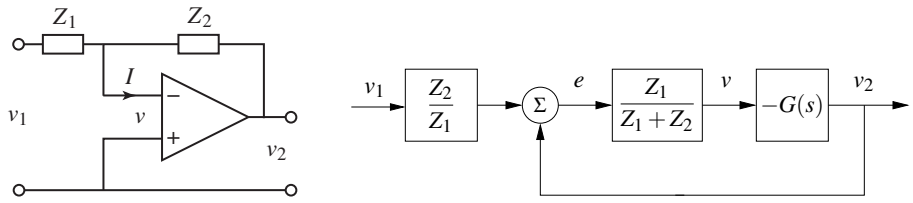
To check stability, we simply compute the roots of the characteristic polynomial and verify that they each have negative real part. This approach is straightforward but it gives little guidance for design: it is not easy to tell how the controller should be modified to make an unstable system stable.

Nyquist's idea was to first investigate conditions under which oscillations can occur in a feedback loop. To study this, we introduce the *loop transfer function* $L(s) = P(s)C(s)$, which is the transfer function obtained by breaking the feedback loop, as shown in Figure 10.1b. The loop transfer function is simply the transfer function from the input at position A to the output at position B multiplied by -1 (to account for the usual convention of negative feedback).

Assume that a sinusoid of frequency ω_0 is injected at point A. In steady state the signal at point B will also be a sinusoid with the frequency ω_0. It seems reasonable that an oscillation can be maintained if the signal at B has the same amplitude and phase as the injected signal because we can then disconnect the injected signal and connect A to B. Tracing signals around the loop, we find that the signals at A and B are identical if there is a frequency ω_0 such that

$$L(i\omega_c 0 = -1, \tag{10.1}$$

which then provides a condition for maintaining an oscillation. The condition (10.1) implies that the Nyquist plot of the loop transfer function goes through the point $L = -1$, which is called the *critical point*. Let ω_c represent a frequency at which $\angle L(i\omega_c) = 180^\circ$, Intuitively it seems reasonable that the system is stable if $|L(i\omega_c)| < 1$, which means that the signal at point B will have smaller amplitude than the injected signal. This is essentially true, but there are several subtleties that require a proper mathematical analysis leading to Nyquist's stability criterion.



(a) Amplifier circuit (b) Block diagram

Figure 10.2: Loop transfer function for an op amp. The op amp circuit (a) has a nominal transfer function $v_2/v_1 = Z_2(s)/Z_1(s)$, where Z_1 and Z_2 are the impedances of the circuit elements. The system can be represented by its block diagram (b), where we now include the op amp dynamics $G(s)$. The loop transfer function is $L = Z_1 G/(Z_1 + Z_2)$.

Example 10.1 Operational amplifier circuit

Consider the op amp circuit in Figure 10.2a, where Z_1 and Z_2 are the transfer functions of the feedback elements from voltage to current. There is feedback because voltage v_2 is related to voltage v through the transfer function $-G$ describing the op amp dynamics and voltage v is related to voltage v_2 through the transfer function $Z_1/(Z_1 + Z_2)$. The loop transfer function is thus

$$L = \frac{GZ_1}{Z_1 + Z_2}. \tag{10.2}$$

Assuming that the current I is zero, the current through the elements Z_1 and Z_2 is the same, which implies

$$\frac{v_1 - v}{Z_1} = \frac{v - v_2}{Z_2}.$$

Solving for v gives

$$v = \frac{Z_2 v_1 + Z_1 v_2}{Z_1 + Z_2} = \frac{Z_2 v_1 - Z_1 G v}{Z_1 + Z_2} = \frac{Z_2}{Z_1}\frac{L}{G} v_1 - Lv.$$

Hence

$$v = \frac{L}{G(1+L)} v_1.$$

Since $v_2 = -Gv$ the input/output relation for the circuiit is then described by the transfer function

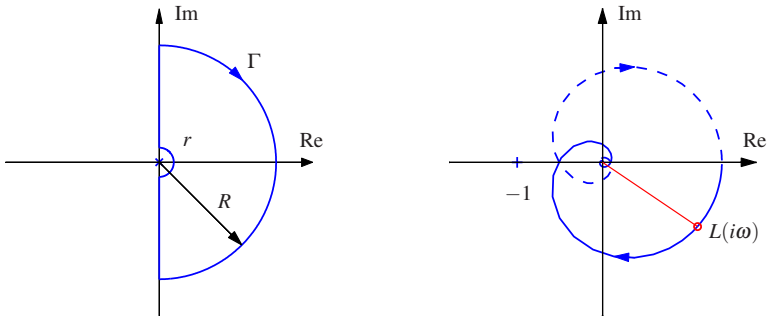
$$G_{v_2 v_1} = -\frac{Z_2}{Z_1}\frac{L}{1+L}.$$

A block diagram is shown in Figure 10.2b. It follows from (10.1) that the condition for oscillation of the op amp circuit is

$$L(i\omega) = \frac{Z_1(i\omega)G(i\omega)}{Z_1(i\omega) + Z_2(i\omega)} = -1 \tag{10.3}$$

∇

One of the powerful concepts embedded in Nyquist's approach to stability analysis is that it allows us to study the stability of the feedback system by looking at



(a) Nyquist D contour (b) Nyquist plot

Figure 10.3: The Nyquist contour Γ and the Nyquist plot. The Nyquist contour (a) encloses the right half-plane, with a small semicircle around any poles of $L(s)$ at the origin or on the imaginary axis (illustrated here at the origin) and an arc whose radius R extends towards ∞. The Nyquist plot (b) is the image of the loop transfer function $L(s)$ when s traverses Γ in the clockwise direction. The solid line corresponds to $\omega > 0$, and the dashed line to $\omega < 0$. The gain and phase at the frequency ω are $g = |L(i\omega)|$ and $\varphi = \angle L(i\omega)$. The curve is generated for $L(s) = 1.4e^{-s}/(s+1)^2$.

properties of the loop transfer function. The advantage of doing this is that it is easy to see how the controller should be chosen to obtain a desired loop transfer function. For example, if we change the gain of the controller, the loop transfer function will be scaled accordingly and the critical point is avoided. A simple way to stabilize an unstable system is to reduce the gain or to modify the controller so that the critical point -1 is avoided. Different ways to do this, called loop shaping, will be developed and will be discussed in Chapter 12.

10.2 The Nyquist Criterion

In this section we present Nyquist's criterion for determining the stability of a feedback system through analysis of the loop transfer function. We begin by introducing a convenient graphical tool, the Nyquist plot, and show how it can be used to ascertain stability.

The Nyquist Plot

We saw in the previous chapter that the dynamics of a linear system can be represented by its frequency response and graphically illustrated by a Bode plot. To study the stability of a system, we will make use of a different representation of the frequency response called a *Nyquist plot*. The Nyquist plot of the loop transfer function $L(s)$ is formed by tracing $s \in \mathbb{C}$ around the Nyquist "D contour," consisting of the imaginary axis combined with an arc at infinity connecting the endpoints of the imaginary axis. The contour, denoted as $\Gamma \in \mathbb{C}$, is illustrated in Figure 10.3a. The image of $L(s)$ when s traverses Γ gives a closed curve in the complex plane and is referred to as the Nyquist plot for $L(s)$, as shown in Figure 10.3b. Note that

if the transfer function $L(s)$ goes to zero as s gets large (the usual case), then the portion of the contour "at infinity" maps to the origin. Furthermore, the portion of the plot corresponding to $\omega < 0$ is the mirror image of the portion with $\omega > 0$.

There is a subtlety in the Nyquist plot when the loop transfer function has poles on the imaginary axis because the gain is infinite at the poles. To solve this problem, we modify the contour Γ to include small deviations that avoid any poles on the imaginary axis, as illustrated in Figure 10.3a (assuming a pole of $L(s)$ at the origin). The deviation consists of a small semicircle to the right of the imaginary axis pole location.

When the loop transfer function has no poles in the right half plane the Nyquist stability criterion is.

Theorem 10.1 (Simplified Nyquist criterion)**.** *Let $L(s)$ be the loop transfer function for a negative feedback system (as shown in Figure 10.1a) and assume that L has no poles in the open right half-plane ($\mathrm{Re}\, s \geq 0$). Then the closed loop system is stable if and only if the closed contour given by $\Omega = \{L(i\omega) : -\infty < \omega < \infty\} \subset \mathbb{C}$ has no net encirclements of the critical point $s = -1$.*

The following conceptual procedure can be used to determine that there are no encirclements. Fix a pin at the critical point $s = -1$, orthogonal to the plane. Attach a string with one end at the critical point and the other on the Nyquist plot. Let the end of the string attached to the Nyquist curve traverse the whole curve. There are no encirclements if the string does not wind up on the pin when the curve is encircled.

Example 10.2 Third-order system
Consider a third-order transfer function

$$L(s) = \frac{1}{(s+a)^3}.$$

To compute the Nyquist plot we start by evaluating points on the imaginary axis $s = i\omega$, which yields

$$L(i\omega) = \frac{1}{(i\omega + a)^3} = \frac{(a - i\omega)^3}{(a^2+\omega^2)^3} = \frac{a^3 - 3a\omega^2}{(a^2+\omega^2)^3} + i\,\frac{\omega^3 - 3a^2\omega}{(a^2+\omega^2)^3}.$$

This is plotted in the complex plane in Figure 10.4, with the points corresponding to $\omega > 0$ drawn as a solid line and $\omega < 0$ as a dashed line. Notice that these curves are mirror images of each other.

To complete the Nyquist plot, we compute $L(s)$ for s on the outer arc of the Nyquist D contour. This arc has the form $s = Re^{i\theta}$ for $R \to \infty$. This gives

$$L(Re^{i\theta}) = \frac{1}{(Re^{i\theta} + a)^3} \to 0 \quad \text{as} \quad R \to \infty.$$

Thus the outer arc of the D contour maps to the origin on the Nyquist plot. ∇

An alternative to computing the Nyquist plot explicitly is to determine the plot from the frequency response (Bode plot), which gives the Nyquist curve for $s = i\omega$,

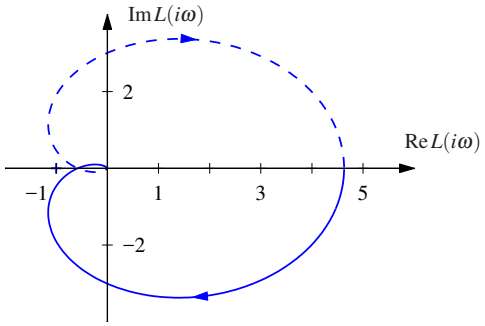


Figure 10.4: Nyquist plot for a third-order transfer function. The Nyquist plot consists of a trace of the loop transfer function $L(s) = 1/(s+a)^3$ with $a = 0.6$. The solid line represents the portion of the transfer function along the positive imaginary axis, and the dashed line the negative imaginary axis. The outer arc of the D contour maps to the origin.

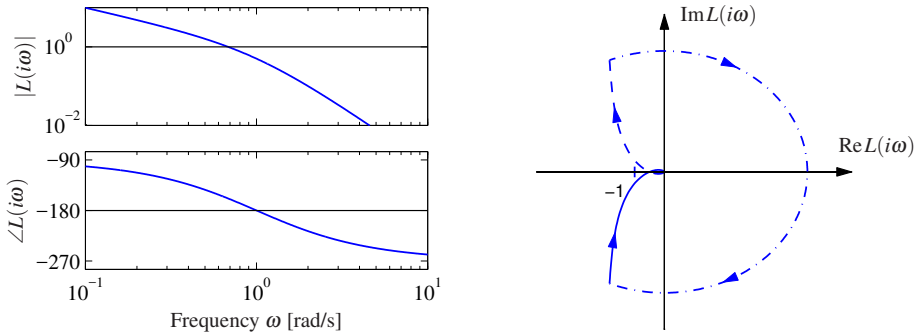
$\omega > 0$. We start by plotting $L(i\omega)$ from $\omega = 0$ to $\omega = \infty$, which can be read off from the magnitude and phase of the transfer function. We then plot $L(Re^{i\theta})$ with $\theta \in [-\pi/2, \pi/2]$ and $R \to \infty$, which goes to zero if the high frequency gain of $L(i\omega)$ goes to zero (iff $L(s)$ is strictly proper). The remaining parts of the plot can be determined by taking the mirror image of the curve thus far (normally plotted using a dashed line). The plot can then be labeled with arrows corresponding to a clockwise traversal around the D contour (the same direction in which the first portion of the curve was plotted).

Example 10.3 Third-order system with a pole at the origin
Consider the transfer function

$$L(s) = \frac{k}{s(s+1)^2},$$

where the gain has the nominal value $k = 1$. The Bode plot is shown in Figure 10.5a. The system has a single pole at $s = 0$ and a double pole at $s = -1$. The gain curve of the Bode plot thus has the slope -1 for low frequencies, and at the double pole $s = 1$ the slope changes to -3. For small s we have $L \approx k/s$, which means that the low-frequency asymptote intersects the unit gain line at $\omega = k$. The phase curve starts at -90° for low frequencies, it is -180° at the breakpoint $\omega = 1$ and it is -270° at high frequencies.

Having obtained the Bode plot, we can now sketch the Nyquist plot, shown in Figure 10.5b. It starts with a phase of -90° for low frequencies, intersects the negative real axis at the breakpoint $\omega = 1$ where $L(i) = -0.5$ and goes to zero along the imaginary axis for high frequencies. The small half-circle of the D contour at the origin is mapped on a large circle enclosing the right half-plane. The Nyquist curve does not encircle the critical point, and it follows from the simplified Nyquist theorem that the closed loop system is stable. Since $L(i) = -k/2$, we find the closed loop system becomes unstable if the gain is increased to $k = 2$ or beyond. ∇



(a) Bode plot (b) Nyquist plot

Figure 10.5: Sketching Nyquist and Bode plots. The loop transfer function is $L(s) = 1/(s(s+1)^2)$. The large semicircle is the map of the small semicircle of the D contour around the pole at the origin. The closed loop is stable because the Nyquist curve does not encircle the critical point. The point where the phase is -180° is marked with a circle in the Bode plot.

The Nyquist criterion does not require that $|L(i\omega_c)| < 1$ for all ω_c corresponding to a crossing of the negative real axis. Rather, it says that the number of encirclements must be zero, allowing for the possibility that the Nyquist curve could cross the negative real axis and cross back at magnitudes greater than 1. The fact that it was possible to have high feedback gains surprised the early designers of feedback amplifiers, as mentioned in the quote in the beginning of this chapter.

One advantage of the Nyquist criterion is that it tells us how a system is influenced by changes of the controller parameters. For example, it is very easy to visualize what happens when the gain is changed since this just scales the Nyquist curve.

Example 10.4 Congestion control

Consider the Internet congestion control system described in Section 4.4. Suppose we have N identical sources and a disturbance d representing an external data source, as shown in Figure 10.6a. We let w represent the individual window size for a source, q represent the end-to-end probability of a dropped packet, b represent the number of packets in the router's buffer and p represent the probability that a packet is dropped by the router. We write $\bar{w}$ for the total number of packets being received from all N sources. We also include a time delay between the router and the senders, representing the time delays between the sender and receiver.

To analyze the stability of the system, we use the transfer functions computed in Exercise 9.12:

$$\tilde{G}_{b\bar{w}}(s) = \frac{1}{\tau_e s + e^{-\tau_f s}}, \qquad G_{wq}(s) = -\frac{1}{q_e(\tau_e s + q_e w_e)}, \qquad G_{pb}(s) = \rho,$$

where (w_e, b_e) is the equilibrium point for the system, N is the number of sources, τ_e is the steady-state round-trip time and τ_f is the forward propagation time. We use $\tilde{G}_{b\bar{w}}$ to represent the transfer function with the forward time delay removed

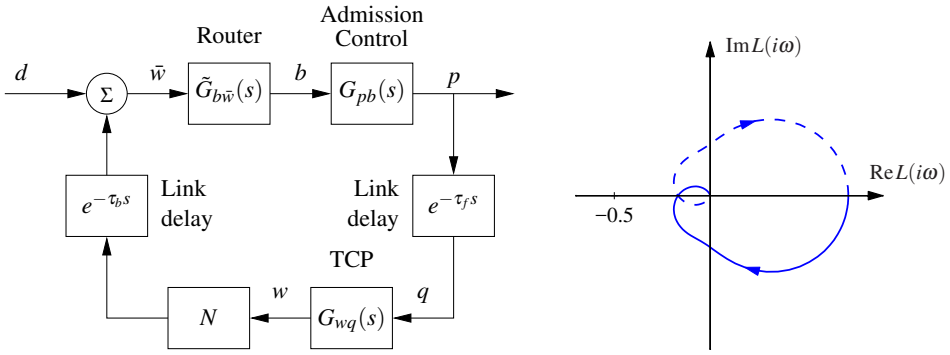


Figure 10.6: Internet congestion control. A set of N sources using TCP/Reno send messages through a single router with admission control (left). Link delays are included for the forward and backward directions. The Nyquist plot for the loop transfer function is shown on the right.

since this is accounted for as a separate block in Figure 10.6a. Similarly, $G_{wq} = G_{\bar{w}q}/N$ since we have pulled out the multiplier N as a separate block as well.

The loop transfer function is given by

$$L(s) = \rho \cdot \frac{N}{\tau_e s + e^{-\tau_f s}} \cdot \frac{1}{q_e(\tau_e s + q_e w_e)} e^{-\tau_e s}.$$

Using the fact that $q_e \approx 2N/w_e^2 = 2N^3/(\tau_e c)^2$ and $w_e = b_e/N = \tau_e c/N$ from equation (4.22), we can show that

$$L(s) = \rho \cdot \frac{N}{\tau_e s + e^{-\tau_f s}} \cdot \frac{c^3 \tau_e^3}{2N^3(c\tau_e^2 s + 2N^2)} e^{-\tau_e s}.$$

Note that we have chosen the sign of $L(s)$ to use the same sign convention as in Figure 10.1b. The exponential term representing the time delay gives significant phase above $\omega = 1/\tau_e$, and the gain at the crossover frequency will determine stability.

To check stability, we require that the gain be sufficiently small at crossover. If we assume that the pole due to the queue dynamics is sufficiently fast that the TCP dynamics are dominant, the gain at the crossover frequency ω_c is given by

$$|L(i\omega_c)| = \rho \cdot N \cdot \frac{c^3 \tau_e^3}{2N^3 c \tau_e^2 \omega_c} = \frac{\rho c^2 \tau_e}{2N^2 \omega_c}.$$

Using the Nyquist criterion, the closed loop system will be unstable if this quantity is greater than 1. In particular, for a fixed time delay, the system will become unstable as the link capacity c is increased. This indicates that the TCP protocol may not be scalable to high-capacity networks, as pointed out by Low et al. [LPD02]. Exercise 10.7 provides some ideas of how this might be overcome. ∇

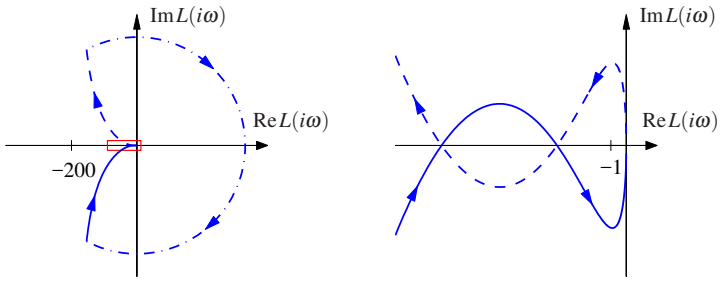


Figure 10.7: Nyquist curve for the loop transfer function $L(s) = \frac{3(s+6)^2}{s(s+1)^2}$. The plot on the right is an enlargement of the box around the origin of the plot on the left. The Nyquist curve intersects the negative real axis twice but has no net encirclements of -1.

Conditional Stability

An unstable systems can often be stabilized simply by reducing the loop gain. There are, however, situations where a system can be stabilized by increasing the gain. This was first encountered by electrical engineers in the design of feedback amplifiers, who coined the term *conditional stability*. The problem was actually a strong motivation for Nyquist to develop his theory. We will illustrate this by an example.

Example 10.5 Third-order system
Consider a feedback system with the loop transfer function

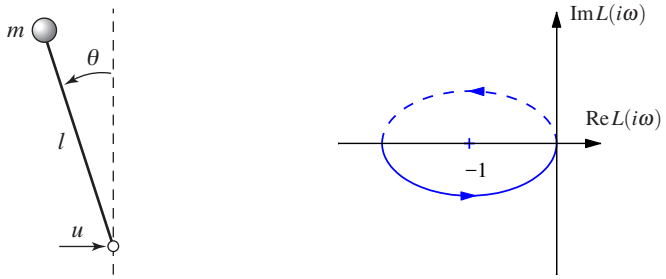
$$L(s) = \frac{3(s+6)^2}{s(s+1)^2}. \tag{10.4}$$

The Nyquist plot of the loop transfer function is shown in Figure 10.7. Notice that the Nyquist curve intersects the negative real axis twice. The first intersection occurs at $L = -12$ for $\omega = 2$, and the second at $L = -4.5$ for $\omega = 3$. The intuitive argument based on signal tracing around the loop in Figure 10.1b is misleading in this case. Injection of a sinusoid with frequency 2 rad/s and amplitude 1 at A gives, in steady state, an oscillation at B that is in phase with the input and has amplitude 12. Intuitively it seems unlikely that closing of the loop will result in a stable system. However, it follows from Nyquist's stability criterion that the system is stable because there are no net encirclements of the critical point. Note, however, that if we *decrease* the gain, then we can get an encirclement, implying that the gain must be sufficiently large for stability. ∇

General Nyquist Criterion

Theorem 10.1 requires that $L(s)$ have no poles in the closed right half-plane. In some situations this is not the case and a more general result is required. Nyquist originally considered this general case, which we summarize as a theorem.

Theorem 10.2 (Nyquist's stability theorem)**.** *Consider a closed loop system with the loop transfer function $L(s)$ that has P poles in the region enclosed by the*



(a) Inverted pendulum (b) Nyquist plot

Figure 10.8: PD control of an inverted pendulum. (a) The system consists of a mass that is balanced by applying a force at the pivot point. A proportional-derivative controller with transfer function $C(s) = k(s+2)$ is used to command u based on θ. (b) A Nyquist plot of the loop transfer function for gain $k = 1$. There is one counterclockwise encirclement of the critical point, giving $N = -1$ clockwise encirclements.

Nyquist contour. Let N be the net number of clockwise encirclements of -1 by $L(s)$ when s encircles the Nyquist contour Γ in the clockwise direction. The closed loop system then has $Z = N + P$ poles in the right half-plane.

The full Nyquist criterion states that if $L(s)$ has P poles in the right half-plane, then the Nyquist curve for $L(s)$ should have P counterclockwise encirclements of -1 (so that $N = -P$). In particular, this *requires* that $|L(i\omega_c)| > 1$ for some ω_c corresponding to a crossing of the negative real axis. Care has to be taken to get the right sign of the encirclements. The Nyquist contour has to be traversed clockwise, which means that ω moves from $-\infty$ to ∞ and N is positive if the Nyquist curve winds clockwise. If the Nyquist curve winds counterclockwise, then N will be negative (the desired case if $P \neq 0$).

As in the case of the simplified Nyquist criterion, we use small semicircles of radius r to avoid any poles on the imaginary axis. By letting $r \to 0$, we can use Theorem 10.2 to reason about stability. Note that the image of the small semicircles generates a section of the Nyquist curve with large magnitude, requiring care in computing the winding number.

Example 10.6 Stabilized inverted pendulum

The linearized dynamics of a normalized inverted pendulum can be represented by the transfer function $P(s) = 1/(s^2 - 1)$, where the input is acceleration of the pivot and the output is the pendulum angle θ, as shown in Figure 10.8 (Exercise 9.3). We attempt to stabilize the pendulum with a proportional-derivative (PD) controller having the transfer function $C(s) = k(s+2)$. The loop transfer function is

$$L(s) = \frac{k(s+2)}{s^2 - 1}.$$

The Nyquist plot of the loop transfer function is shown in Figure 10.8b. We have $L(0) = -2k$ and $L(\infty) = 0$. If $k > 0.5$, the Nyquist curve encircles the critical point $s = -1$ in the counterclockwise direction when the Nyquist contour γ is encircled

in the clockwise direction. The number of encirclements is thus $N = -1$. Since the loop transfer function has one pole in the right half-plane ($P = 1$), we find that $Z = N + P = 0$ and the system is thus stable for $k > 0.5$. If $k < 0.5$, there is no encirclement and the closed loop will have one pole in the right half-plane. ∇

Derivation of Nyquist's Stability Theorem

We will now prove the Nyquist stability theorem for a general loop transfer function $L(s)$. This requires some results from the theory of complex variables, for which the reader can consult Ahlfors [Ahl66]. Since some precision is needed in stating Nyquist's criterion properly, we will use a more mathematical style of presentation. We also follow the mathematical convention of counting encirclements in the counterclockwise direction for the remainder of this section. The key result is the following theorem about functions of complex variables.

Theorem 10.3 (Principle of variation of the argument)**.** *Let D be a closed and bounded region in the complex plane and let Γ be the boundary of the region. Assume the function $f : \mathbb{C} \to \mathbb{C}$ is analytic in D and on Γ, except at a finite number of poles and zeros. Then the* winding number *w_n is given by*

$$w_n = \frac{1}{2\pi}\Delta_\Gamma \arg f(z) = \frac{1}{2\pi i}\int_\Gamma \frac{f'(z)}{f(z)}dz = Z - P,$$

where Δ_Γ is the net variation in the angle when z traverses the contour Γ in the counterclockwise direction, Z is the number of zeros in D and P is the number of poles in D. Poles and zeros of multiplicity m are counted m times.

Proof. Assume that $z = a$ is a zero of multiplicity m. In the neighborhood of $z = a$ we have

$$f(z) = (z-a)^m g(z),$$

where the function g is analytic and different from zero. The ratio of the derivative of f to itself is then given by

$$\frac{f'(z)}{f(z)} = \frac{m}{z-a} + \frac{g'(z)}{g(z)},$$

and the second term is analytic at $z = a$. The function f'/f thus has a single pole at $z = a$ with the residue m. The sum of the residues at the zeros of the function is Z. Similarly, we find that the sum of the residues for the poles is $-P$, and hence

$$Z - P = \frac{1}{2\pi i}\int_\Gamma \frac{f'(z)}{f(z)}\,dz = \frac{1}{2\pi i}\int_\Gamma \frac{d}{dz}\log f(z)\,dz = \frac{1}{2\pi i}\Delta_\Gamma \log f(z),$$

where Δ_Γ again denotes the variation along the contour Γ. We have

$$\log f(z) = \log|f(z)| + i\arg f(z),$$

and since the variation of $|f(z)|$ around a closed contour is zero it follows that

$$\Delta_\Gamma \log f(z) = i\Delta_\Gamma \arg f(z),$$

and the theorem is proved. □

This theorem is useful in determining the number of poles and zeros of a function of complex variables in a given region. By choosing an appropriate closed region D with boundary Γ, we can determine the difference between the number of poles and zeros through computation of the winding number.

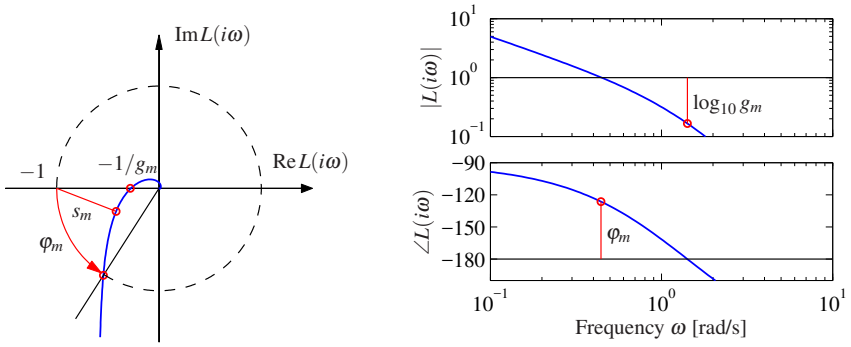
Theorem 10.3 can be used to prove Nyquist's stability theorem by choosing Γ as the Nyquist contour shown in Figure 10.3a, which encloses the right half-plane. To construct the contour, we start with part of the imaginary axis $-jR \leq s \leq jR$ and a semicircle to the right with radius R. If the function f has poles on the imaginary axis, we introduce small semicircles with radii r to the right of the poles as shown in the figure. The Nyquist contour is obtained by selecting R large enough and r small enough so that all opne-loop right half-plane poles are enclosed. Note that Γ has orientation *opposite* that shown in Figure 10.3a. (The convention in engineering is to traverse the Nyquist contour in the clockwise direction since this corresponds to moving upwards along the imaginary axis, which makes it easy to sketch the Nyquist contour from a Bode plot.)

To see how we use the principle of variation of the argument to compute stability, consider a closed loop system with the loop transfer function $L(s)$. The closed loop poles of the system are the zeros of the function $f(s) = 1 + L(s)$. To find the number of zeros in the right half-plane, we investigate the winding number of the function $f(s) = 1 + L(s)$ as s moves along the Nyquist contour Γ in the *counterclockwise* direction. The winding number can conveniently be determined from the Nyquist plot. A direct application of Theorem 10.3 gives the Nyquist criterion, taking care to flip the orientation. Since the image of $1 + L(s)$ is a shifted version of $L(s)$, we usually state the Nyquist criterion as net encirclements of the -1 point by the image of $L(s)$.

10.3 Stability Margins

In practice it is not enough that a system is stable. There must also be some margins of stability that describe how far from instability the system is and its robustness to perturbations. There are many ways to express this, but one of the most common is the use of gain and phase margins, inspired by Nyquist's stability criterion. The key idea is that it is easy to plot the loop transfer function $L(s)$. An increase in controller gain simply expands the Nyquist plot radially. An increase in the phase of the controller twists the Nyquist plot. Hence from the Nyquist plot we can easily pick off the amount of gain or phase that can be added without causing the system to become unstable.

Formally, the *gain margin* g_m of a closed-loop system is defined as the smallest amount that the loop gain can be increased before the system goes unstable. For a system whose phase decreases monotonically as a function of frequency starting at 0°, the gain margin can be computed based on the smallest frequency where the phase of the loop transfer function $L(s)$ is -180°. Let ω_{pc} represent this frequency,



(a) Nyquist plot (b) Bode plot

Figure 10.9: Stability margins. The gain margin g_m and phase margin φ_m are shown on the the Nyquist plot (a) and the Bode plot (b). The gain margin corresponds to the smallest increase in gain that creates an encirclement, and the phase margin is the smallest change in phase that creates an encirclement. The Nyquist plot also shows the stability margin s_m, which is the shortest distance to the critical point -1.

called the *phase crossover frequency*. Then the gain margin for the system is given by

$$g_m = \frac{1}{|L(i\omega_{\text{pc}})|}. \tag{10.5}$$

Similarly, the *phase margin* is the amount of phase lag required to reach the stability limit. Let ω_{gc} be the *gain crossover frequency*, the smallest frequency where the loop transfer function $L(s)$ has unit magnitude. Then for a system with monotonically decreasing gain, the phase margin is given by

$$\varphi_m = \pi + \arg L(i\omega_{\text{gc}}). \tag{10.6}$$

These margins have simple geometric interpretations in the Nyquist plot, as shown in Figure 10.9a, where we have plotted the portion of the curve corresponding to $\omega > 0$. The gain margin is given by the inverse of the distance to the nearest point between -1 and 0 where the loop transfer function crosses the negative real axis. The phase margin is given by the smallest angle on the unit circle between -1 and the loop transfer function. When the gain or phase is monotonic, this geometric interpretation agrees with the formulas above.

A drawback with gain and phase margins is that both have to be given to guarantee that the Nyquist curve is not close to the critical point. An alternative way to express margins is by a single number, the *stability margin* s_m, which is the shortest distance from the Nyquist curve to the critical point. The gain and phase margins can also be determined from the Bode plot as indicated in Figure 10.9.

Example 10.7 Third-order system

Consider a loop transfer function $L(s) = 3/(s+1)^3$. The Nyquist and Bode plots are shown in Figure 10.10. To compute the gain, phase and stability margins, we

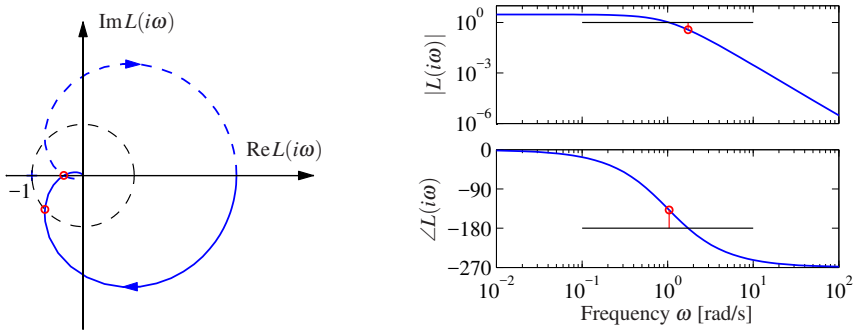


Figure 10.10: Stability margins for a third-order transfer function. The Nyquist plot on the left allows the gain, phase and stability margins to be determined by measuring the distances of relevant features. The gain and phase margins can also be read off of the Bode plot on the right.

can use the Nyquist plot shown in Figure 10.10. This yields the following values:

$$g_m = 2.67, \qquad \varphi_m = 41.7^\circ, \qquad s_m = 0.464.$$

The gain and phase margins can also be determined from the Bode plot. ∇

The gain and phase margins are classical robustness measures that have been used for a long time in control system design. The gain margin is well defined if the Nyquist curve intersects the negative real axis once. Analogously, the phase margin is well defined if the Nyquist curve intersects the unit circle at only one point. Other more general robustness measures will be introduced in Chapter 13.

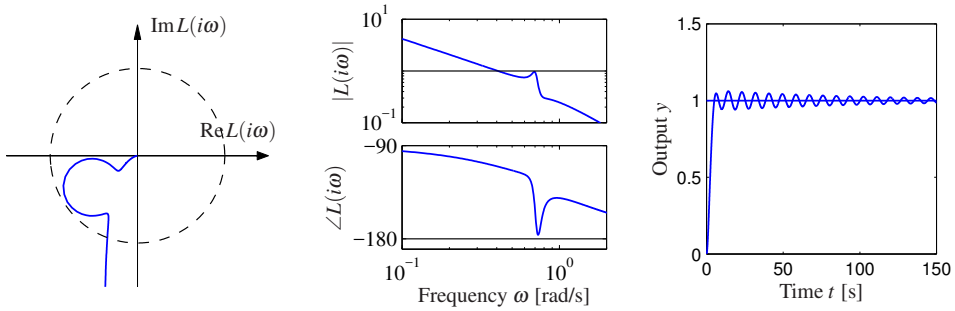
Even if both the gain and phase margins are reasonable, the system may still not be robust, as is illustrated by the following example.

Example 10.8 Good gain and phase margins but poor stability margins
Consider a system with the loop transfer function

$$L(s) = \frac{0.38(s^2 + 0.1s + 0.55)}{s(s+1)(s^2 + 0.06s + 0.5)}.$$

A numerical calculation gives the gain margin as $g_m = 266$, and the phase margin is 70°. These values indicate that the system is robust, but the Nyquist curve is still close to the critical point, as shown in Figure 10.11. The stability margin is $s_m = 0.27$, which is very low. The closed loop system has two resonant modes, one with damping ratio $\zeta = 0.81$ and the other with $\zeta = 0.014$. The step response of the system is highly oscillatory, as shown in Figure 10.11c. ∇

When designing feedback systems, it will often be useful to define the robustness of the system using gain, phase and stability margins. These numbers tell us how much the system can vary from our nominal model and still be stable. Reasonable values of the margins are phase margin $\varphi_m = 30^\circ$–60°, gain margin $g_m =$ 2–5 and stability margin $s_m = 0.5$–0.8.



(a) (b) (c)

Figure 10.11: System with good gain and phase margins but a poor stability margin. Nyquist (a) and Bode (b) plots of the loop transfer function and step response (c) for a system with good gain and phase margins but with a poor stability margin. The Nyquist plot shows on the portion of the curve corresponding to $\omega > 0$.

There are also other stability measures, such as the *delay margin*, which is the smallest time delay required to make the system unstable. For loop transfer functions that decay quickly, the delay margin is closely related to the phase margin, but for systems where the gain curve of the loop transfer function has several peaks at high frequencies, the delay margin is a more relevant measure.

Example 10.9 Nanopositioning system for an atomic force microscope
Consider the system for horizontal positioning of the sample in an atomic force microscope. The system has oscillatory dynamics, and a simple model is a spring–mass system with low damping. The normalized transfer function is given by

$$P(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \tag{10.7}$$

where the damping ratio typically is a very small number, e.g., $\zeta = 0.1$.

We will start with a controller that has only integral action. The resulting loop transfer function is

$$L(s) = \frac{k_i\omega_0^2}{s(s^2 + 2\zeta\omega_0 s + \omega_0^2)},$$

where k_i is the gain of the controller. Nyquist and Bode plots of the loop transfer function are shown in Figure 10.12. Notice that the part of the Nyquist curve that is close to the critical point -1 is approximately circular.

From the Bode plot in Figure 10.12b, we see that the phase crossover frequency is $\omega_{pc} = \omega_0$, which will be independent of the gain k_i. Evaluating the loop transfer function at this frequency, we have $L(i\omega_0) = -k_i/(2\zeta\omega_0)$, which means that the stability margin is $s_m = 1 - k_i/(2\zeta\omega_0)$. To have a desired stability margin of s_m the integral gain should be chosen as

$$k_i = 2\zeta\omega_0(1 - s_m).$$

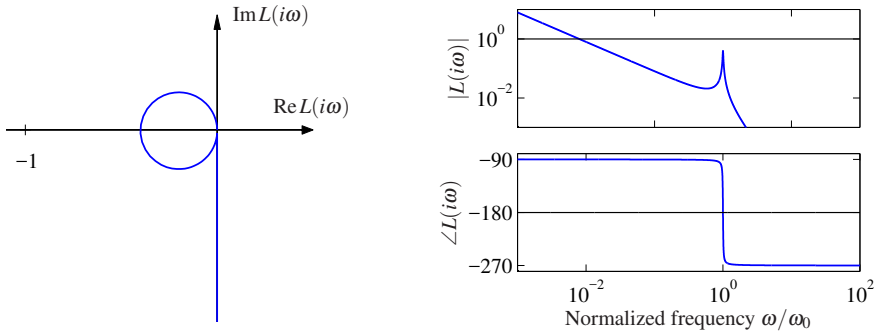


Figure 10.12: Nyquist and Bode plots of the loop transfer function for the AFM system (10.7) with an integral controller. The frequency in the Bode plot is normalized by ω_0. The parameters are $\zeta = 0.01$ and $k_i = 0.008$.

Figure 10.12 shows Nyquist and Bode plots for the system with gain margin $g_m = 2.5$ and stability margin $s_m = 0.6$. The gain curve in the Bode plot is almost a straight line for low frequencies and has a resonant peak at $\omega = \omega_0$. The gain crossover frequency is approximately equal to k_i and the phase decreases monotonically from -90° to -270°: it is equal to -180° at $\omega = \omega_0$. The curve can be shifted vertically by changing k_i: increasing k_i shifts the gain curve upward and increases the gain crossover frequency. ∇

10.4 Bode's Relations and Minimum Phase Systems

An analysis of Bode plots reveals that there appears to be a relation between the gain curve and the phase curve. Consider, for example, the Bode plots for the differentiator and the integrator (shown in Figure 9.12). For the differentiator the slope is $+1$ and the phase is a constant $\pi/2$ radians. For the integrator the slope is -1 and the phase is $-\pi/2$. For the first-order system $G(s) = s + a$, the amplitude curve has the slope 0 for small frequencies and the slope $+1$ for high frequencies, and the phase is 0 for low frequencies and $\pi/2$ for high frequencies.

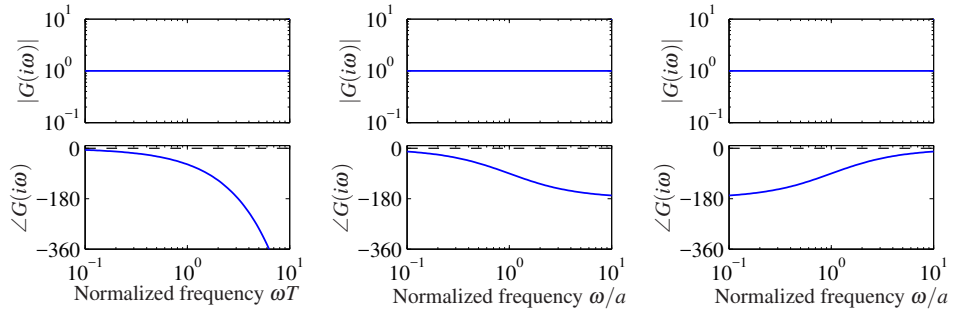
Bode investigated the relations between the curves for systems with no poles or zeros in the right half-plane. He found that, for systems that do not have poles or zeros in the right half plane, the phase was uniquely given by the shape of the gain curve, and vice versa:

$$\arg G(i\omega_0) = \frac{\pi}{2}\int_0^\infty f(\omega)\frac{d\log|G(i\omega)|}{d\log\omega}\,d\log\omega \approx \frac{\pi}{2}\frac{d\log|G(i\omega_0)|}{d\log\omega_0}, \qquad (10.8)$$

where f is the weighting kernel

$$f(\omega) = \frac{2}{\pi^2}\log\left|\frac{\omega+\omega_0}{\omega-\omega_0}\right|.$$

The phase curve is thus a weighted average of the derivative of the gain curve. If the gain curve has constant slope n, the phase curve has constant value $n\pi/2$.



(a) Time delay (b) RHP zero (c) RHP pole

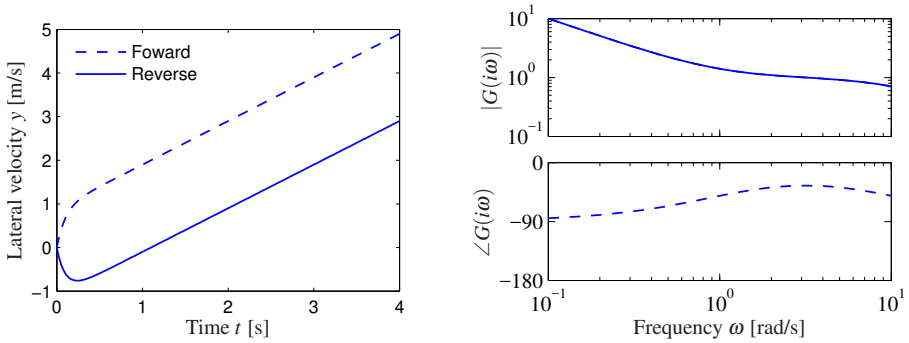
Figure 10.13: Bode plots of systems that are not minimum phase. (a) Time delay $G(s) = e^{-sT}$, (b) system with a right half-plane (RHP) zero $G(s) = (a-s)/(a+s)$ and (c) system with right half-plane pole. The corresponding minimum phase system has the transfer function $G(s) = 1$ in all cases, the phase curves for that system are shown as dashed lines.

Systems that do not have poles or zeros in the right half plane are called *minimum phase systems* because they have the smallest phase lag of all systems with the same gain curve. We will now give a few examples of nonminimum phase transfer functions.

The transfer function of a time delay of τ units is $G(s) = e^{-s\tau}$. This transfer function has unit gain $|G(i\omega)| = 1$, and the phase is $\arg G(i\omega) = -\omega\tau$. The corresponding minimum phase system with unit gain has the transfer function $G(s) = 1$. The time delay thus has an additional phase lag of $\omega\tau$. Notice that the phase lag increases linearly with frequency. Figure 10.13a shows the Bode plot of the transfer function. (Because we use a log scale for frequency, the phase falls off exponentially in the plot.)

Consider a system with the transfer function $G(s) = (a-s)/(a+s)$ with $a > 0$, which has a zero $s = a$ in the right half-plane. The transfer function has unit gain $|G(i\omega)| = 1$, and the phase is $\arg G(i\omega) = -2\arctan(\omega/a)$. The corresponding minimum phase system with unit gain has the transfer function $G(s) = 1$. Figure 10.13b shows the Bode plot of the transfer function. A similar analysis of the transfer function $G(s) = (s+a)/(s-a)$ with $a > 0$, which has a pole in the right half-plane, shows that its phase is $\arg G(i\omega) = -2\arctan(a/\omega)$. The Bode plot is shown in Figure 10.13c.

The presence of poles and zeros in the right half-plane imposes severe limitations on the achievable performance as will be discussed in Section 12.6. Dynamics of this type should be avoided by redesign of the system. While the poles are intrinsic properties of the system and they do not depend on sensors and actuators, the zeros depend on how inputs and outputs of a system are coupled to the states. Zeros can thus be changed by moving sensors and actuators or by introducing new sensors and actuators. Nonminimum phase systems are unfortunately quite common in practice.



(a) Step response (b) Frequency response

Figure 10.14: Vehicle steering for driving in reverse. (a) Step responses from steering angle to lateral translation for a simple kinematics model when driving forward (dashed) and reverse (solid). With rear-wheel steering the center of mass first moves in the wrong direction and that the overall response with rear-wheel steering is significantly delayed compared with that for front-wheel steering. (b) Frequency response for driving forward (dashed) and reverse (solid). Notice that the gain curves are identical, but the phase curve for driving in reverse has nonminimum phase.

The following example gives a system theoretic interpretation of the common experience that it is more difficult to drive in reverse gear and illustrates some of the properties of transfer functions in terms of their poles and zeros.

Example 10.10 Vehicle steering

The unnormalized transfer function from steering angle to lateral velocity for the simple vehicle model is

$$G(s) = \frac{av_0 s + v_0^2}{bs},$$

where v_0 is the velocity of the vehicle and $a, b > 0$ (see Example 6.12). The transfer function has a zero at $s = v_0/a$. In normal driving this zero is in the left half-plane, but it is in the right half-plane when driving in reverse, $v_0 < 0$. The unit step response is

$$y(t) = \frac{av_0}{b} + \frac{v_0^2 t}{b}.$$

The lateral velocity thus responds immediately to a steering command. For reverse steering v_0 is negative and the initial response is in the wrong direction, a behavior that is representative for nonminimum phase systems (called an *inverse response*).

Figure 10.14 shows the step response for forward and reverse driving. In this simulation we have added an extra pole with the time constant T to approximately account for the dynamics in the steering system. The parameters are $a = b = 1$, $T = 0.1$, $v_0 = 1$ for forward driving and $v_0 = -1$ for reverse driving. Notice that for $t > t_0 = a/v_0$, where t_0 is the time required to drive the distance a, the step response for reverse driving is that of forward driving with the time delay t_0. The position of the zero v_0/a depends on the location of the sensor. In our calculation

we have assumed that the sensor is at the center of mass. The zero in the transfer function disappears if the sensor is located at the rear wheel. The difficulty with zeros in the right half-plane can thus be visualized by a thought experiment where we drive a car in forward and reverse and observe the lateral position through a hole in the floor of the car. ∇

10.5 Generalized Notions of Gain and Phase

A key idea in frequency domain analysis is to trace the behavior of sinusoidal signals through a system. The concepts of gain and phase represented by the transfer function are strongly intuitive because they describe amplitude and phase relations between input and output. In this section we will see how to extend the concepts of gain and phase to more general systems, including some nonlinear systems. We will also show that there are analogs of Nyquist's stability criterion if signals are approximately sinusoidal.

System Gain

We begin by considering the case of a static linear system $y = Au$, where A is a matrix whose elements are complex numbers. The matrix does not have to be square. Let the inputs and outputs be vectors whose elements are complex numbers and use the Euclidean norm

$$\|u\| = \sqrt{\Sigma |u_i|^2}. \tag{10.9}$$

The norm of the output is

$$\|y\|^2 = u^* A^* A u,$$

where $*$ denotes the complex conjugate transpose. The matrix A^*A is symmetric and positive semidefinite, and the right-hand side is a quadratic form. The square root of eigenvalues of the matrix A^*A are all real, and we have

$$\|y\|^2 \leq \bar{\lambda}(A^*A)\|u\|^2,$$

where $\bar{\lambda}$ denotes the largest eigenvalue. The gain of the system can then be defined as the maximum ratio of the output to the input over all possible inputs:

$$\gamma = \max_u \frac{\|y\|}{\|u\|} = \sqrt{\bar{\lambda}(A^*A)}. \tag{10.10}$$

The square root of the eigenvalues of the matrix A^*A are called the *singular values* of the matrix A, and the largest singular value is denoted by $\bar{\sigma}(A)$.

To generalize this to the case of an input/output dynamical system, we need to think of the inputs and outputs not as vectors of real numbers but as vectors of *signals*. For simplicity, consider first the case of scalar signals and let the signal space L_2 be square-integrable functions with the norm

$$\|u\|_2 = \sqrt{\int_0^\infty |u|^2(\tau)\, d\tau}.$$

This definition can be generalized to vector signals by replacing the absolute value with the vector norm (10.9). We can now formally define the gain of a system taking inputs $u \in L_2$ and producing outputs $y \in L_2$ as

$$\gamma = \sup_{u \in L_2} \frac{\|y\|}{\|u\|}, \tag{10.11}$$

where sup is the *supremum,* defined as the smallest number that is larger or equal to its argument. The reason for using the supremum is that the maximum may not be defined for $u \in L_2$. This definition of the system gain is quite general and can even be used for some classes of nonlinear systems, though one needs to be careful about how initial conditions and global nonlinearities are handled.

The norm (10.11) has some nice properties in the case of linear systems. In particular, given a single-input, single-output stable linear system with transfer function $G(s)$, it can be shown that the norm of the system is given by

$$\gamma = \sup_{\omega} |G(i\omega)| =: \|G\|_\infty. \tag{10.12}$$

In other words, the gain of the system corresponds to the peak value of the frequency response. This corresponds to our intuition that an input produces the largest output when we are at the resonant frequencies of the system. $\|G\|_\infty$ is called the *infinity norm* of the transfer function $G(s)$.

This notion of gain can be generalized to the multi-input, multi-output case as well. For a linear multivariable system with a transfer function matrix $G(s)$ we can define the gain as

$$\gamma = \|G\|_\infty = \sup_{\omega} \bar{\sigma}(G(i\omega)). \tag{10.13}$$

Thus we can combine the idea of the gain of a matrix with the idea of the gain of a linear system by looking at the maximum singular value over all frequencies.

Small Gain and Passivity

For linear systems it follows from Nyquist's theorem that the closed loop is stable if the gain of the loop transfer function is less than 1 for all frequencies. This result can be extended to a larger class of systems by using the concept of the system gain defined in equation (10.11).

Theorem 10.4 (Small gain theorem)**.** *Consider the closed loop system shown in Figure 10.15, where H_1 and H_2 are input/output stable systems and the signal spaces are properly defined. Let the gains of the systems H_1 and H_2 be γ_1 and γ_2. Then the closed loop system is input/output stable if $\gamma_1\gamma_2 < 1$, and the gain of the closed loop system is*

$$\gamma = \frac{\gamma_1}{1 - \gamma_1\gamma_2}.$$

Notice that if systems H_1 and H_2 are linear, it follows from the Nyquist stability theorem that the closed loop is stable because if $\gamma_1\gamma_2 < 1$, the Nyquist curve is

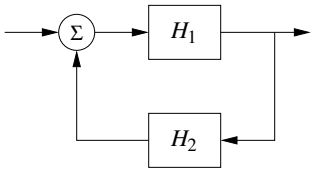


Figure 10.15: A feedback connection of two general nonlinear systems H_1 and H_2. The stability of the system can be explored using the small gain theorem.

always inside the unit circle. The small gain theorem is thus an extension of the Nyquist stability theorem.

Although we have focused on linear systems, the small gain theorem also holds for nonlinear input/output systems. The definition of gain in equation (10.11) holds for nonlinear systems as well, with some care needed in handling the initial condition.

The main limitation of the small gain theorem is that it does not consider the phasing of signals around the loop, so it can be very conservative. To define the notion of phase we require that there be a scalar product. For square-integrable functions this can be defined as

$$\langle u, y \rangle = \int_0^\infty u(\tau) y(\tau)\, d\tau.$$

The phase φ between two signals can now be defined as

$$\langle u, y \rangle = \|u\| \|y\| \cos(\varphi).$$

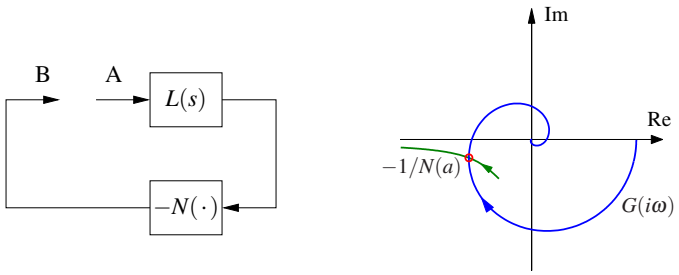
Systems where the phase between inputs and outputs is 90° or less for all inputs are called *passive systems*. It follows from the Nyquist stability theorem that a closed loop linear system is stable if the phase of the loop transfer function is between $-\pi$ and π. This result can be extended to nonlinear systems as well. It is called the *passivity theorem* and is closely related to the small gain theorem. See Khalil [Kha01] for a more detailed description.

Additional applications of the small gain theorem and its application to robust stability are given in Chapter 13.

Describing Functions

For special nonlinear systems like the one shown in Figure 10.16a, which consists of a feedback connection between a linear system and a static nonlinearity, it is possible to obtain a generalization of Nyquist's stability criterion based on the idea of *describing functions*. Following the approach of the Nyquist stability condition, we will investigate the conditions for maintaining an oscillation in the system. If the linear subsystem has low-pass character, its output is approximately sinusoidal even if its input is highly irregular. The condition for oscillation can then be found by exploring the propagation of a sinusoid that corresponds to the first harmonic.

To carry out this analysis, we have to analyze how a sinusoidal signal propagates through a static nonlinear system. In particular we investigate how the first



(a) Block diagram (b) Nyquist plot

Figure 10.16: Describing function analysis. A feedback connection between a static nonlinearity and a linear system is shown in (a). The linear system is characterized by its transfer function $L(s)$, which depends on frequency, and the nonlinearity by its describing function $N(a)$, which depends on the amplitude a of its input. The Nyquist plot of $L(i\omega)$ and the plot of the $-1/N(a)$ are shown in (b). The intersection of the curves represents a possible limit cycle.

harmonic of the output of the nonlinearity is related to its (sinusoidal) input. Letting F represent the nonlinear function, we expand $F(e^{i\omega t})$ in terms of its harmonics:

$$F(ae^{i\omega t}) = \sum_{n=0}^{\infty} M_n(a)e^{i(n\omega t+\varphi_n(a))},$$

where $M_n(a)$ and $\varphi_n(a)$ represent the gain and phase of the nth harmonic, which depend on the input amplitude since the function F is nonlinear. We define the describing function to be the complex gain of the first harmonic:

$$N(a) = M_1(a)e^{i\varphi_1(a)}. \tag{10.14}$$

The function can also be computed by assuming that the input is a sinusoid and using the first term in the Fourier series of the resulting output.

Neglecting higher harmonics and arguing as we did when deriving Nyquist's stability criterion, we find that an oscillation can be maintained if

$$L(i\omega)N(a) = -1. \tag{10.15}$$

This equation means that if we inject a sinusoid of amplitude a at A in Figure 10.16, the same signal will appear at B and an oscillation can be maintained by connecting the points. Equation (10.15) gives two conditions for finding the frequency ω of the oscillation and its amplitude a: the phase must be 180°, and the magnitude must be unity. A convenient way to solve the equation is to plot $L(i\omega)$ and $-1/N(a)$ on the same diagram as shown in Figure 10.16b. The diagram is similar to the Nyquist plot where the critical point -1 is replaced by the curve $-1/N(a)$ and a ranges from 0 to ∞.

It is possible to define describing functions for types of inputs other than sinusoids. Describing function analysis is a simple method, but it is approximate because it assumes that higher harmonics can be neglected. Excellent treatments

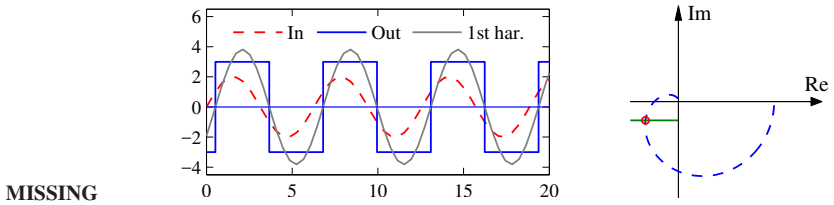


Figure 10.17: Describing function analysis for a relay with hysteresis. The input/output relation of the hysteresis is shown in (a) and the input with amplitude $a = 2$, the output and its first harmonic are shown in (b). The Nyquist plots of the transfer function $L(s) = (s+1)^{-4}$ and the negative of the inverse describing function for the relay with $b = 3$ and $c = 1$ are shown in (c).

of describing function techniques can be found in the texts by Atherton [Ath75] and Graham and McRuer [GM61].

Example 10.11 Relay with hysteresis
Consider a linear system with a nonlinearity consisting of a relay with hysteresis. The output has amplitude b and the relay switches when the input is $\pm c$, as shown in Figure 10.17a. Assuming that the input is $u = a\sin(\omega t)$, we find that the output is zero if $a \leq c$, and if $a > c$, the output is a square wave with amplitude b that switches at times $\omega t = \arcsin(c/a) + n\pi$. The first harmonic is then $y(t) = (4b/\pi)\sin(\omega t - \alpha)$, where $\sin\alpha = c/a$. For $a > c$ the describing function and its inverse are

$$N(a) = \frac{4b}{a\pi}\left(\sqrt{1-\frac{c^2}{a^2}} - i\frac{c}{a}\right), \qquad \frac{1}{N(a)} = \frac{\pi\sqrt{a^2-c^2}}{4b} + i\frac{\pi c}{4b},$$

where the inverse is obtained after simple calculations. Figure 10.17b shows the response of the relay to a sinusoidal input with the first harmonic of the output shown as a dashed line. Describing function analysis is illustrated in Figure 10.17c, which shows the Nyquist plot of the transfer function $L(s) = 2/(s+1)^4$ (dashed line) and the negative inverse describing function of a relay with $b = 1$ and $c = 0.5$. The curves intersect for $a = 1$ and $\omega = 0.77$ rad/s, indicating the amplitude and frequency for a possible oscillation if the process and the relay are connected in a a feedback loop. ∇

10.6 Further Reading

Nyquist's original paper giving his now famous stability criterion was published in the *Bell Systems Technical Journal* in 1932 [Nyq32]. More accessible versions are found in the book [BK64], which also includes other interesting early papers on control. Nyquist's paper is also reprinted in an IEEE collection of seminal papers on control [Bas01]. Nyquist used $+1$ as the critical point, but Bode changed it to

-1, which is now the standard notation. Interesting perspectives on early developments are given by Black [Bla77], Bode [Bod60] and Bennett [Ben93]. Nyquist did a direct calculation based on his insight into the propagation of sinusoidal signals through systems; he did not use results from the theory of complex functions. The idea that a short proof can be given by using the principle of variation of the argument is presented in the delightful book by MacColl [Mac45]. Bode made extensive use of complex function theory in his book [Bod45], which laid the foundation for frequency response analysis where the notion of minimum phase was treated in detail. A good source for complex function theory is the classic by Ahlfors [Ahl66]. Frequency response analysis was a key element in the emergence of control theory as described in the early texts by James et al. [JNP47], Brown and Campbell [BC48] and Oldenburger [Old56], and it became one of the cornerstones of early control theory. Frequency response methods underwent a resurgence when robust control emerged in the 1980s, as will be discussed in Chapter 13.

Exercises

10.1 (Operational amplifier) Consider an op amp circuit with $Z_1 = Z_2$ that gives a closed loop system with nominally unit gain. Let the transfer function of the operational amplifier be

$$G(s) = \frac{ka_1a_2}{(s+a)(s+a_1)(s+a_2)},$$

where $a_1, a_2 \gg a$. Show that the condition for oscillation is $k < a_1 + a_2$ and compute the gain margin of the system. Hint: Assume $a = 0$.

10.2 (Atomic force microscope) The dynamics of the tapping mode of an atomic force microscope are dominated by the damping of the cantilever vibrations and the system that averages the vibrations. Modeling the cantilever as a spring–mass system with low damping, we find that the amplitude of the vibrations decays as $\exp(-\zeta\omega t)$, where ζ is the damping ratio and ω is the undamped natural frequency of the cantilever. The cantilever dynamics can thus be modeled by the transfer function

$$G(s) = \frac{a}{s+a},$$

where $a = \zeta\omega_0$. The averaging process can be modeled by the input/output relation

$$y(t) = \frac{1}{\tau}\int_{t-\tau}^{t} u(v)dv,$$

where the averaging time is a multiple n of the period of the oscillation $2\pi/\omega$. The dynamics of the piezo scanner can be neglected in the first approximation because they are typically much faster than a. A simple model for the complete system is thus given by the transfer function

$$P(s) = \frac{a(1-e^{-s\tau})}{s\tau(s+a)}.$$

Plot the Nyquist curve of the system and determine the gain of a proportional controller that brings the system to the boundary of stability.

10.3 (Heat conduction) A simple model for heat conduction in a solid is given by the transfer function

$$P(s) = ke^{-\sqrt{s}}.$$

Sketch the Nyquist plot of the system. Determine the frequency where the phase of the process is -180° and the gain at that frequency. Show that the gain required to bring the system to the stability boundary is $k = e^{\pi}$.

10.4 (Vectored thrust aircraft) Consider the state space controller designed for the vectored thrust aircraft in Examples 7.8 and 8.6. The controller consists of two components: an optimal estimator to compute the state of the system from the output and a state feedback compensator that computes the input given the (estimated) state. Compute the loop transfer function for the system and determine the gain, phase and stability margins for the closed loop dynamics.

10.5 (Vehicle steering) Consider the linearized model for vehicle steering with a controller based on state feedback discussed in Example 8.4. The transfer functions for the process and controller are given by

$$P(s) = \frac{\gamma s + 1}{s^2}, \quad C(s) = \frac{s(k_1 l_1 + k_2 l_2) + k_1 l_2}{s^2 + s(\gamma k_1 + k_2 + l_1) + k_1 + l_2 + k_2 l_1 - \gamma k_2 l_2},$$

as computed in Example 9.10. Let the process parameter be $\gamma = 0.5$ and assume that the state feedback gains are $k_1 = 1$ and $k_2 = 0.914$ and that the observer gains are $l_1 = 2.828$ and $l_2 = 4$. Compute the stability margins numerically.

10.6 (Stability margins for second-order systems) A process whose dynamics is described by a double integrator is controlled by an ideal PD controller with the transfer function $C(s) = k_d s + k_p$, where the gains are $k_d = 2\zeta\omega_0$ and $k_p = \omega_0^2$. Calculate and plot the gain, phase and stability margins as a function ζ.

10.7 (Congestion control in overload conditions) A strongly simplified flow model of a TCP loop under overload conditions is given by the loop transfer function

$$L(s) = \frac{k}{s} e^{-s\tau},$$

where the queuing dynamics are modeled by an integrator, the TCP window control is a time delay τ and the controller is simply a proportional controller. A major difficulty is that the time delay may change significantly during the operation of the system. Show that if we can measure the time delay, it is possible to choose a gain that gives a stability margin of $s_m \geq 0.6$ for all time delays τ.

10.8 (Bode's formula) Consider Bode's formula (10.8) for the relation between gain and phase for a transfer function that has all its singularities in the left half-plane. Plot the weighting function and make an assessment of the frequencies where the approximation $\arg G \approx (\pi/2) d\log|G|/d\log\omega$ is valid.

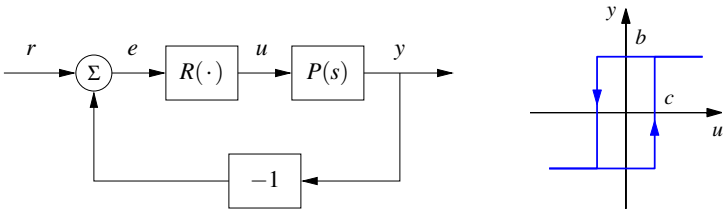
10.9 (Padé approximation to a time delay) Consider the transfer functions

$$G_1(s) = e^{-s\tau}, \qquad G_2(s) = e^{-s\tau} \approx \frac{1 - s\tau/2}{1 + s\tau/2}. \tag{10.16}$$

Show that the minimum phase properties of the transfer functions are similar for frequencies $\omega < 1/\tau$. A long time delay τ is thus equivalent to a small right half-plane zero. The approximation (10.16) is called a first-order *Padé approximation*.

10.10 (Inverse response) Consider a system whose input/output response is modeled by $G(s) = 6(-s+1)/(s^2+5s+6)$, which has a zero in the right half-plane. Compute the step response for the system, and show that the output goes in the wrong direction initially, which is also referred to as an *inverse response*. Compare the response to a minimum phase system by replacing the zero at $s = 1$ with a zero at $s = -1$.

10.11 (Describing function analysis) Consider the system with the block diagram shown on the left below.



The block R is a relay with hysteresis whose input/output response is shown on the right and the process transfer function is $P(s) = e^{-s\tau}/s$. Use describing function analysis to determine frequency and amplitude of possible limit cycles. Simulate the system and compare with the results of the describing function analysis.

Chapter Eleven
PID Control

Based on a survey of over eleven thousand controllers in the refining, chemicals and pulp and paper industries, 97% of regulatory controllers utilize PID feedback.

L. Desborough and R. Miller, 2002 [DM02a].

This chapter treats the basic properties of proportional-integral-derivative (PID) control and the methods for choosing the parameters of the controllers. We also analyze the effects of actuator saturation and time delay, two important features of many feedback systems, and describe methods for compensating for these effects. Finally, we will discuss the implementation of PID controllers as an example of how to implement feedback control systems using analog or digital computation.

11.1 Basic Control Functions

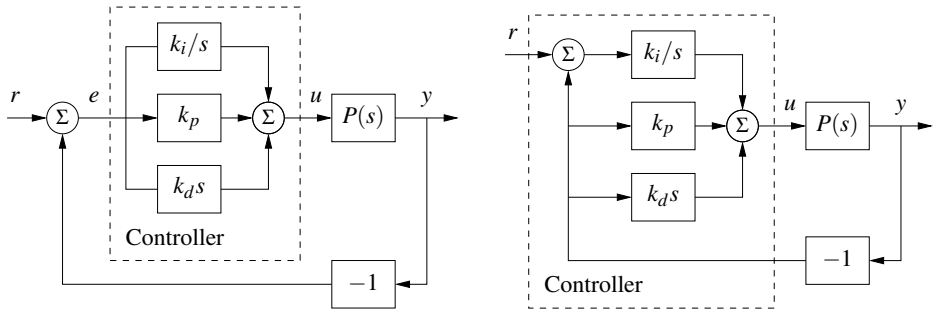
PID control, which was introduced in Section 1.6 and has been used in several examples, is by far the most common way of using feedback in engineering systems. It appears in simple devices and in large factories with thousands of controllers. PID controllers appear in many different forms: as stand-alone controllers, as part of hierarchical, distributed control systems and built into embedded components. Most PID controllers do not use derivative action, so they should strictly speaking be called PI controllers; we will, however, use PID as a generic term for this class of controller. There is also growing evidence that PID control appears in biological systems [YHSD00b].

Block diagrams of closed loop systems with PID controllers are shown in Figure 11.1. The control signal u for the system in Figure 11.1a is formed entirely from the error e; there is no feedforward term (which would correspond to $k_r r$ in the state feedback case). A common alternative in which proportional and derivative action do not act on the reference is shown in Figure 11.1b; combinations of the schemes will be discussed in Section 11.5. The command signal r is called the reference signal in regulation problems, or the *setpoint* in the literature of PID control. We generally assume that its value is in the same units as the output (otherwise a scaling factor may be required).

The input/output relation for an ideal PID controller with error feedback is

$$u = k_p e + k_i \int_0^t e(\tau)\,d\tau + k_d \frac{de}{dt} = k_p\Big(e + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau + T_d \frac{de}{dt}\Big). \qquad (11.1)$$

The control action is thus the sum of three terms: proportional feedback, the integral term and derivative action. For this reason PID controllers were originally



(a) PID using error feedback (b) PID using two degrees of freedom

Figure 11.1: Block diagrams of closed loop systems with ideal PID controllers. Both controllers have one output, the control signal u. The controller in (a), which is based on error feedback, has one input, the control error $e = r - y$. For this controller proportional, integral and derivative action acts on the error $e = r - y$. The two degree-of-freedom controller in (b) has two inputs, the reference r and the process output y. Integral action acts on the error, but proportional and derivative action act on the process output y.

called *three-term controllers*. The controller parameters are the proportional gain k_p, the integral gain k_i and the derivative gain k_d. The controller can also be parameterized with the time constants $T_i = k_p/k_i$ and $T_d = k_d/k_p$, called integral time (constant) and derivative time (constant).

The controller (11.1) represents an idealized controller. It is a useful abstraction for understanding the PID controller, but several modifications must be made to obtain a controller that is practically useful. Before discussing these practical issues we will develop some intuition about PID control.

We start by considering pure proportional feedback. Figure 11.2a shows the responses of the process output to a unit step in the reference value for a system with pure proportional control at different gain settings. In the absence of a feedforward term, the output never reaches the reference, and hence we are left with nonzero steady-state error. Letting the process transfer function be $P(s)$, with proportional feedback we have $C(s) = k_p$ and the transfer function from reference to error is

$$G_{er}(s) = \frac{1}{1 + C(s)P(s)} = \frac{1}{1 + k_p P(s)}. \tag{11.2}$$

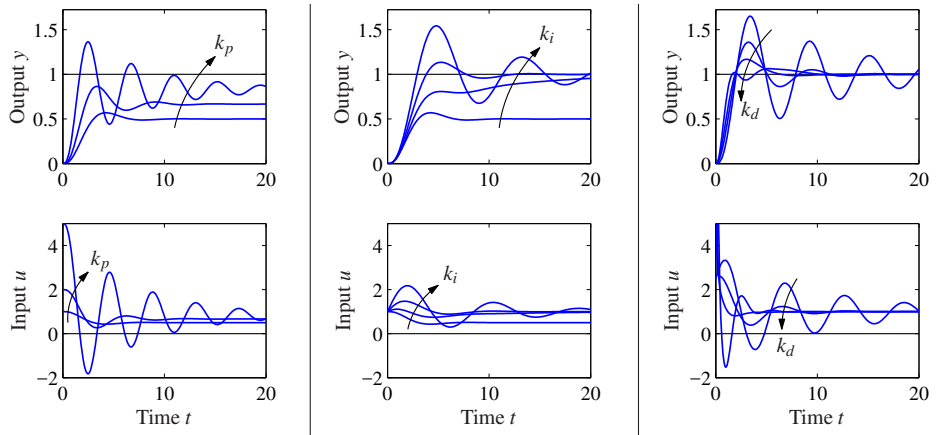
Assuming that the closed loop is stable, the steady-state error for a unit step is

$$G_{er}(0) = \frac{1}{1 + C(0)P(0)} = \frac{1}{1 + k_p P(0)}.$$

For the system in Figure 11.2a with gains $k_p = 1$, 2 and 5, the steady-state error is 0.5, 0.33 and 0.17. The error decreases with increasing gain, but the system also becomes more oscillatory. The system becomes unstable for $k_p = 8$. Notice in the figure that the initial value of the control signal equals the controller gain.

To avoid having a steady-state error, the proportional term can be changed to

$$u(t) = k_p e(t) + u_{\text{ff}}, \tag{11.3}$$



(a) Proportional control (b) PI control (c) PID control

Figure 11.2: Responses to step changes in the reference value for a system with a proportional controller (a), PI controller (b) and PID controller (c). The process has the transfer function $P(s) = 1/(s+1)^3$, the proportional controller has parameters $k_p = 1$, 2 and 5, the PI controller has parameters $k_p = 1$, $k_i = 0$, 0.2, 0.5 and 1, and the PID controller has parameters $k_p = 2.5$, $k_i = 1.5$ and $k_d = 0$, 1, 2 and 4.

where u_{ff} is a feedforward term that is adjusted to give the desired steady-state value. If the reference value r is constant and we choose $u_{\text{ff}} = r/P(0) = k_r r$, then the steady-state output will be exactly equal to the reference value, as it was in the state space case, provided that there are no disturbances. However, this requires exact knowledge of the process dynamics, which is usually not available. The parameter u_{ff}, called *reset* in the PID literature, must therefore be adjusted manually.

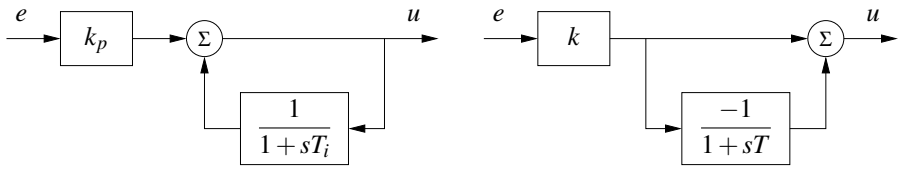
Another alternative to avoid a steady state error is to multiply the reference by $1 + k_p P(0)$, but this also requires precise knowledge of the steady state gain of the process.

As we saw in Section 7.4, integral action guarantees that the process output agrees with the reference in steady-state and provides an alternative to the feedforward term. Since this result is so important, we will provide a general proof. Consider the controller given by equation (11.1) with $k_i \neq 0$. Assume that $u(t)$ and $e(t)$ converge to steady-state values $u = u_0$ and $e = e_0$. It then follows from equation (11.1) that

$$u_0 = k_p e_0 + k_i \lim_{t\to\infty} \int_0^t e(t)dt.$$

The limit of the right hand side is not finite unless $e(t)$ goes to zero, which implies that $e_0 = 0$. We can thus conclude that with integral action the error will be zero if it reaches a steady state. Notice that we have not assumed that the process is linear or time invariant. We have, however, assumed that an equilibrium is reached. Analysis based on transfer functions require that the closed loop system is linear and time invariant.

Using integral action to achieve zero steady-state error is much better than us-



(a) Automatic reset (b) Derivative action

Figure 11.3: The block diagram in (a) shows how integral action is implemented using *positive feedback* with a first-order system, sometimes called automatic reset. The block diagram in (b) shows how derivative action can be implemented by taking differences between a static system and a first-order system.

ing feedforward, which requires a precise knowledge of process parameters.

The effect of integral action can also be understood from frequency domain analysis. The transfer function of the PID controller is

$$C(s) = k_p + \frac{k_i}{s} + k_d s. \tag{11.4}$$

The controller has infinite gain at zero frequency ($C(0) = \infty$), and it then follows from equation (11.2) that $G_{er}(0) = 0$, which implies that there is no steady-state error for a step input.

Integral action can also be viewed as a method for generating the feedforward term u_{ff} in the proportional controller (11.3) automatically. This is shown in Figure 11.3a, where the controller output is low-pass-filtered and fed back with positive gain. This implementation, called *automatic reset*, was one of the early inventions of integral control. The transfer function of the system in Figure 11.3a is obtained by block diagram algebra; we have

$$G_{ue} = k_p \frac{1+sT_i}{sT_i} = k_p + \frac{k_p}{sT_i},$$

which is the transfer function for a PI controller.

The properties of integral action are illustrated in Figure 11.2b for a step input. The proportional gain is constant, $k_p = 1$, and the integral gains are $k_i = 0$, 0.2, 0.5 and 1. The case $k_i = 0$ corresponds to pure proportional control, with a steady-state error of 50%. The steady-state error is eliminated when integral gain action is used. The response creeps slowly toward the reference for small values of k_i and goes faster for larger integral gains, but the system also becomes more oscillatory.

The integral gain k_i is a useful measure for attenuation of load disturbances. Consider a closed loop system under PID control, like the one in Figure 9.1. Assume that the system is stable and initially at rest with all signals being zero. Apply a unit step load disturbance at the process input. After a transient the process output goes to zero and the controller output settles at a value that compensates for the disturbance. Since $e(t)$ goes to zero as $t \to \infty$ it follows from equation (11.1) that

$$u(\infty) = k_i \int_0^\infty e(t)dt.$$

The integrated error $\int_0^\infty e(t)dt$ is thus inversely proportional to the integral gain k_i. The integral gain is thus a measure of the effectiveness of disturbance attenuation. A large gain k_i attenuates disturbances effectively, but too large a gain gives oscillatory behavior, poor robustness and possibly instability.

We now return to the general PID controller and consider the effect of the derivative term k_d. Recall that the original motivation for derivative feedback was to provide predictive or anticipatory action. Notice that the combination of the proportional and the derivative terms can be written as

$$u = k_p e + k_d \frac{de}{dt} = k_p\big(e + T_d \frac{de}{dt}\big) = k_p e_p,$$

where $e_p(t)$ can be interpreted as a prediction of the error at time $t + T_d$ by linear extrapolation. The prediction time $T_d = k_d/k_p$ is the *derivative time constant* of the controller.

Derivative action can be implemented by taking the difference between the signal and its low-pass filtered version as shown in Figure 11.3b. The transfer function for the system is

$$G_{ue}(s) = k\Big(1 - \frac{1}{1+sT}\Big) = k\frac{sT}{1+sT} = \frac{kTs}{1+sT}. \tag{11.5}$$

The transfer function $G_{ue}(s)$ approximates a derivative for low frequencies, because for $|s| \ll 1/T$ we have $G(s) \approx kTs$. The transfer function G_{ue} acts like a differentiator for signals with frequencies low frequencies and as a constant gain k for high frequency signals.

Figure 11.2c illustrates the effect of derivative action: the system is oscillatory when no derivative action is used, and it becomes more damped as the derivative gain is increased. Performance deteriorates if the derivative gain is too high. When the input is a step, the controller output generated by the derivative term will be an impulse. This is clearly visible in Figure 11.2c. The impulse can be avoided by using the controller configuration shown in Figure 11.1b.

Another way to generate the filtered derivative is to filter the measured signal y with a second order filter with the transfer function

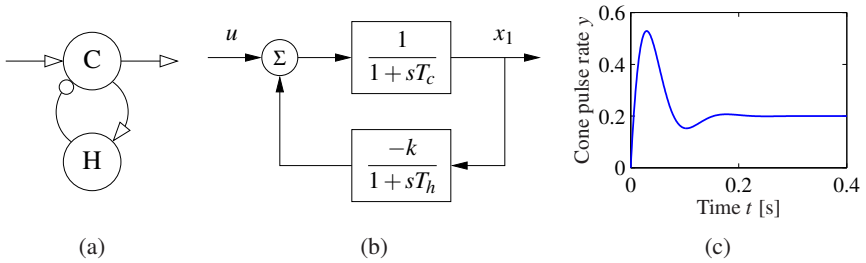
$$G_f(s) = \frac{1}{1 + sT_f + (sT_f)^2/2},$$

which has the realization

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -2/T_f & -2/T_f \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2/T_f \end{pmatrix} y, \tag{11.6}$$

where the states are $x_1 = y_f$ and $x_2 = T_f \frac{dy_f}{dt}$. The filter thus gives filtered versions of the measured signal and its derivative. The second-order filter also provides good high-frequency roll-off, which improves robustness.

Although PID control was developed in the context of engineering applications, it also appears in nature. Disturbance attenuation by feedback in biological systems is often called *adaptation*. A typical example is the pupillary reflex discussed in



(a) (b) (c)

Figure 11.4: Schematic diagram of cone photoreceptors (C) and horizontal cells (H) in the retina. In the schematic diagram in (a), excitatory feedback is indicated by arrows and inhibitory feedback by circles. A block diagram is shown in (b) and the step response in (c).

Example 9.14, where it is said that the eye adapts to changing light intensity. Analogously, feedback with integral action is called perfect adaptation [YHSD00b]. In biological systems proportional, integral and derivative action is generated by combining subsystems with dynamical behavior similarly to what is done in engineering systems. For example, PI action can be generated by the interaction of several hormones [ESGK02].

Example 11.1 PD action in the retina
The response of cone photoreceptors in the retina is an example where proportional and derivative action is generated by a combination of cones and horizontal cells. The cones are the primary receptors stimulated by light, which in turn stimulate the horizontal cells, and the horizontal cells give inhibitory (negative) feedback to the cones. A schematic diagram of the system is shown in Figure 11.4a. The system can be modeled by ordinary differential equations by representing neuron signals as continuous variables representing the average pulse rate. In [Wil99] it is shown that the system can be represented by the differential equations

$$\frac{dx_1}{dt} = \frac{1}{T_c}(-x_1 - kx_2 + u), \qquad \frac{dx_2}{dt} = \frac{1}{T_h}(x_1 - x_2),$$

where u is the light intensity and x_1 and x_2 are the average pulse rates from the cones and the horizontal cells. A block diagram of the system is shown in Figure 11.4b. The step response of the system shown in Figure 11.4c shows that the system has a large initial response followed by a lower, constant steady-state response typical of proportional and derivative action. The parameters used in the simulation are $k = 4$, $T_c = 0.025$ and $T_h = 0.08$. ∇

11.2 Simple Controllers for Complex Systems

Many of the design methods discussed in previous chapters have the property that the complexity of the controller is a direct reflection of the complexity of the model. When designing controllers by output feedback in Chapter 8, we found for single-input, single-output systems that the order of the controller was the same as

the order of the model, possibly one order higher if integral action was required. Applying these design methods to PID control requires that the models must be of first or second order.

Low-order models can be obtained from first principles. Any stable system can be modeled by a static system if its inputs are sufficiently slow. Similarly a first-order model is sufficient if the storage of mass, momentum or energy can be captured by only one variable; typical examples are the velocity of a car on a road, angular velocity of a stiff rotational system, the level in a tank and the concentration in a volume with good mixing. System dynamics are of second order if the storage of mass, energy and momentum can be captured by two state variables; typical examples are the position and velocity of a car on the road, the orientation and angular velocity of satellites, the levels in two connected tanks and the concentrations in two-compartment models. A wide range of techniques for model reduction are also available. In this chapter we will focus on design techniques where we simplify the models to capture the essential properties that are needed for PID design.

We begin by analyzing the case of integral control. A stable system can be controlled by an integral controller provided that the requirements on the closed loop system are modest. To design the controller we assume that the transfer function of the process is a constant $K = P(0)$. The loop transfer function under integral control then becomes Kk_i/s, and the closed loop characteristic polynomial is simply $s + Kk_i$. Specifying performance by the desired time constant T_{cl} of the closed loop system, we find that the integral gain is given by

$$k_i = \frac{1}{T_{cl}P(0)}.$$

The analysis requires that T_{cl} be sufficiently large that the process transfer function can be approximated by a constant.

For systems that are not well represented by a constant gain, we can obtain a better approximation by using the Taylor series expansion of the loop transfer function:

$$L(s) = \frac{k_i P(s)}{s} \approx \frac{k_i(P(0) + sP'(0))}{s} = k_i P'(0) + \frac{k_i P(0)}{s}.$$

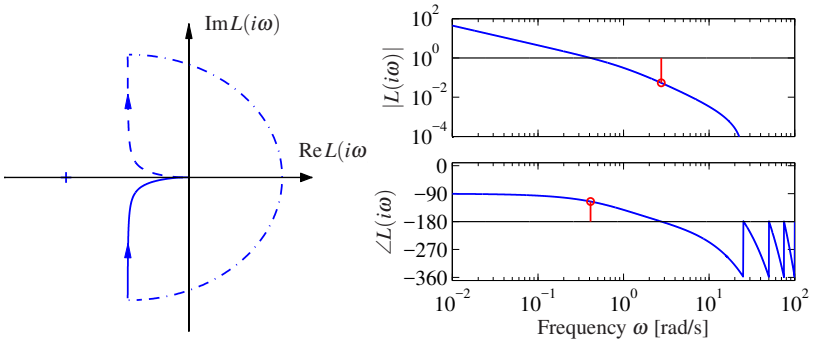
Choosing $k_i P'(0) = -0.5$ gives a system with good robustness, as will be discussed in Section 13.5. The controller gain is then given by

$$k_i = -\frac{1}{2P'(0)}, \tag{11.7}$$

and the expected closed loop time constant is $T_{cl} \approx -P'(0)/P(0)$.

Example 11.2 Integral control of AFM in tapping mode

A simplified model of the dynamics of the vertical motion of an atomic force microscope in tapping mode was discussed in Exercise 10.2. The transfer function



(a) Nyquist plot (b) Bode plot

Figure 11.5: Integral control for AFM in tapping mode. An integral controller is designed based on the slope of the process transfer function at 0. The controller gives good robustness properties based on a very simple analysis.

for the system dynamics is

$$P(s) = \frac{a(1-e^{-s\tau})}{s\tau(s+a)},$$

where $a = \zeta\omega_0$, $\tau = 2\pi n/\omega_0$ and the gain has been normalized to 1. We have $P(0) = 1$ and $P'(0) = -\tau/2 - 1/a$, and it follows from (11.7) that the integral gain can be chosen as $k_i = a/(2+a\tau)$. Nyquist and Bode plots for the resulting loop transfer function are shown in Figure 11.5. ∇

A first-order system has the transfer function

$$P(s) = \frac{b}{s+a}.$$

With a PI controller the closed loop system has the characteristic polynomial

$$s(s+a) + bk_p s + bk_i = s^2 + (a+bk_p)s + bk_i.$$

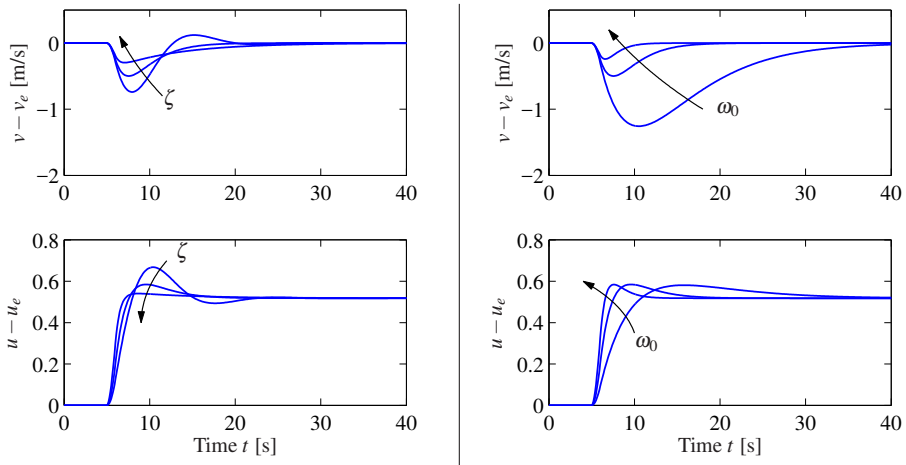
The closed loop poles can thus be assigned arbitrary values by proper choice of the controller gains k_p and k_i. Requiring that the closed loop system have the characteristic polynomial

$$p(s) = s^2 + a_1 s + a_2,$$

we find that the controller parameters are

$$k_p = \frac{a_1 - a}{b}, \qquad k_i = \frac{a_2}{b}. \tag{11.8}$$

If we require a response of the closed loop system that is slower than that of the open loop system, a reasonable choice is $a_1 = a + \alpha$ and $a_2 = \alpha a$, where $\alpha < a$ determines the closed loop response. If a response faster than that of the open loop system is required, a possible choice is $a_1 = 2\zeta\omega_0$ and $a_2 = \omega_0^2$, where ω_0 and ζ are undamped natural frequency and damping ratio of the dominant mode.



(a) $\omega_0 = 0.5$, $\zeta = 0.5, 1, 2$ (b) $\zeta = 1$, $\omega_0 = 0.2, 0.5, 1$

Figure 11.6: Cruise control using PI feedback. The step responses for the error and input illustrate the effect of parameters $\zeta = 1$ and ω_0 on the response of a car with cruise control. The slope of the road changes linearly from 0° to 4° between $t = 5$ and 6 s. (a) Responses for $\omega_0 = 0.5$ and $\zeta = 0.5$, 1 and 2. Choosing $\zeta = 1$ gives no overshoot in the velocity v. (b) Responses for $\zeta = 1$ and $\omega_0 = 0.2$, 0.5 and 1.0.

The choice of ω_0 has a significant impact on the robustness of the system and will be discussed in Section 13.4. An upper limit to ω_0 is given by the validity of the model. Large values of ω_0 will require fast control actions, and actuators may saturate if the value is too large. A first-order model is unlikely to represent the true dynamics for high frequencies.

Example 11.3 Cruise control using PI feedback

Consider the problem of maintaining the speed of a car as it goes up a hill. In Example 6.14 we found that there was little difference between the linear and nonlinear models when investigating PI control, provided that the throttle did not reach the saturation limits. A simple linear model of a car was given in Example 6.11:

$$\frac{d(v-v_e)}{dt} = -a(v-v_e) + b(u-u_e) - g\theta, \tag{11.9}$$

where v is the velocity of the car, u is the input to the engine (throttle) and θ is the slope of the hill. The parameters were $a = 0.0101$, $b = 1.3203$, $g = 9.8$, $v_e = 20$ and $u_e = 0.1616$. This model will be used to find suitable parameters of a vehicle speed controller. The transfer function from throttle to velocity is a first-order system. Since the open loop dynamics are so slow ($1/a \approx 100$ s), it is natural to specify a faster closed loop system by requiring that the closed loop system be of second order with damping ratio ζ and undamped natural frequency ω_0. The controller gains are given by equation (11.8).

Figure 11.6 shows the velocity and the throttle for a car that initially moves on a horizontal road and encounters a hill with a slope of 4° at time $t = 6$ s. To

design a PI controller we choose $\zeta = 1$ to obtain a response without overshoot, as shown in Figure 11.6a. The choice of ω_0 is a compromise between response speed and control actions: a large value gives a fast response, but it requires fast control action. The trade-off is illustrated in Figure 11.6b. The largest velocity error decreases with increasing ω_0, but the control signal also changes more rapidly. In the simple model (11.9) it was assumed that the force responds instantaneously to throttle commands. For rapid changes there may be additional dynamics that have to be accounted for. There are also physical limitations to the rate of change of the force, which also restricts the admissible value of ω_0. A reasonable choice of ω_0 is in the range 0.5–1.0. Notice in Figure 11.6 that even with $\omega_0 = 0.2$ the largest velocity error is only 1 m/s. ∇

A PI controller can also be used for a process with second-order dynamics, but there will be restrictions on the possible locations of the closed loop poles. Using a PID controller, it is possible to control a system of second order in such a way that the closed loop poles have arbitrary locations; see Exercise 11.2.

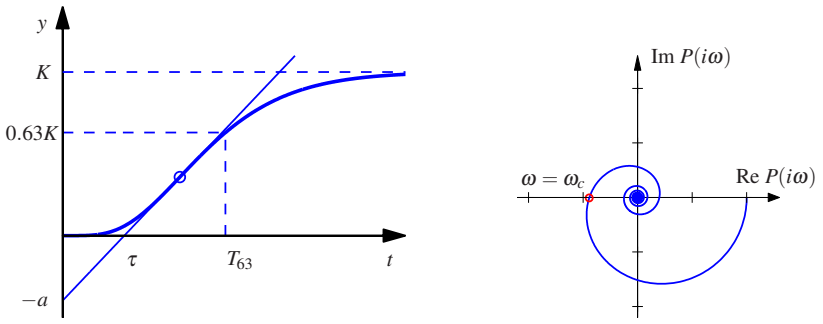
Instead of finding a low-order model and designing controllers for them, we can also use a high-order model and attempt to place only a few dominant poles. An integral controller has one parameter, and it is possible to position one pole. Consider a process with the transfer function $P(s)$. The loop transfer function with an integral controller is $L(s) = k_i P(s)/s$. The roots of the closed loop characteristic polynomial are the roots of $s + k_i P(s) = 0$. Requiring that $s = -a$ be a root, the controller gain should be chosen as

$$k_i = \frac{a}{P(-a)}. \tag{11.10}$$

The pole $s = -a$ will be a dominant closed loop pole if a is smaller than the magnitude of the open loop process poles. A similar approach can be applied to PI and PID controllers; see Exercise 11.3.

11.3 PID Tuning

Users of control systems are frequently faced with the task of adjusting the controller parameters to obtain a desired behavior. There are many different ways to do this. One approach is to go through the conventional steps of modeling and control design as described in the previous section. Since the PID controller has so few parameters, a number of special empirical methods have also been developed for direct adjustment of the controller parameters. The first tuning rules were developed by Ziegler and Nichols [ZN42]. Their idea was to perform a simple experiment, extract some features of process dynamics from the experiment and determine the controller parameters from the features.



(a) Step response method (b) Frequency response method

Figure 11.7: Ziegler–Nichols step and frequency response experiments. The open loop unit step response in (a) is characterized by the parameters a and τ. The frequency response method (b) characterizes process dynamics by the point where the Nyquist curve of the process transfer function first intersects the negative real axis and the frequency ω_c where this occurs.

Ziegler–Nichols' Tuning

In the 1940s, Ziegler and Nichols developed two methods for controller tuning based on simple characterization of process dynamics in the time and frequency domains.

The time domain method is based on a measurement of part of the open loop unit step response of the process, as shown in Figure 11.7a. The step response is measured by a *bump test*. The process is first brought to steady-state, the input is then changed by a suitable amount, the output is measured and scaled to correspond to a unit step input. Newton and Ziegler characterized the step response by only two parameters a and τ, which are the intercepts of the steepest tangent of the step response with the coordinate axes. The parameter τ is an approximation of the time delay of the system and a/τ is the steepest slope of the step response. Notice that it is not necessary to wait until steady state is reached to be able to determine the parameters; it suffices to wait until the response has had an inflection point. The suggested controller parameters are given in Table 11.1. They were obtained by extensive simulation of a range of representative processes. A controller was tuned manually for each process, and an attempt was then made to correlate the controller parameters with a and τ.

In the frequency domain method, a controller is connected to the process, the integral and derivative gains are set to zero and the proportional gain is increased until the system starts to oscillate. The critical value k_c of the proportional gain is observed together with the period of oscillation T_c. It follows from Nyquist's stability criterion that the Nyquist contour for the loop transfer function $L = k_c P(s)$ passes through the critical point at the frequency $\omega_c = 2\pi/T_c$. The experiment thus gives the point on the Nyquist curve of the process transfer function $P(s)$ where the phase lag is 180°, as shown in Figure 11.7b. The suggested controller parameters are then given by Table 11.1b.

Table 11.1: Ziegler–Nichols tuning rules. (a) The step response methods give the parameters in terms of the intercept a and the apparent time delay τ. (b) The frequency response method gives controller parameters in terms of *critical gain* k_c and *critical period* T_c.

Type	k_p	T_i	T_d
P	1/a		
PI	0.9/a	3τ	
PID	1.2/a	2τ	0.5τ

(a) Step response method

Type	k_p	T_i	T_d
P	$0.5k_c$		
PI	$0.4k_c$	$0.8T_c$	
PID	$0.6k_c$	$0.5T_c$	$0.125T_c$

(b) Frequency response method

The Ziegler–Nichols methods had a huge impact when they were introduced in the 1940s. The rules were simple to use and gave initial conditions for manual tuning. The ideas were adopted by manufacturers of controllers for routine use. The Ziegler–Nichols tuning rules unfortunately have two severe drawbacks: too little process information is used, and the closed loop systems that are obtained lack robustness.

The step response method can be improved significantly by characterizing the unit step response by the FOTD (First Order and Time Delay) model

$$P(s) = \frac{K}{1+sT} e^{-\tau s}. \tag{11.11}$$

This model is commonly used to approximate the step response of systems with monotone step responses. The parameters of the model are be obtained from a bump test as indicated in Figure 11.7a. The zero frequency gain K is the steady-state value of the unit step response. The time delay τ is the intercept of the steepest tangent with the time axis, as in the Ziegler-Nichols method. The time T_{63} is the time where the output has reached 63% of its steady-state value and T is then given by $T = T_{63} - \tau$. Notice that it takes longer time to find an FOTD model than the Ziegler-Nichols model (a and τ) because to determine K it is necessary to wait until the steady state has been reached.

The frequency response method can be improved by measuring more points on the Nyquist curve of $P(s)$, e.g., the zero frequency gain K or the point where the process has a 90° phase lag. This latter point can be obtained by connecting an integral controller and increasing its gain until the system reaches the stability limit. The experiment can also be automated by using relay feedback, as will be discussed later in this section.

There are many versions of improved tuning rules for the model (11.11). As an

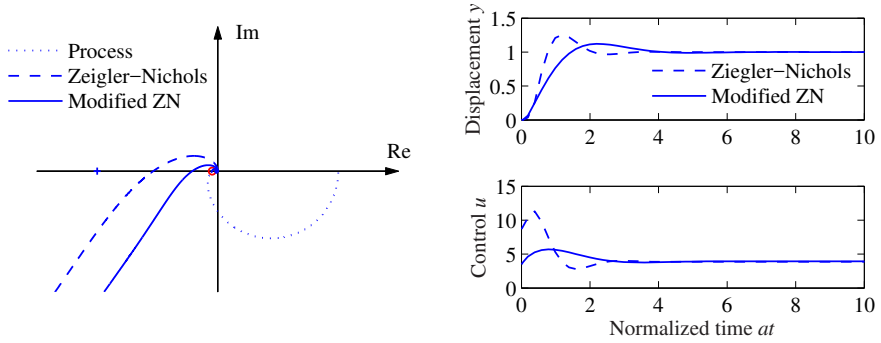


Figure 11.8: PI control of an AFM in tapping mode. Nyquist plots (a) and step responses (b) for PI control of the vertical motion of an atomic force microscope in tapping mode. Results with Ziegler–Nichols tuning are shown by dashed lines, and modified Ziegler–Nichols tuning is shown by solid lines. The Nyquist plot of the process transfer function is shown by dotted lines.

illustration we give the following rules for PI control, based on [ÅH05]:

$$k_p = \frac{0.15\tau + 0.35T}{K\tau} \quad \left(\frac{0.9T}{K\tau}\right), \qquad k_i = \frac{0.46\tau + 0.02T}{K\tau^2} \quad \left(\frac{0.3T}{K\tau^2}\right), \tag{11.12a}$$

$$k_p = 0.22k_c - \frac{0.07}{K} \quad \left(0.4k_c\right), \qquad k_i = \frac{0.16k_c}{T_c} + \frac{0.62}{KT_c} \quad \left(\frac{0.5k_c}{T_c}\right). \tag{11.12b}$$

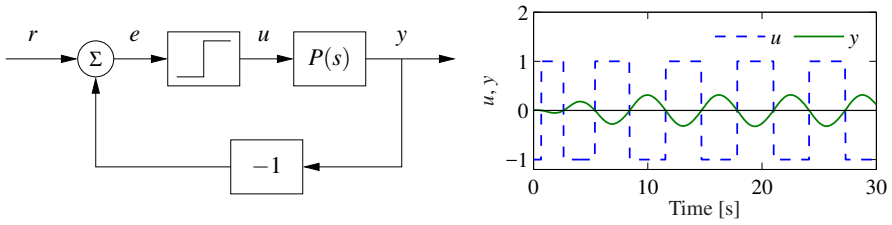
The values for the Ziegler–Nichols rule from Table 11.1 are given in parentheses. Notice that the improved formulas typically give lower controller gains than the Ziegler–Nichols method. The integral gain is higher for systems where the dynamics are delay-dominated, $\tau \gg T$.

Example 11.4 Atomic force microscope in tapping mode
A simplified model of the dynamics of the vertical motion of an atomic force microscope in tapping mode was discussed in Example 11.2. The transfer function is normalized by choosing $1/a$ as the time unit, yielding

$$P(s) = \frac{1 - e^{-sT_n}}{sT_n(s+1)},$$

where $T_n = 2n\pi a/\omega_0 = 2n\pi\zeta$. The Nyquist plot of $P(s)$ is shown as a dotted line in Figure 11.8a for $\zeta = 0.002$ and $n = 20$. The first intersection with the real axis occurs at $\mathrm{Re}\, s = -0.0461$ for $\omega_c = 13.1$. The critical gain is thus $k_c = 21.7$ and the critical period is $T_c = 0.48$. Using the Ziegler–Nichols tuning rule, we find the parameters $k_p = 8.87$ and $k_i = 22.6$ ($T_i = 0.384$) for a PI controller. With this controller the stability margin is $s_m = 0.31$, which is quite small. The step response of the controller is shown using dashed lines in Figure 11.8. Notice in particular that there is a large overshoot in the control signal.

The modified Ziegler–Nichols rule (11.12b) gives the controller parameters



(a) Relay feedback (b) Oscillatory response

Figure 11.9: Block diagram of a process with relay feedback (a) and typical signals (b). The process output y is a solid line, and the relay output u is a dashed line. Notice that the signals u and y have opposite phases.

$k_p = 3.47$ and $k_i = 8.73$ ($T_i = 0.459$) and the stability margin becomes $s_m = 0.61$. The step response with this controller is shown in Figure 11.8. A comparison of the responses obtained with the original Ziegler–Nichols rule shows that the overshoot has been reduced. Notice that the control signal reaches its steady-state value almost instantaneously. It follows from Example 11.2 that a pure integral controller has the normalized gain $k_i = 1/(2+T_n) = 0.44$ which is more than an order of magnitude smaller than the integral gain of the PI controller.. ∇

Relay Feedback

The Ziegler–Nichols frequency response method increases the gain of a proportional controller until oscillation to determine the critical gain k_c and the corresponding critical period T_c or, equivalently, the point where the Nyquist curve intersects the negative real axis. One way to obtain this information automatically is to connect the process in a feedback loop with a nonlinear element having a relay function as shown in Figure 11.9a. For many systems there will then be an oscillation, as shown in Figure 11.9b, where the relay output u is a square wave and the process output y is close to a sinusoid. Moreover, the fundamental sinusoidal components of the input and the output are 180° out of phase, which means that the system oscillates with the critical period T_c. Notice that an oscillation with constant period is established quickly.

To determine the critical gain k_c we expand the square wave relay output in a Fourier series. Notice in the figure that the process output is practically sinusoidal because the process attenuates higher harmonics. It is then sufficient to consider only the first harmonic component of the input. Letting d be the relay amplitude, this component has amplitude $4d/\pi$. If a is the amplitude of the process output, the process gain at the critical frequency $\omega_c = 2\pi/T_c$ is $|P(i\omega_c)| = \pi a/(4d)$ and the critical gain is

$$k_c = \frac{4d}{a\pi}. \tag{11.13}$$

Having obtained the critical gain k_c and the critical period T_c, the controller parameters can then be determined using the Ziegler–Nichols rules. Improved tuning

can be obtained by fitting a model to the data obtained from the relay experiment.

The relay experiment can be automated. Since the amplitude of the oscillation is proportional to the relay output, it is easy to control it by adjusting the relay output. *Automatic tuning* based on relay feedback is used in many commercial PID controllers. Tuning is accomplished simply by pushing a button that activates relay feedback. The relay amplitude is automatically adjusted to keep the oscillations sufficiently small, and the relay feedback is replaced by a PID controller when the tuning is finished.

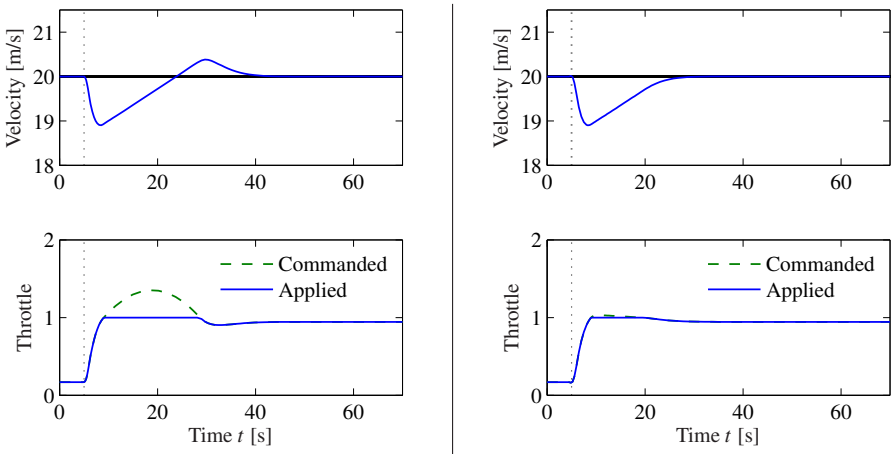
11.4 Integrator Windup

Many aspects of a control system can be understood from linear models. There are, however, some nonlinear phenomena that must be taken into account. These are typically limitations in the actuators: a motor has limited speed, a valve cannot be more than fully opened or fully closed, etc. For a system that operates over a wide range of conditions, it may happen that the control variable reaches the actuator limits. When this happens, the feedback loop is broken and the system runs in open loop because the actuator remains at its limit independently of the process output as long as the actuator remains saturated. The integral term will also build up since the error is typically nonzero. The integral term and the controller output may then become very large. The control signal will then remain saturated even when the error changes, and it may take a long time before the integrator and the controller output come inside the saturation range. The consequence is that there are large transients. This situation is referred to as *integrator windup*, illustrated in the following example.

Example 11.5 Cruise control

The windup effect is illustrated in Figure 11.10a, which shows what happens when a car encounters a hill that is so steep (6°) that the throttle saturates when the cruise controller attempts to maintain speed. When encountering the slope at time $t = 5$, the velocity decreases and the throttle increases to generate more torque. However, the torque required is so large that the throttle saturates. The error decreases slowly because the torque generated by the engine is just a little larger than the torque required to compensate for gravity. The error is large and the integral continues to build up until the error reaches zero at time 30, but the controller output is still larger than the saturation limit and the actuator remains saturated. The integral term starts to decrease, and the velocity settles to the desired value at time $t = 40$. Also notice the large overshoot. ∇

There are many methods to avoid windup. One method is illustrated in Figure 11.11: the system has an extra feedback path that is generated by a mathematical model of the saturating actuator. The signal e_s is the difference between the outputs of the controller v and the actuator model u is fed to the input of the integrator through gain k_t. The signal e_s is zero when there is no saturation and the extra feedback loop has no effect on the system. When the actuator saturates, the



(a) Windup (b) Anti-windup

Figure 11.10: Simulation of PI cruise control with windup (a) and anti-windup (b). The figure shows the speed v and the throttle u for a car that encounters a slope that is so steep that the throttle saturates. The controller output is a dashed line. The controller parameters are $k_p = 0.5$ and $k_i = 0.1$. The anti-windup compensator eliminates the overshoot by preventing the error for building up in the integral term of the controller.

signal e_s is fed back to the integrator in such a way that e_s goes toward zero. This implies that controller output is kept close to the saturation limit. The controller output will then change as soon as the error changes sign and integral windup is avoided.

The rate at which the controller output is reset is governed by the feedback gain k_t; a large value of k_t gives a short reset time. The parameter k_t cannot be too large because measurement noise can then cause an undesirable reset. A reasonable choice is to choose k_t as a multiple of $1/T_i = k_i/k_p$.

Notice that the controller also has an input u_{ff} for a feedforward signal. By entering the feedforward signal as shown in Figure 11.11 the anti-windup scheme also deals with saturations caused by the feedforward signal.

There are situations where decoupling has severe drawbacks because of the basic physics of the system. A typical example is flight control, where turns are executed very effectively by first rolling. The large forces on the wings can then be used to turn the aircraft. If the system is decoupled in yaw and roll a turn must be performed by using the rudder which gives much smaller forces than when the wings are used. For flight control lateral control is thus performed by treating yaw and roll as a coupled system.

The anti-windup scheme in Figure 11.11 assumes that the actuator output is available to the controller, when this is not the case we will instead use a model of the actuator as will be done in Section 11.5.

We illustrate how integral windup can be avoided by investigating the cruise control system.

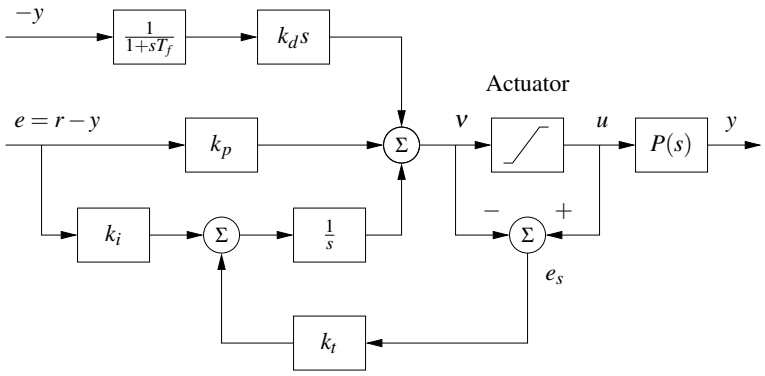


Figure 11.11: PID controller with anti-windup. The controller has an input u_{ff} for feedforward signals and filtering of the measured signal. The input to the integrator $(1/s)$ consists of the error term plus a "reset" based on input saturation. If the actuator is not saturated, then $e_s = u - v$, otherwise e_s will decrease the integrator input to prevent windup.

Example 11.6 Cruise control with anti-windup

Figure 11.10b shows what happens when a controller with anti-windup is applied to the system simulated in Figure 11.10a. Because of the feedback from the actuator model, the output of the integrator is quickly reset to a value such that the controller output is at the saturation limit. The behavior is drastically different from that in Figure 11.10a and the large overshoot is avoided. The tracking gain used in the simulation is $k_t = 2$ which is an order of magnitude larger than $1/T_i = k_i/k_p = 0.2$. ∇

Manual Control

Automatic control is often combined with manual control. This can be accomplished by providing the controller with a switch from manual to automatic. Manual control is typically actuated using buttons for increasing and decreasing the control signal. The control signal increases at constant rate when pushing the increase button and it decreases at constant rate when the decrease button is pushed. There are more sophisticated schemes where the rate increases when the switch is pushed a longer time, like the search mechanism in an iPod.

Care has to be taken to avoid transients when switching modes. This can be accomplished by the arrangement shown in Figure 11.12 where the integrator for the PI controller is used both for manual and automatic control. In automatic mode (A) the switch S connects the error signal to the input of the integrator. In manual mode (M) the switch disconnects the error signal and connects the integral to the increase-decrease buttons in manual mode. The switching transients will be small since the output of the integrator is the same in manual and automatic mode. There proportional term will create a switching transient if the error e is not zero when the mode is changed.

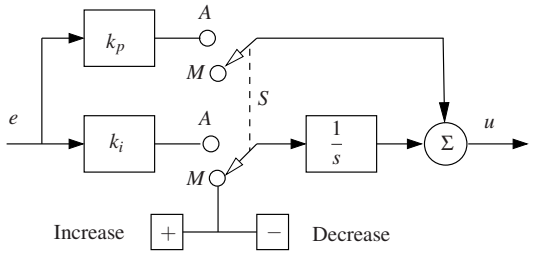


Figure 11.12: Block diagram of a PI controller that can switch between manual and automatic control. Automatic mode is indicated by *A* and manual mode by *M*. Manual control is executed by the buttons marked + and − which increase or decrease the control signal.

Controller with a Tracking Mode

When continuous control is combined with logic it is important that the controller state is handled properly when switching modes. In the controller with manual and automatic control in Figure 11.12 the problem was solved by using the same integrator for manual and automatic control. This ensures that there will be no switching transients. A general approach is to introduce a *tracking mode* in the controller.

A block diagram of a controller with a tracking mode is shown in Figure 11.13. The controller has modes for automatic control and tracking, selected by a switch. In the automatic control mode (A) the controller works as a normal controller which generates the control signal u from the reference r and the process output y. The controller has two degrees of freedom and is governed by the differential equation

$$a(p)u(t) = b(p)r(t) - c(p)y(t). \tag{11.14}$$

In tracking mode (T) the output u of the controller is equal to the tracking signal w and the behavior is governed by the differential equation

$$a_o(p)\,z(t) = b(p)\,r(t) - c(p)\,y(t) + \big(a_o(p) - a(p)\big)\,u(t), \tag{11.15}$$

where the polynomial $a_o(s)$ is stable, monic, and of the same degree as $a(s)$. If the

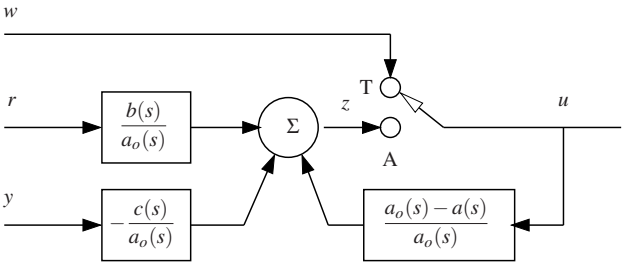


Figure 11.13: Block diagram of a controller with a tracking mode. The controller has three continuous input signals: reference r, process output y and tracking signal w, and a switch for selecting automatic A control or tracking T.

controller has integral action we have

$$a(0) = 0, \qquad b(0) = c(0), \qquad a_o(0) \neq 0.$$

In steady state where all signals are constant we have $w = u$. The controller can then be switched to automatic mode without transients. The polynomial $a_o(s)$ determines how quickly the state is reached. By continuity we have $w(t) \approx u(t)$ for slowly varying signals.

When the controller is switched to automatic we set $z = u$ and equation (11.15) becomes

$$a_o(p)\,u(t) = b(p)\,r(t) - c(p)\,y(t) + \big(a_o(p) - a(p)\big)\,u(t),$$

which is identical to (11.14). We will illustrate with an example.

Example 11.7 PI controller with tracking mode
A PI controller with two degrees of freedom has the input/output relation

$$u(t) = k_p\big(\beta r(t) - y(t)\big) + k_i \int_0^t \big(r(\tau) - y(\tau)\big)\,d\tau,$$

which is equivalent to the differential equation

$$\frac{du}{dt} = \left(k_p\beta\frac{dr}{dt} + k_i r\right) - \left(k_p\frac{dy}{dt} + k_i y\right).$$

In tracking mode the controller is described by

$$\frac{dz}{dt} + a_o z = \left(k_p\beta\frac{dr}{dt} + r\right) - \left(k_p\frac{dy}{dt} + k_i y\right) + a_o u$$

(see Figure 11.13). A convenient way to combine automatic and tracking mode is to represent the controller as

$$\frac{dz}{dt} = \left(k_p\beta\frac{dr}{dt} + r\right) - \left(k_p\frac{dy}{dt} + k_i y\right) + a_o(u - z)$$
$$u = \begin{cases} z & \text{automatic mode} \\ w & \text{tracking mode} \end{cases}.$$

In this representation the controller is a first-order system with one state z whose dynamics has the characteristic polynomial $s + a_o$. In automatic mode the controller output is equal to the state z. In the tracking mode the output is equal to the external tracking signal w and the controller state z tracks the signal w. ∇

11.5 Implementation

There are many practical issues that have to be considered when implementing PID controllers. They have been developed over time based on practical experience. In this section we consider some of the most common. Similar considerations also apply to other types of controllers.

Filtering the Derivative

A drawback with derivative action is that an ideal derivative has high gain for high-frequency signals. This means that high-frequency measurement noise will generate large variations in the control signal. The effect of measurement noise may be reduced by replacing the term $k_d s$ by $k_d s/(1+sT_f)$, which can be interpreted as an ideal derivative of a low-pass filtered signal. The time constant of the filter is typically chosen as $T_f = (k_d/k_p)/N = T_d/N$, with N in the range 5–20. Filtering is obtained automatically if the derivative is implemented by taking the difference between the signal and its filtered version as shown in Figure 11.3b; see also equation (11.5). Note that in the implementation in Figure 11.3b, the filter time constant T_f is restricted to be equal to the derivative time constant T_d ($N = 1$).

Instead of filtering just the derivative, it is also possible to use an ideal controller and filter the measured signal. Choosing a second order Butterworth filter the transfer function the controller with the filter becomes

$$C(s) = k_p\left(1+\frac{1}{sT_i}+sT_d\right)\frac{1}{1+sT_f+(sT_f)^2/2}. \tag{11.16}$$

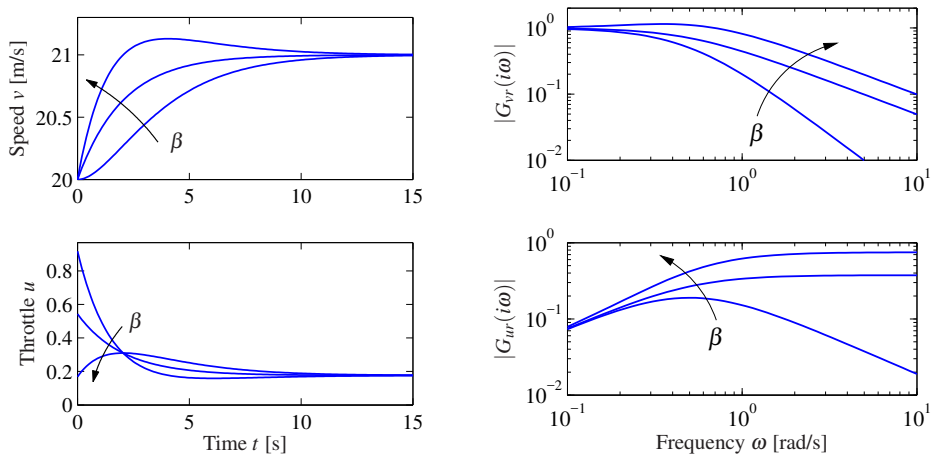
Setpoint Weighting

Figure 11.1 shows two configurations of a PID controller. The system in Figure 11.1a has a controller with *error feedback* where proportional, integral and derivative action acts on the error. In the simulation of PID controllers in Figure 11.2c there is a large initial peak in the control signal, which is caused by the derivative of the reference signal. The peak can be avoided by using the controller in Figure 11.1b, where proportional and derivative action acts only on the process output. An intermediate form is given by

$$u = k_p\big(\beta r - y\big) + k_i\int_0^t \big(r(\tau)-y(\tau)\big)\,d\tau + k_d\Big(\gamma\frac{dr}{dt}-\frac{dy}{dt}\Big), \tag{11.17}$$

where the proportional and derivative actions act on fractions β and γ of the reference. Integral action has to act on the error to make sure that the error goes to zero in steady state. The closed loop systems obtained for different values of β and γ respond to load disturbances and measurement noise in the same way. The response to reference signals is different because it depends on the values of β and γ, which are called *reference weights* or *setpoint weights*. We illustrate the effect of setpoint weighting by an example.

Example 11.8 Cruise control with setpoint weighting
Consider the PI controller for the cruise control system derived in Example 11.3. Figure 11.14 shows the effect of setpoint weighting on the response of the system to a reference signal. With $\beta = 1$ (error feedback) there is an overshoot in velocity and the control signal (throttle) is initially close to the saturation limit. There is no overshoot with $\beta = 0$ and the control signal is much smaller, clearly a much better drive comfort. The frequency responses gives another view of the same effect. The



(a) Step response (b) Frequency responses

Figure 11.14: Time and frequency responses for PI cruise control with setpoint weighting. Step responses are shown in (a), and the gain curves of the frequency responses in (b). The controller gains are $k_p = 0.74$ and $k_i = 0.19$. The setpoint weights are $\beta = 0$, 0.5 and 1, and $\gamma = 0$.

parameter β is typically in the range 0–1, and γ is normally zero to avoid large transients in the control signal when the reference is changed. ∇

The controller given by equation (11.17) is a special case of the general controller structure having two degrees of freedom, which was discussed in Section 8.5.

Implementation Based on Operational Amplifiers

PID controllers have been implemented in different technologies. Figure 11.15 shows how PI and PID controllers can be implemented by feedback around operational amplifiers.

To show that the circuit in Figure 11.15b is a PID controller we will use the approximate relation between the input voltage e and the output voltage u of the operational amplifier derived in Example 9.3,

$$u = -\frac{Z_2}{Z_1} e.$$

In this equation Z_1 is the impedance between the negative input of the amplifier and the input voltage e, and Z_2 is the impedance between the zero input of the amplifier and the output voltage u. The impedances are given by

$$Z_1(s) = \frac{R_1}{1 + R_1 C_1 s}, \qquad Z_2(s) = R_2 + \frac{1}{C_2 s},$$

and we find the following relation between the input voltage e and the output volt-

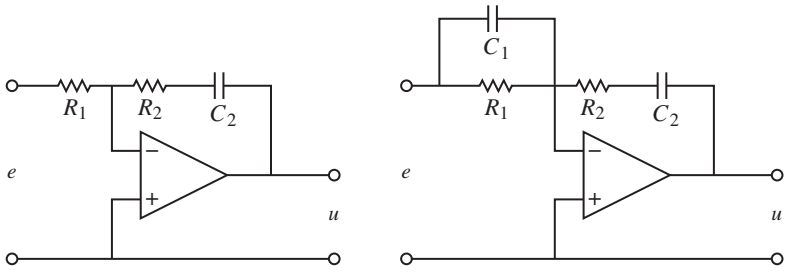


Figure 11.15: Schematic diagrams for PI and PID controllers using op amps. The circuit in (a) uses a capacitor in the feedback path to store the integral of the error. The circuit in (b) adds a filter on the input to provide derivative action.

age u:

$$u = -\frac{Z_2}{Z_1}e = -\frac{R_2}{R_1}\frac{(1+R_1C_1s)(1+R_2C_2s)}{R_2C_2s}e.$$

This is the input/output relation for a PID controller of the form (11.1) with parameters

$$k_p = \frac{R_1C_1+R_2C_2}{R_1C_2}, \qquad T_i = R_1C_1+R_2C_2, \qquad T_d = \frac{R_1R_2C_1C_2}{R_1C_1+R_2C_2}.$$

The corresponding results for a PI controller are obtained by setting $C_1 = 0$ (removing the capacitor).

Computer Implementation

In this section we briefly describe how a PID controller may be implemented using a computer. The computer typically operates periodically, with signals from the sensors sampled and converted to digital form by the A/D converter, and the control signal computed and then converted to analog form for the actuators. The sequence of operation is as follows:

1. Wait for clock interrupt
2. Read input from sensor
3. Compute control output
4. Send output to the actuator
5. Update controller state
6. Repeat

Notice that an output is sent to the actuators as soon as it is available. The time delay is minimized by making the calculations in step 3 as short as possible and performing all updates after the output is commanded. This simple way of reducing the latency is, unfortunately, seldom used in commercial systems.

As an illustration we consider the PID controller in Figure 11.11, which has a filtered derivative, setpoint weighting and protection against integral windup (anti-windup). The controller is a continuous-time dynamical system. To implement it using a computer, the continuous-time system has to be approximated by a discrete-time system.

In Figure 11.11, the signal v is the sum of the proportional, integral and derivative terms, and the controller output is $u = \text{sat}(v)$, where sat is the saturation function that models the actuator. The proportional term $P = k_p(\beta r - y)$ is implemented simply by replacing the continuous variables with their sampled versions. Hence

$$P(t_k) = k_p\left(\beta r(t_k) - y(t_k)\right), \tag{11.18}$$

where $\{t_k\}$ denotes the sampling instants, i.e., the times when the computer reads its input. We let h represent the sampling time, so that $t_{k+1} = t_k + h$. The integral term is obtained by approximating the integral with a sum,

$$I(t_{k+1}) = I(t_k) + k_i h\, e(t_k) + \frac{h}{T_t}\big(\text{sat}(v) - v\big), \tag{11.19}$$

where $T_t = h/k_t$ represents the anti-windup term. The filtered derivative term D is given by the differential equation

$$T_f \frac{dD}{dt} + D = -k_d \dot{y}.$$

Approximating the derivative with a backward difference gives

$$T_f \frac{D(t_k) - D(t_{k-1})}{h} + D(t_k) = -k_d \frac{y(t_k) - y(t_{k-1})}{h},$$

which can be rewritten as

$$D(t_k) = \frac{T_f}{T_f + h} D(t_{k-1}) - \frac{k_d}{T_f + h}\left(y(t_k) - y(t_{k-1})\right). \tag{11.20}$$

The advantage of using a backward difference is that the parameter $T_f/(T_f + h)$ is nonnegative and less than 1 for all $h > 0$, which guarantees that the difference equation is stable. Reorganizing equations (11.18)–(11.20), the PID controller can be described by the following pseudocode:

```
% Precompute controller coefficients
bi=ki*h
ad=Tf/(Tf+h)
bd=kd/(Tf+h)
br=h/Tt

% Control algorithm - main loop
while (running) {
  r=adin(ch1)               % read setpoint from ch1
  y=adin(ch2)               % read process variable from ch2
  P=kp*(b*r-y)              % compute proportional part
  D=ad*D-bd*(y-yold)        % compute derivative part
  v=P+I+D                   % compute temporary output
  u=sat(v,ulow,uhigh)       % simulate actuator saturation
  daout(ch1)                % set analog output ch1
  I=I+bi*(r-y)+br*(u-v)     % update integral state
  yold=y                    % update derivative state
  sleep(h)                  % wait until next update interval
}
```

Precomputation of the coefficients `bi`, `ad`, `bd` and `br` saves computer time in the main loop. These calculations have to be done only when controller parameters are changed. The main loop is executed once every sampling period. The program has three states: `yold`, `I`, and `D`. One state variable can be eliminated at the cost of less readable code. The latency between reading the analog input and setting the analog output consists of four multiplications, four additions and evaluation of the `sat` function. All computations can be done using fixed-point calculations if necessary. Notice that the code computes the filtered derivative of the process output and that it has setpoint weighting and anti-windup protection. Note also that in this code we apply the actuator saturation inside the controller, rather than measuring the actuator output as in Figure 11.11.

11.6 Further Reading

The history of PID control is very rich and stretches back to the beginning of the foundation of control theory. Very readable treatments are given by Bennett [Ben79, Ben93] and Mindel [Min02]. The Ziegler–Nichols rules for tuning PID controllers, first presented in 1942 [ZN42], were developed based on extensive experiments with pneumatic simulators and Vannevar Bush's differential analyzer at MIT. An interesting view of the development of the Ziegler–Nichols rules is given in an interview with Ziegler [Bli90]. An industrial perspective on PID control is given in [Bia95], [Shi96] and [YH91] and in the paper [DM02a] cited in the beginning of this chapter. PID can also be used to explain popular monetary policy rules [HSH15]. A comprehensive presentation of PID control is given in [ÅH05]. Interactive learning tools for PID control can be downloaded from http://www.calerga.com/contrib.

Exercises

11.1 (Ideal PID controllers) Consider the systems represented by the block diagrams in Figure 11.1. Assume that the process has the transfer function $P(s) = b/(s+a)$ and show that the transfer functions from r to y are

$$(a) \quad G_{yr}(s) = \frac{bk_d s^2 + bk_p s + bk_i}{(1+bk_d)s^2 + (a+bk_p)s + bk_i},$$

$$(b) \quad G_{yr}(s) = \frac{bk_i}{(1+bk_d)s^2 + (a+bk_p)s + bk_i}.$$

Pick some parameters and compare the step responses of the systems.

11.2 Consider a second-order process with the transfer function

$$P(s) = \frac{b}{s^2 + a_1 s + a_2}.$$

The closed loop system with a PI controller is a third-order system. Show that it is possible to position the closed loop poles as long as the sum of the poles is $-a_1$. Give equations for the parameters that give the closed loop characteristic polynomial

$$(s+\alpha_0)(s^2+2\zeta_0\omega_0 s+\omega_0^2).$$

11.3 Consider a system with the transfer function $P(s) = (s+1)^{-2}$. Find an integral controller that gives a closed loop pole at $s = -a$ and determine the value of a that maximizes the integral gain. Determine the other poles of the system and judge if the pole can be considered dominant. Compare with the value of the integral gain given by equation (11.7).

11.4 (Ziegler–Nichols tuning) Consider a system with transfer function $P(s) = e^{-s}/s$. Determine the parameters of P, PI and PID controllers using Ziegler–Nichols step and frequency response methods. Compare the parameter values obtained by the different rules and discuss the results.

11.5 (Vehicle steering) Design a proportional-integral controller for the vehicle steering system that gives the closed loop characteristic polynomial

$$s^3+2\omega_0 s^2+2\omega_0 s+\omega_0^3.$$

11.6 (Congestion control) A simplified flow model for TCP transmission is derived in [HMTG00, LPD02]. The linearized dynamics are modeled by the transfer function

$$G_{qp}(s) = \frac{b}{(s+a_1)(s+a_2)}e^{-s\tau_e},$$

which describes the dynamics relating the expected queue length q to the expected packet drop p. The parameters are given by $a_1 = 2N^2/(c\tau_e^2)$, $a_2 = 1/\tau_e$ and $b = c^2/(2N)$. The parameter c is the bottleneck capacity, N is the number of sources feeding the link and τ_e is the round-trip delay time. Use the parameter values $N = 75$ sources, $C = 1250$ packets/s and $\tau_e = 0.15$ and find the parameters of a PI controller using one of the Ziegler–Nichols rules and the corresponding improved rule. Simulate the responses of the closed loop systems obtained with the PI controllers.

11.7 (Motor drive) Consider the model of the motor drive in Exercise 3.10. Develop an approximate second-order model of the system and use it to design an ideal PD controller that gives a closed loop system with eigenvalues in $\zeta\omega_0 \pm i\omega_0\sqrt{1-\zeta^2}$. Add low-pass filtering as shown in equation (11.16) and explore how large ω_0 can be made while maintaining a good stability margin. Simulate the closed loop system with the chosen controller and compare the results with the controller based on state feedback in Exercise 7.12.

11.8 Consider the system in Exercise 11.7 investigate what happens if the second-order filtering of the derivative is replace by a first-order filter.

11.9 (Tuning rules) Apply the Ziegler–Nichols and the modified tuning rules to design PI controllers for systems with the transfer functions

$$P_1 = \frac{e^{-s}}{s}, \qquad P_2 = \frac{e^{-s}}{s+1}, \qquad P_3 = e^{-s}.$$

Compute the stability margins and explore any patterns.

11.10 (Windup and anti-windup) Consider a PI controller of the form $C(s) = 1 + 1/s$ for a process with input that saturates when $|u| > 1$, and whose linear dynamics are given by the transfer function $P(s) = 1/s$. Simulate the response of the system to step changes in the reference signal of magnitude 1, 1.5 and 3. Repeat the simulation when the windup protection scheme in Figure 11.11 is used.

11.11 (Windup protection by conditional integration) Many methods have been proposed to avoid integrator windup. One method called *conditional integration* is to update the integral only when the error is sufficiently small. To illustrate this method we consider a system with PI control described by

$$\frac{dx_1}{dt} = u, \qquad u = \text{sat}_{u_0}(k_p e + k_i x_2), \qquad \frac{dx_2}{dt} = \begin{cases} e & \text{if } |e| < e_0 \\ 0 & \text{if } |e| \geq e_0, \end{cases}$$

where $e = r - x$. Plot the phase portrait of the system for the parameter values $k_p = 1$, $k_i = 1$, $u_0 = 1$ and $e_0 = 1$ and discuss the properties of the system. The example illustrates the difficulties of introducing ad hoc nonlinearities without careful analysis.

Chapter Twelve
Frequency Domain Design

Sensitivity improvements in one frequency range must be paid for with sensitivity deteriorations in another frequency range, and the price is higher if the plant is open-loop unstable. This applies to every controller, no matter how it was designed.

Gunter Stein in the inaugural IEEE Bode Lecture, 1989 [Ste03].

In this chapter we continue to explore the use of frequency domain techniques with a focus on the design of feedback systems. We begin with a more thorough description of the performance specifications for control systems and then introduce the concept of "loop shaping" as a mechanism for designing controllers in the frequency domain. We also introduce some fundamental limitations to performance for systems with time delays and right half-plane poles and zeros.

12.1 Sensitivity Functions

In the previous chapter, we considered the use of proportional-integral-derivative (PID) feedback as a mechanism for designing a feedback controller for a given process. In this chapter we will expand our approach to include a richer repertoire of tools for shaping the frequency response of the closed loop system.

One of the key ideas in this chapter is that we can design the behavior of the closed loop system by focusing on the open loop transfer function. This same approach was used in studying stability using the Nyquist criterion: we plotted the Nyquist plot for the *open* loop transfer function to determine the stability of the *closed* loop system. From a design perspective, the use of loop analysis tools is very powerful: since the loop transfer function is $L = PC$, if we can specify the desired performance in terms of properties of L, we can directly see the impact of changes in the controller C. This is much easier, for example, than trying to reason directly about the tracking response of the closed loop system, whose transfer function is given by $G_{yr} = PC/(1+PC)$.

We will start by investigating some key properties of the feedback loop. A block diagram of a basic feedback loop is shown in Figure 12.1. The system loop is composed of two components: the process and the controller. The controller itself has two blocks: the feedback block C and the feedforward block F. There are two disturbances acting on the process, the load disturbance v and the measurement noise w. The load disturbance represents disturbances that drive the process away from its desired behavior, while the measurement noise represents disturbances that corrupt information about the process given by the sensors. For example in

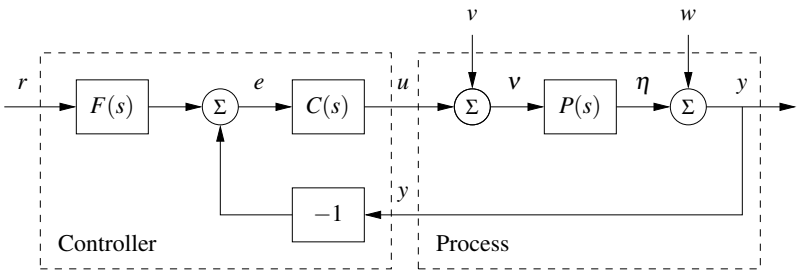


Figure 12.1: Block diagram of a basic feedback loop with two degrees of freedom. The controller has a feedback block C and a feedforward block F. The external signals are the reference signal r, the load disturbance v and the measurement noise w. The process output is η, and the control signal is u.

cruise control the major load disturbance is changes in the slope of the road and measurement noise is caused by the electronics that convert pulses measured on a rotating shaft to a velocity signal. The load disturbances typically have low frequencies, lower than the servo bandwidth, while measurement noise typically has higher frequencies. In the figure, the load disturbance is assumed to act on the process input. This is a simplification since disturbances often enter the process in many different ways, but it allows us to streamline the presentation without significant loss of generality.

The process output η is the real variable that we want to control. Control is based on the measured signal y, where the measurements are corrupted by measurement noise w. The process is influenced by the controller via the control variable u. The process is thus a system with three inputs—the control variable u, the load disturbance v and the measurement noise w—and one output—the measured signal y. The controller is a system with two inputs and one output. The inputs are the measured signal y and the reference signal r, and the output is the control signal u. Note that the control signal u is an input to the process and the output of the controller, and that the measured signal y is the output of the process and an input to the controller.

Since the system is linear, the relations between the inputs and the interesting signals can be expressed in terms of the transfer functions. The following relations are obtained from the block diagram in Figure 12.1:

$$\begin{pmatrix} y \\ \eta \\ v \\ u \\ e \end{pmatrix} = \begin{pmatrix} \frac{PCF}{1+PC} & \frac{P}{1+PC} & \frac{1}{1+PC} \\ \frac{PCF}{1+PC} & \frac{P}{1+PC} & \frac{-PC}{1+PC} \\ \frac{CF}{1+PC} & \frac{1}{1+PC} & \frac{-C}{1+PC} \\ \frac{CF}{1+PC} & \frac{-PC}{1+PC} & \frac{-C}{1+PC} \\ \frac{F}{1+PC} & \frac{-P}{1+PC} & \frac{-1}{1+PC} \end{pmatrix} \begin{pmatrix} r \\ v \\ w \end{pmatrix}. \tag{12.1}$$

In addition, we can write the transfer function for the error between the reference r and the output η (not an explicit signal in the diagram), which satisfies

$$\varepsilon = r - \eta = \Big(1 - \frac{PCF}{1+PC}\Big) r + \frac{-P}{1+PC} v + \frac{PC}{1+PC} w.$$

Notice that many of the transfer functions in (12.1) are the same. Only four transfer functions are required to describe how the system reacts to load disturbances and measurement noise, three additional transfer functions are required to describe how the system responds to reference signals.

The special case of $F = 1$ is called a system with (pure) *error feedback* because all control actions are based on feedback from the error. In this case the system (12.1) is completely characterized by four transfer functions, which are called the sensitivity functions:

$$\begin{aligned} S &= \frac{1}{1+PC} && \text{sensitivity function} & \qquad PS &= \frac{P}{1+PC} && \text{load sensitivity function} \\ T &= \frac{PC}{1+PC} && \text{complementary sensitivity function} & \qquad CS &= \frac{C}{1+PC} && \text{noise sensitivity function} \end{aligned} \tag{12.2}$$

These transfer functions and their equivalent systems are called the *Gang of Four*. The load sensitivity function is sometimes called the input sensitivity function and the noise sensitivity function is sometimes called the output sensitivity function. These transfer functions have many interesting properties that will be discussed in detail in the rest of the chapter. Good insight into these properties is essential in understanding the performance of feedback systems for the purposes of both analysis and design. The transfer functions are also used to formulate requirements on the closed loop system.

When $F \neq 1$ there are three additional transfer functions in (12.1).

$$PS = \frac{P}{1+PC}, \quad CSF = \frac{PCT}{1+PC}, \quad TF = \frac{PCF}{1+PC} \tag{12.3}$$

To summarize we find that the closed loop system can be characterized by seven transfer functions given by (12.2) and (12.3), which we call the *Gang of Seven*.

The closed loop system is stable if all transfer functions (12.2) are stable, which is called *internal stability*. If there are no pole-zero cancellations in the product PC the closed loop system is stable if the rational function $1 + PC$ has all its zeros in the left half plane.

Analyzing the Gang of Seven, we find that the feedback controller C influences the effects of load disturbances and measurement noise. Notice that measurement noise enters the process via the feedback. In Section 13.2 it will be shown that the controller influences the sensitivity of the closed loop to process variations. The feedforward part F of the controller influences only the response to command signals.

In Chapter 10 we focused on the loop transfer function, and we found that its

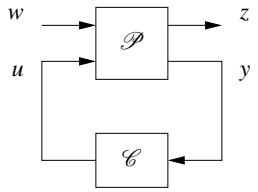


Figure 12.2: A more general representation of a feedback system. The process input u represents the control signal, which can be manipulated, and the process input w represents other signals that influence the process. The process output y is the vector of measured variables and z are other signals of interest.

properties gave useful insights into the properties of a system. To make a proper assessment of a feedback system it is necessary to consider the properties of all the transfer functions (12.2) and (12.3) in the Gang of Seven (or the Gang of Four, if there is no feedforward controller), as illustrated in the following example.

Example 12.1 The loop transfer function gives only limited insight
Consider a process with the transfer function $P(s) = 1/(s-a)$ controlled by a PI controller with error feedback having the transfer function $C(s) = k(s-a)/s$. The loop transfer function is $L = k/s$, and the sensitivity functions are

$$T = \frac{PC}{1+PC} = \frac{k}{s+k}, \qquad PS = \frac{P}{1+PC} = \frac{s}{(s-a)(s+k)},$$
$$CS = \frac{C}{1+PC} = \frac{k(s-a)}{s+k}, \qquad S = \frac{1}{1+PC} = \frac{s}{s+k}.$$

Notice that the factor $s-a$ is canceled when computing the loop transfer function and that this factor also does not appear in the sensitivity function or the complementary sensitivity function. However, cancellation of the factor is very serious if $a > 0$ since the transfer function PS relating load disturbances to process output is then unstable. In particular, a small disturbance v can lead to an unbounded output, which is clearly not desirable. ∇

The system in Figure 12.1 represents a special case because it is assumed that the load disturbance enters at the process input and that the measured output is the sum of the process variable and the measurement noise. Disturbances can enter in many different ways, and the sensors may have dynamics. A more abstract way to capture the general case is shown in Figure 12.2, which has only two blocks representing the process ($\mathscr{P}$) and the controller ($\mathscr{C}$). The process has two inputs, the control signal u and a vector of disturbances w, and two outputs, the measured signal y and a vector of signals z that is used to specify performance. The system in Figure 12.1 can be captured by choosing $w = (r,d,n)$ and $z = (e,u,\nu,\eta,\varepsilon)$. The process transfer function $\mathscr{P}$ from w to z is a 5×3 matrix, and the controller transfer function $\mathscr{C}$ from y,r to u is a 2×1 matrix; compare with Exercise 12.3.

Processes with multiple inputs and outputs can also be considered by regarding u and y as vectors. Representations at these higher levels of abstraction are useful

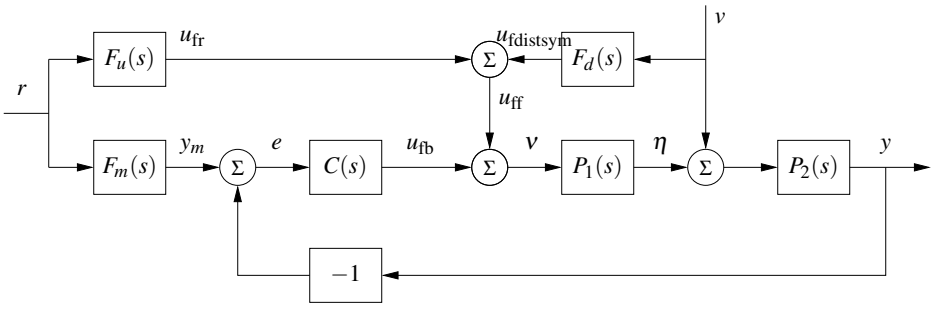


Figure 12.3: Block diagram of a system with feedforward compensation for improved response to reference signals and measured disturbances (2 DOF system). Three feedforward elements are present: $G_{\text{m}}(s)$ sets the desired output value, $F_r(s)$ generates the feedforward command u_{fr} to improve command signal response and $F_v(s)$ generates the feedforward signal u_{fv} that reduces the effect of the measured disturbance v.

for the development of theory because they make it possible to focus on fundamentals and to solve general problems with a wide range of applications. However, care must be exercised to maintain the coupling to the real-world control problems we intend to solve.

12.2 Feedforward Design

Feedforward is a simple and powerful technique that complements feedback. It can be used both to improve the response to reference signals and to reduce the effect of measurable disturbances. Design of feedforward for controllers based on state feedback and observers was developed in Section 8.5, see Figure 8.11. Section 11.5 presented set point weighting as simple form of feedforward for PID controllers, see equation (11.17). In this section we will use transfer functions to develop more advanced methods for feedforward design.

Figure 12.1 shows a block diagram of a system with feedback and feedforward control. The process transfer function $P(s)$ is separated in two blocks $P_1(s)$ and $P_2(s)$, where the measured disturbance v enters at the input of the block P_2. The transfer function G_{m} represents the desired response to reference signals. There are two feedforward blocks with the transfer functions F_r and F_v to deal with the reference signal r and the measured disturbances v.

A major advantage of controllers with two degrees of freedom that combine feedback and feedforward is that the control design problem can be split in two parts. The feedback transfer function C can be designed to give good robustness and effective disturbance attenuation, and the feedforward transfer functions F_r and F_v can be designed independently to give the dessired responses to reference signals and to reduce effects of measured disturbances.

We will first explore the response to command signals. The transfer function

$F_{yr}(s)$ from reference input r to process output y is

$$\begin{aligned} G_{yr}(s) &= \frac{P(CG_{\mathrm{m}}+F_r)}{1+PC} = G_{\mathrm{m}} + \frac{PF_r - G_{\mathrm{m}}}{1+PC} \\ &= G_{\mathrm{m}} + S(PF_r - G_{\mathrm{m}}) = TG_{\mathrm{m}} + SPF_r, \end{aligned} \tag{12.4}$$

where S is the sensitivity function and T the complementary sensitivity function, see (12.2). We can make G_{yr} close to the desired transfer function G_{m} in two different ways, by choosing the feedforward transfer function F_r so that $PF_r - G_{\mathrm{m}}$ is small, or by choosing the feedback transfer function C so that the sensitivity $S = 1/(1+PC)$ is small. Perfect feedforward compensation is obtained for

$$F_r = \frac{G_{\mathrm{m}}}{P}, \tag{12.5}$$

which gives $G_{yr} = G_{\mathrm{m}}$. Notice that the feedforward compensator F_r contains an inverse model of the process dynamics.

Next we will consider attenuation of disturbances that can be measured. The transfer function from load disturbance v to process output y:

$$G_{yv} = \frac{P_2(1-P_1F_v)}{1+PC} = P_2S(1-P_1F_v). \tag{12.6}$$

recall that S is the sensitivity function (12.3). The transfer function G_{yv} can be made small in two different ways, by choosing the feedforward transfer function F_r so that $PF_r - G_{\mathrm{m}}$ is small, or by choosing the feedback transfer function C so that the sensitivity $S = 1/(1+PC)$ is small. Perfect compensation is obtained by choosing

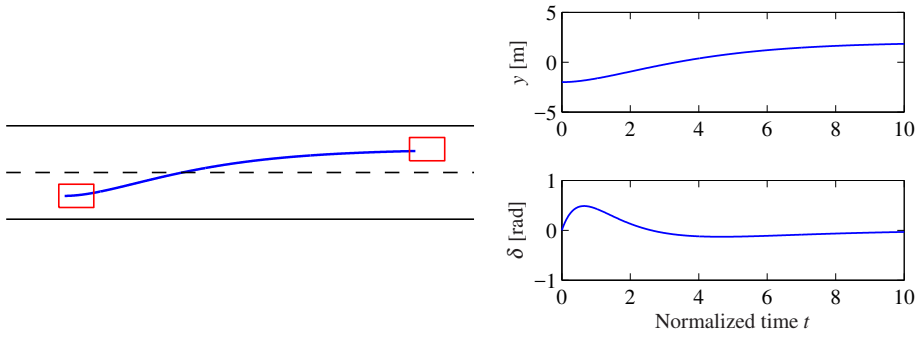
$$F_v = P_1^{-1}. \tag{12.7}$$

Design of feedforward to improve responses to command signals and disturbances using transfer functions is thus a simple task, but it requires inversion of process models. We illustrate with an example.

Example 12.2 Vehicle steering
A linearized model for vehicle steering was given in Example 7.4. The normalized transfer function from steering angle δ to lateral deviation y is $P(s) = (\gamma s+1)/s^2$. For a lane transfer system we would like to have a nice response without overshoot, and we therefore choose the desired response as $G_m(s) = a^2/(s+a)^2$, where the response speed or aggressiveness of the steering is governed by the parameter a. Equation (12.5) gives

$$F_r = \frac{G_m}{P} = \frac{a^2s^2}{(\gamma s+1)(s+a)^2},$$

which is a stable transfer function as long as $\gamma > 0$. Figure 12.4 shows the responses of the system for $a = 0.5$. The figure shows that a lane change is accomplished in about 10 vehicle lengths with smooth steering angles. The largest steering angle is slightly larger than 0.1 rad (6°). Using the scaled variables, the curve showing



(a) Overhead view (b) Position and steering

Figure 12.4: Feedforward control for vehicle steering. The plot on the left shows the trajectory generated by the controller for changing lanes. The plots on the right show the lateral deviation y (top) and the steering angle δ (bottom) for a smooth lane change control using feedforward (based on the linearized model).

lateral deviations (y as a function of t) can also be interpreted as the vehicle path (y as a function of x) with the vehicle length as the length unit. ∇

Sensitivity to Process Variations

Control design is based on knowledge of the transfer function of the process. Since there are always modeling errors we will investigate the effects of small modeling errors. We will first investigate the response to reference signals. The transfer function from reference signals to process output G_{yr} is given by (12.4). Taking the logarithms of this equation gives

$$\log G_{yr} = \log P + \log(F_r + CG_m) - \log(1 + PC).$$

Differentiating with respect to P gives the following expression for the sensitivity of the closed loop transfer function.

$$\frac{dG_{yr}}{G_{yr}} = \frac{dP}{P} - \frac{CdP}{1+PC} = \frac{1}{1+PC}\frac{dP}{P} = S\frac{dP}{P}. \tag{12.8}$$

With pure feedforward we have $C = 0$ and $S = 1$, the relative error in G_{yr} is the same as the relative error in the process transfer function. With feedback the relative model error dP/P is multiplied by the sensitivity function S. Notice in particular that with a controller having integral action we have $S(0) = 0$. Feedback thus gives a significant reduction of the effects of modeling error at low frequencies.

Next we will investigate the transfer function G_{yv} from load disturbances v to process output is

$$G_{yv} = \frac{P_2(1 - P_1F_v)}{1+PC} \tag{12.9}$$

	FF	FB	FB and FF
G_{yr}	G_{m}	TG_{m}	G_{m}
$\dfrac{dG_{yr}}{G_{yr}}$	$\dfrac{dP}{P}$	$S\dfrac{dP}{P}$	$S\dfrac{dP}{P}$
G_{yv}	0	SP_2	0
$\dfrac{dG_{yv}}{P_2}$	$-\dfrac{dP_1}{P_1}$	$-S\dfrac{dP_1}{P_1}$	$-S\dfrac{dP_1}{P_1}$

Table 12.1: Properties of command signal following G_{yr} and disturbance attenuation G_{yv} of closed loop systems based on feedback (FB), feedforward (FF) and combined feedback and feedforward.

For small variations dP of the process transfer function P we have

$$\begin{aligned} dG_{yv} &= \frac{(1+PC)\big(dP_2(1-P_1F_v)-P_2dP_1F_v\big)-P_2(1-P_1F_v)dPC}{(1+PC)^2} \\ &= -\frac{P_2dP_1F_v}{1+PC}+(1-P_1F_v)\frac{dP_2(1+PC)-P_2dPC}{(1+PC)^2} \end{aligned}$$

Using the expression (12.7) for the feedforward transfer function F_r we find

$$\frac{dG_{yv}}{P_2} = -\frac{dP_1}{(1+PC)P_1} = -S\frac{dP_1}{P_1}. \tag{12.10}$$

Notice that the transfer function P_1 is the critical element, variations in P_2 are not as serious.

The properties of feedback and feedforward are summarized in Table 12.1. With feedforward control we have $G_{yr} = G_{\mathrm{m}}$ and $G_{yv} = 0$, with feedback control we have instead $G_{yr} = TG_{\mathrm{m}}$ and $G_{yv} = SP_2$. The sensitivity to modeling errors with pure feedforward is always equal to one, feedback reduces it significantly.

Feedback and feedforward have complementary properties. Feedforward acts by matching two transfer functions, requiring precise knowledge of the process dynamics, while feedback attempts to make the error small by multiplication with a the sensitivity function S. For a controller having integral action, the sensitivity function is small for low frequencies, and it is thus sufficient that the condition for ideal feedforward holds at higher frequencies. This is easier than trying to satisfy the conditions (12.5) and (12.5) for all frequencies.

Difficulties with Feedforward

The feedforward compensators in Figure 12.3 are given by

$$F_r = \frac{G_{\mathrm{m}}}{P}, \qquad F_v = \frac{1}{P_1}. \tag{12.11}$$

Both transfer functions require inversion of process transfer functions. There are problems with inversion if the process transfer function has time delays, right half plane zeros or high pole excess. Inversion of time delays require prediction which can not be done prefectly except in the situation when the command signal is known in advance. The inverse process transfer function is unstable if the process transfer functions has zeros in the right half plane. The inverse requires differentiation if the pole excess of the process transfer funtion is greater than zero. We must then require that the reference signal is sufficiently smooth there may also be problems with noise. Approximate inverses have to be used when the inverse is unstable.

There is an extra freedom when finding the transfer function F_r because it also contains the transfer function G_{m}, which specifies the ideal behavior. A stable feedforward transfer function can be obtained if G_{m} has the same time delays and right half plane zeros as the process. We illustrate with an example.

Example 12.3 Process with a Right Half Plane Zero
Let the process and the desired response have the transfer functions

$$P(s) = \frac{1-s}{(s+1)^2}, \qquad G_{\mathrm{m}}(s) = \frac{\omega_m^2(1-s)}{s^2+2\zeta_c\omega_m s+\omega_m^2}.$$

Since the process has a right half plane zero at $s = 1$ the desired transfer function $G_{\mathrm{m}}(s)$ must have the same zeros to avoid having an unstable feedforward transfer function F_{r}. Equation (12.5) gives the feedforward transfer function

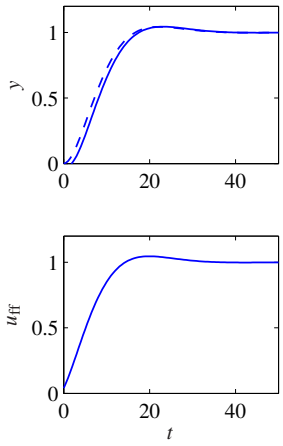
$$F_{\mathrm{r}}(s) = \frac{\omega_m^2(s+1)^2}{s^2+2\zeta_c\omega_m s+\omega_m^2}. \tag{12.12}$$

Figure 12.5 shows the outputs y and the feedforward signals u_{ff} for different values of ω_m. The response to the command signal goes in the wrong direction initially because of the right half plane zero at $s = 1$. This effect, called *inverse response*, is barely noticeable if the response is slow ($\omega_m = 1$) but increases with increasing response speed. For $\omega_m = 5$ the undershoot is more than 200%. A reasonable choise of ω_m is in the range 0.2 to 0.5.

Notice that the feedforward transfer function (12.12) is also obtained if the process and the desired model have the transfer functions

$$P(s) = \frac{1}{(s+1)^2}, \qquad G_{\mathrm{m}}(s) = \frac{\omega_m^2}{s^2+2\zeta_c\omega_m s+\omega_m^2}.$$

The corresponding responses are shown in dashed lines in Figure 12.5. When there is no right half plane zero it is thus possible to get nice fast responses. A comparison of the full and solid lines in Figure 12.5 shows that the presence of a right half plane zero in the process transfer function is a severe limitation. The control signals for different values of ω_m differ significantly. To understand what happens we note that $F_{\mathrm{r}}(0) = 1$ and that $F_{\mathrm{r}}(\infty) = \omega_m^2$. For a unit step reference signal $r = 1$, the initial value of the control signal is $u_{\mathrm{ff}}(0) = F_{\mathrm{r}}(\infty) = \omega_m^2$ and the final value is $u_{\mathrm{ff}}(\infty) = F_{\mathrm{f}}(0) = 1$. For $\omega_m = 0.2$ the control signal grows from 0.04 to the final



(a) $\omega_m = 0.2$ **MISSING** (b) $\omega_m = 1$ **MISSING** (c) $\omega_m = 5$

Figure 12.5: Outputs y (top plots) and feedforward signals u_{ff} (lower plots) for a unit step command signal. The design parameter has the values $\omega_m = 0.2$ (left), 1 (center) and 0.5 (right) for a unit step command in the reference signal. The dashed curve shows the response that could be achieved if the process did not have the right half plane zero.

value 1 with a small overshoot. For $\omega_m = 1$ the control signal starts from 1 has an overshoot and settles on the final value 1. For $\omega_m = 5$ the control signals starts at 25 and decays towards the final value 1 with an undershoot. ∇

Approximate Inverses

Processes with right half plane zeros do not have stable inverses. To design feedforward compensators for such systems we need approximate inverses that are stable. The following theorem with is presented without proof is useful.

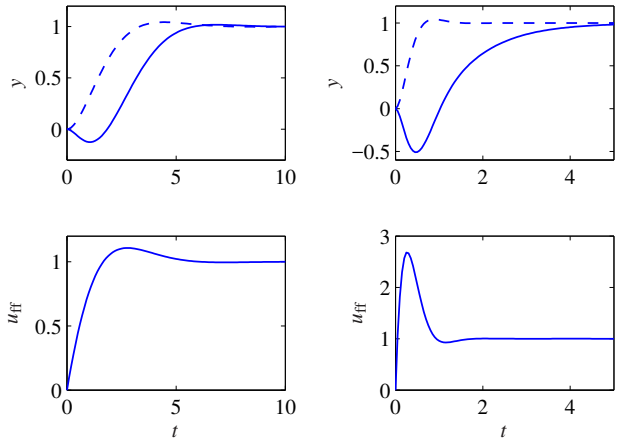
Theorem 12.1 (Approximate Inverse)**.** *Let the rational transfer function $G(s)$ have all its poles in the left half plane and no zeros on the imaginary axis. Factor the transfer function as $G(s) = G^+(s)G^-(s)$, where $G^+(s)$ has all its zeros in the left half plane and $G^-(s)$ has all its zeros in the right half plane. An approximate stable inverse of $G(s)$ that minimizes the mean square error for a step input is*

$$G^\dagger(s) = \frac{1}{G^+(s)G^-(-s)}. \tag{12.13}$$

We illustrate the theorem with an example

Example 12.4 Approximate Inverses
Let the transfer functions of the process and the model be

$$P(s) = \frac{1-s}{(s+1)^2}, \qquad G_{\text{m}}(s) = \frac{\omega_m^2}{s^2 + 2\zeta_c\omega_m s + \omega_m^2},$$



MISSING

(a) $\omega_m = 0.2$ (b) $\omega_m = 1$ (c) $\omega_m = 5$

Figure 12.6: Feedforward design based on an approximate inverse. Outputs y (top plots) and feedforward signals u_{ff} (lower plots) for a unit step command signal. The design parameter has the values $\omega_m = 0.2$ (left) 1 (center) and 0.5 (right) for a unit step command in the reference signal. The dashed curve shows the response that could be achieved if the process did not have the right half plane zero.

compare with Example 12.3. The process transfer function can be factored as

$$P^-(s) = 1 - s, \quad P^+(s) = \frac{1}{s+1)^2}.$$

Theorem 12.1 then gives the following approximate inverse

$$P^\dagger(s) = \frac{1}{P^+(s)P^-(s)} = \frac{1+s}{(s+1)^2} = \frac{1}{s+1}.$$

The feedforward transfer function is then

$$F_r(s) = \frac{G_{\text{m}}(s)}{P^\dagger(s)} = \frac{\omega_m^2(s+1)}{s^2 + 2\zeta_c\omega_m s + \omega_m^2},$$

compared with (12.12). The transfer function from reference r to output y is then

$$G_{yr}(s) = P(s)F_r(s) = \frac{1-s}{(s^2 + 2\zeta_c\omega_m s + \omega_m^2)(s+1)}.$$

Figure 12.6 iillustrates the step responses for different values of ω_m.

Comparing Figures Figure 12.5 and Figure 12.6 we find that there are small differences for $\omega_m = 0.2$. and very large differences for $\omega_m = 5$, notice in particular the differences between the feedforward signal u_{ff}. The design based on the approximate inverse has smaller undershoot but the response have larger lags. ∇

Summary

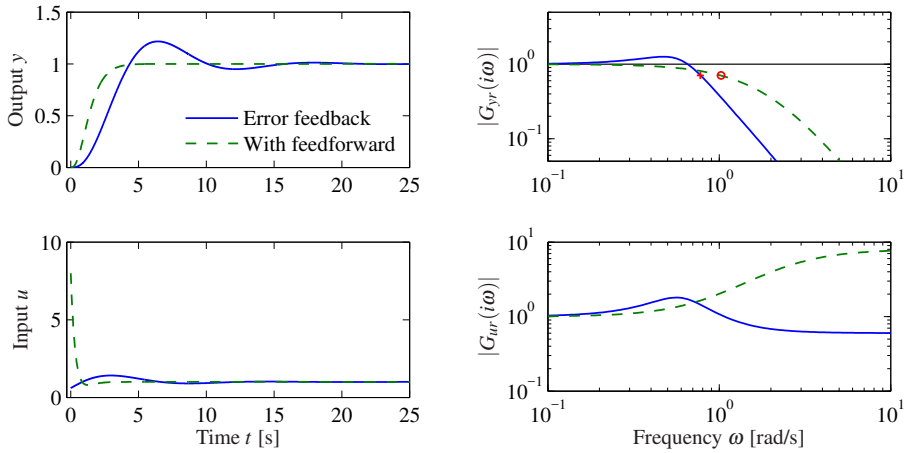
Feedforward can be used to improve the responses to command signals and to reduce the effects of load disturbances that can be measured. Feedforward compensation has inheriently unit sensitivity to model uncertainty, a combination with feedback can reduce the effects of model uncertainty significantly. A feedback controller with integral action reduces the effects of model errors at low frequencies. Design of feedforward is a simple manipulation of transfer functions. There are limitations if the process has time delays, right half plane zeros and high pole excess.

12.3 Performance Specifications

A key element of the control design process is how we specify the desired performance of the system. It is also important for users to understand performance specifications so that they know what to ask for and how to test a system. Specifications are often given in terms of robustness to process variations and responses to reference signals and disturbances. They can be given in terms of both time and frequency responses. Specifications for the step response to reference signals were given in Figure 6.9 in Section 6.3 and in Section 7.3. Robustness specifications based on frequency domain concepts were provided in Section 10.3 and will be considered further in Chapter 13. The specifications discussed previously were based on the loop transfer function. Since we found in Section 12.1 that a single transfer function did not always characterize the properties of the closed loop completely, we will give a more complete discussion of specifications in this section, based on the full Gang of Seven.

The transfer function gives a good characterization of the linear behavior of a system. To provide specifications it is desirable to capture the characteristic properties of a system with a few parameters. Common features of step responses are overshoot, rise time and settling time, as shown in Figure 6.9. Common features of frequency responses are resonant peak, peak frequency, gain crossover frequency and bandwidth. A *resonant peak* is a maximum of the gain, and the peak frequency is the corresponding frequency. The *gain crossover frequency* is the smallest frequency where the open loop gain is equal one. The *bandwidth* is defined as the frequency range where the closed loop gain is $1/\sqrt{2}$ of the low-frequency gain (low-pass), mid-frequency gain (band-pass) or high-frequency gain (high-pass). The crossover frequency and the bandwidth are only well defined if the curves are monotone, if this is not the case the bandwidth is typically defined as the lowest frequency.

There are interesting relations between specifications in the time and frequency domains. Roughly speaking, the behavior of time responses for short times is related to the behavior of frequency responses at high frequencies, and vice versa. The precise relations are given by the Laplace transform.



(a) Step responses (b) Frequency responses

Figure 12.7: Reference signal responses. The responses in process output y and control signal u to a unit step in the reference signal r are shown in (a), and the gain curves of G_{yr} and G_{ur} are shown in (b). Results with PI control with error feedback are shown by solid lines, and the dashed lines show results for a controller with a feedforward compensator.

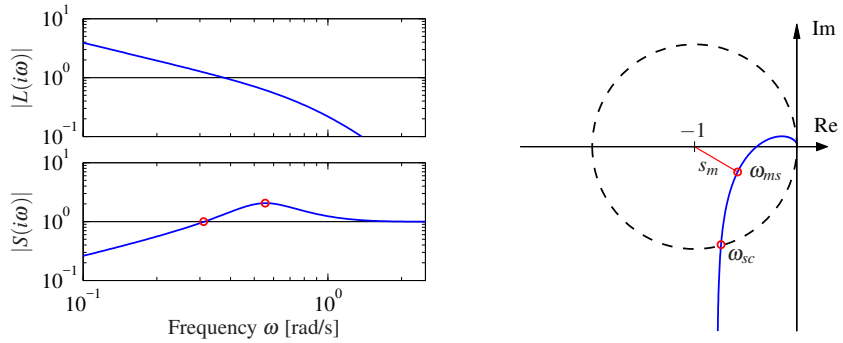
Response to Reference Signals

Consider the basic feedback loop in Figure 12.1. The response to reference signals is described by the transfer functions $G_{yr} = PCF/(1+PC)$ and $G_{ur} = CF/(1+PC)$ ($F = 1$ for systems with error feedback). Notice that it is useful to consider both the response of the output and that of the control signal. In particular, the control signal response allows us to judge the magnitude and rate of the control signal required to obtain the output response.

Example 12.5 Third-order system
Consider a process with the transfer function $P(s) = (s+1)^{-3}$ and a PI controller with error feedback having the gains $k_p = 0.6$ and $k_i = 0.5$. The responses are illustrated in Figure 12.7. The solid lines show results for a proportional-integral (PI) controller with error feedback. The dashed lines show results for a controller with feedforward designed to give the transfer function $G_{yr} = (0.5s+1)^{-3}$. Looking at the time responses, we find that the controller with feedforward gives a faster response with no overshoot. However, much larger control signals are required to obtain the fast response. The largest value of the control signal is 8, compared to 1.2 for the regular PI controller. The controller with feedforward has a larger bandwidth (marked with ∘) and no resonant peak. The transfer function G_{ur} also has higher gain at high frequencies. ∇

Response to Load Disturbances and Measurement Noise

A simple criterion for disturbance attenuation is to compare the output of the closed loop system in Figure 12.1 with the output of the corresponding open loop



(a) Gain curves (b) Nyquist plot

Figure 12.8: Graphical interpretation of the sensitivity function. Gain curves of the loop transfer function and the sensitivity function (a) can be used to calculate the properties of the sensitivity function through the relation $S = 1/(1+L)$. The sensitivity crossover frequency ω_{sc} and the frequency ω_{ms} where the sensitivity has its largest value are indicated in the sensitivity plot. The Nyquist plot (b) shows the same information in a different form. All points inside the dashed circle have sensitivities greater than 1.

system obtained by setting $C = 0$. If we let the disturbances for the open and closed loop systems be identical, the output of the closed loop system is then obtained simply by passing the open loop output through a system with the transfer function S. The sensitivity function tells how the variations in the output are influenced by feedback (Exercise 12.7). Disturbances with frequencies such that $|S(i\omega)| < 1$ are attenuated, but disturbances with frequencies such that $|S(i\omega)| > 1$ are amplified by feedback. The maximum sensitivity M_s, which occurs at the frequency ω_{ms}, is thus a measure of the largest amplification of the disturbances. The maximum magnitude of $1/(1+L)$ corresponds to the minimum of $|1+L|$, which is precisely the stability margin s_m defined in Section 10.3, so that $M_s = 1/s_m$. The maximum sensitivity is therefore also a robustness measure.

If the sensitivity function is known, the potential improvements by feedback can be evaluated simply by recording a typical output and filtering it through the sensitivity function. A plot of the gain curve of the sensitivity function is a good way to make an assessment of the disturbance attenuation. Since the sensitivity function depends only on the loop transfer function, its properties can also be visualized graphically using the Nyquist plot of the loop transfer function. This is illustrated in Figure 12.8. The complex number $1 + L(i\omega)$ can be represented as the vector from the point -1 to the point $L(i\omega)$ on the Nyquist curve. The sensitivity is thus less than 1 for all points outside a circle with radius 1 and center at -1. Disturbances with frequencies in this range are attenuated by the feedback.

The transfer function G_{yv} from load disturbance v to process output y for the system in Figure 12.1 is

$$G_{yv} = \frac{P}{1+PC} = PS = \frac{T}{C}. \tag{12.14}$$

Since load disturbances typically have low frequencies, it is natural to focus on the behavior of the transfer function at low frequencies. Consider a system with $P(0) \neq 0$ and a controller with integral action with integral gaink_i. The controller transfer function goes to infinity as k_i/s for small s and we have the following approximation:

$$G_{yv} = \frac{T}{C} \approx \frac{1}{C} \approx \frac{s}{k_i}, \tag{12.15}$$

The process transfer function P typically goes to zero for large s as does a well designed controller C. The loop transfer function PC then also goes to zero and the sensitivity function S goes to 1 for large s. We then have the approximation $G_{yv} \approx P$ for large s.

Measurement noise, which typically has high frequencies, generates rapid variations in the control variable that are detrimental because they cause wear in many actuators and can even saturate an actuator. It is thus important to keep variations in the control signal due to measurement noise at reasonable levels—a typical requirement is that the variations are only a fraction of the span of the control signal. The variations can be influenced by filtering and by proper design of the high-frequency properties of the controller.

The effects of measurement noise are captured by the transfer function from the measurement noise to the control signal,

$$-G_{uw} = \frac{C}{1+PC} = CS = \frac{T}{P}. \tag{12.16}$$

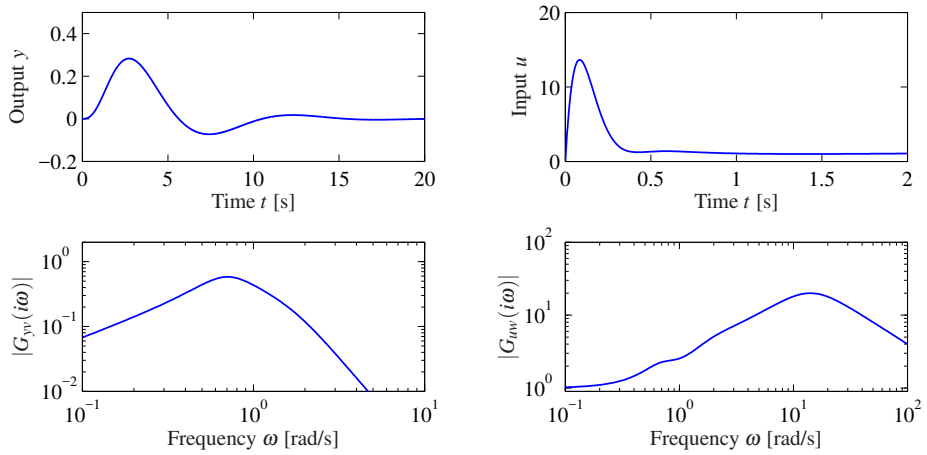
For controllers with integral action, the complementary sensitivity function is close to 1 for low frequencies ($\omega < \omega_{\text{gc}}$), and G_{uw} can be approximated by $-1/P$. Similarly, the sensitivity function is close to 1 for high frequencies ($\omega > \omega_{\text{gc}}$), and G_{uw} can be approximated by $-C$.

Example 12.6 Third-order system
Consider a process with the transfer function $P(s) = (s+1)^{-3}$ and a proportional-integral-derivative (PID) controller with gains $k_p = 0.6$, $k_i = 0.5$ and $k_d = 2.0$. We augment the controller using a second-order noise filter with damping ratio $1/\sqrt{2}$ and $T_f = 0.1$. The controller transfer function then becomes

$$C(s) = \frac{k_d s^2 + k_p s + k_i}{s(s^2 T_f^2/2 + sT_f + 1)}.$$

The closed loop system responses are illustrated in Figure 12.9. The closed loop response of the output to a step in the load disturbance in the top part of Figure 12.9a has a peak of 0.28 at time $t = 2.73$ s. The frequency response in Figure 12.9a shows that the gain has a maximum of 0.58 at $\omega = 0.7$ rad/s.

The closed loop response of the control signal to a step in measurement noise is shown in Figure 12.9b. The high-frequency roll-off of the transfer function $G_{uw}(i\omega)$ is due to filtering; without it the gain curve in Figure 12.9b would continue to rise after 20 rad/s. The step response has a peak of 13 at $t = 0.08$ s. The frequency response has its peak 20 at $\omega = 14$ rad/s. Notice that the peak occurs



(a) Output load response (b) Input noise response

Figure 12.9: Closed loop disturbance responses for Example 12.6. The closed loop time and frequency responses of process output y to load disturbance v are shown in (a) and the responses of control signal u to measurement noise w are shown in (b).

at a frequency far above the peak of the response to load disturbances and far above the gain crossover frequency $\omega_{gc} = 0.78$ rad/s. An approximation derived in Exercise 12.9 gives $\max |CS(i\omega)| \approx k_d/T_f = 20$, which occurs at $\omega = \sqrt{2}/T_d = 14.1$ rad/s. ∇

12.4 Feedback Design via Loop Shaping

One advantage of the Nyquist stability theorem is that it is based on the loop transfer function, which is related to the controller transfer function through $L = PC$. It is thus easy to see how the controller influences the loop transfer function. To make an unstable system stable we simply have to bend the Nyquist curve away from the critical point.

This simple idea is the basis of several different design methods collectively called *loop shaping*. These methods are based on choosing a compensator that gives a loop transfer function with a desired shape. One possibility is to determine a loop transfer function that gives a closed loop system with the desired properties and to compute the controller as $C = L/P$. This approach may lead to controllers of high order and there are limitaions if the process transfer function has poles and zeros in the right half plane as will be discussed in Section 12.6. Another possibility is to start with the process transfer function, change its gain and then add poles and zeros until the desired shape is obtained. In this section we will explore different loop-shaping methods for control law design.

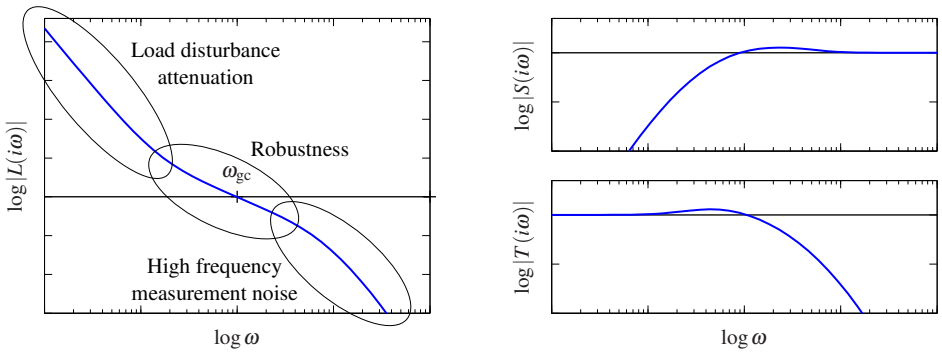


Figure 12.10: Gain curve and sensitivity functions for a typical loop transfer function. The plot on the left shows the gain curve and the plots on the right show the sensitivity function and complementary sensitivity function. The crossover frequency ω_{gc} determines the attenuation of load disturbances, bandwidth and response time of the closed loop system. The slope n_{gc} determines the robustness of the closed loop systems, see (12.17). At low frequencies, a large magnitude of L provides good load disturbance rejection and reference tracking, while at high frequencies a small loop gain avoids injecting too much measurement noise.

Design Considerations

We will first discuss a suitable shape for the loop transfer function that gives good performance and good stability margins. Figure 12.10 shows a typical loop transfer function. Good robustness requires good stability margins (or good gain and phase margins), which imposes requirements on the loop transfer function around the crossover frequencies ω_{pc} and ω_{gc}. The gain of L at low frequencies must be large in order to have good tracking of command signals and good attenuation of low-frequency disturbances. Since $S = 1/(1+L)$, it follows that for frequencies where $|L| > 101$ disturbances will be attenuated by a factor of 100 and the tracking error is less than 1%. It is therefore desirable to have a large crossover frequency and a steep (negative) slope of the gain curve. The choice of gain crossover frequency ω_{gc} is a compromise among attenuation of load disturbances, injection of measurement noise and robustness. The controller gain at low frequencies can be increased by a controller with integral action, which is also called *lag compensation*. To avoid injecting too much measurement noise into the system, the controller transfer function should have low gain at high frequencies, a property called *high-frequency roll-off*.

Bode's relations (see Section 10.4) impose restrictions on the shape of the loop transfer function. Equation (10.8) implies that the slope of the gain curve at gain crossover cannot be too steep. If the gain curve has a constant slope, we have the following relation between slope n_{gc} and phase margin φ_m:

$$n_{gc} = -2 + \frac{2\varphi_m}{\pi}. \tag{12.17}$$

This formula is a reasonable approximation when the gain curve does not deviate too much from a straight line. It follows from equation (12.17) that the phase

margins 30°, 45° and 60° correspond to the slopes $-5/3$, $-3/2$ and $-4/3$.

Loop shaping is a trial-and-error procedure. We typically start with a Bode plot of the process transfer function. We then attempt to shape the loop transfer function by changing the controller gain and adding poles and zeros to the controller transfer function. Different performance specifications are evaluated for each controller as we attempt to balance many different requirements by adjusting controller parameters and complexity. Loop shaping is straightforward to apply to single-input, single-output systems. It can also be applied to systems with one input and many outputs by closing the loops one at a time. The only limitation for minimum phase systems is that large phase leads and high controller gains may be required to obtain closed loop systems with a fast response. Many specific procedures are available: they all require experience, but they also give good insight into the conflicting requirements. There are fundamental limitations to what can be achieved for systems that are not minimum phase; they will be discussed in the next section.

Lead and Lag Compensation

A simple way to do loop shaping is to start with the transfer function of the process and add simple compensators with the transfer function

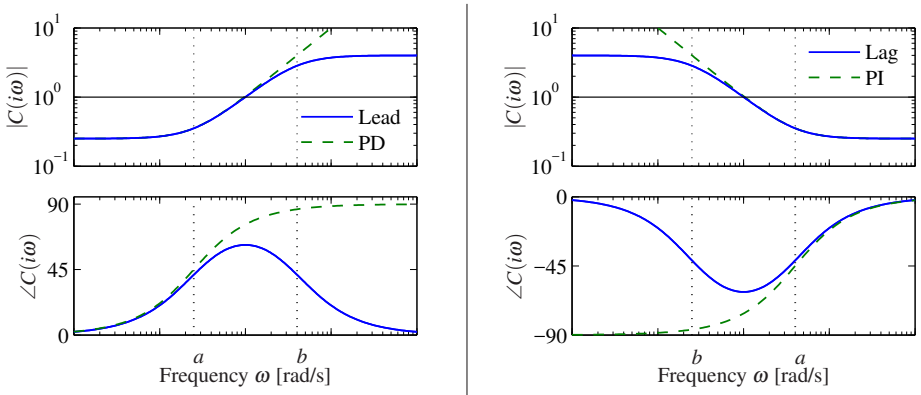
$$C(s) = k\frac{s+a}{s+b}, \quad a > 0, b > 0. \tag{12.18}$$

The compensator is called a *lead compensator* if $a < b$, and a *lag compensator* if $a > b$. The PI controller is a special case of a lag compensator with $b = 0$, and the ideal PD controller is a special case of a lead compensator with $a = 0$. Bode plots of lead and lag compensators are shown in Figure 12.11. Lag compensation, which increases the gain at low frequencies, is typically used to improve tracking performance and disturbance attenuation at low frequencies. Lead and lag compensators can also be combined to form a lead-lag compensator (see Exercise 12.11). Compensators that are tailored to specific disturbances can be also designed, as shown in Exercise 12.10. Lead compensation is typically used to improve phase margin. The following examples gives an illustration of the use of lag compensation (via PI control).

Example 12.7 Atomic force microscope in tapping mode
A simple model of the dynamics of the vertical motion of an atomic force microscope in tapping mode was given in Exercise 10.2. The transfer function for the system dynamics is

$$P(s) = \frac{a(1-e^{-s\tau})}{s\tau(s+a)},$$

and the parameters $a = \zeta\omega_0$, $\tau = 2\pi n/\omega_0$ are explained in XXX. The gain has been normalized to 1. A Bode plot of this transfer function for the parameters $a = 1$ and $\tau = 0.25$ is shown in dashed curves in Figure 12.12a. To improve the attenuation of load disturbances we increase the low-frequency gain by introducing



(a) Lead compensation, $a < b$ (b) Lag compensation, $b < a$

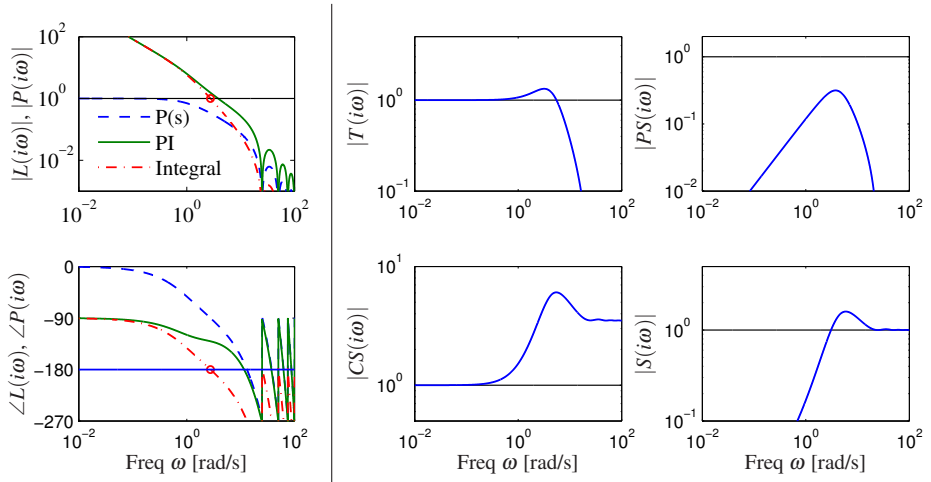
Figure 12.11: Frequency response for lead and lag compensators $C(s) = k(s+a)/(s+b)$. Lead compensation (a) occurs when $a < b$ and provides phase lead between $\omega = a$ and $\omega = b$. Lag compensation (b) corresponds to $a > b$ and provides low-frequency gain. PI control is a special case of lag compensation and PD control is a special case of lead compensation. PI/PD frequency responses are shown by dashed curves.

an integral controller. The loop transfer function then becomes $L = k_i P(s)/s$, and we start by adjusting the gain k_i so that the closed loop system is marginally stable, giving $k_i = 8.3$. The Bode plot is shown by the dash-dotted line in Figure 12.12a, where the critical point is indicated by $\circ$. Notice the increase of the gain at low frequencies. To obtain a reasonable phase margin we introduce proportional action and we increase the proportional gain k_p gradually until reasonable values of the sensitivities are obtained. The value $k_p = 3.5$ gives maximum sensitivity $M_s = 1.6$ and maximum complementary sensitivity $M_t = 1.3$. The loop transfer function is shown in solid lines in Figure 12.12a. Notice the significant increase of the phase margin compared with the purely integral controller (dash-dotted line).

To evaluate the design we also compute the gain curves of the transfer functions in the Gang of Four. They are shown in Figure 12.12b. The peaks of the sensitivity curves are reasonable, and the plot of PS shows that the largest value of PS is 0.3, which implies that the load disturbances are well attenuated. The plot of CS shows that the largest noise gain $|C(i\omega)S(i\omega)|$ is 6. The controller has a gain $k_p = 3.5$ at high frequencies, and hence we may consider adding high-frequency roll-off to make CS smaller at high frequencies. ∇

A common problem in the design of feedback systems is that the phase margin is too small, and phase *lead* must then be added to the system. If we set $a < b$ in equation (12.18), we add phase lead in the frequency range between the pole/zero pair (and extending approximately $10\times$ in frequency in each direction). By appropriately choosing the location of this phase lead, we can provide additional phase margin at the gain crossover frequency.

Because the phase of a transfer function is related to the slope of the magnitude, increasing the phase requires increasing the gain of the loop transfer function over



(a) Loop shaping (b) Gang of Four

Figure 12.12: Loop-shaping design of a controller for an atomic force microscope in tapping mode. (a) Bode plots of the process (dashed), the loop transfer function for an integral controller with critical gain (dash-dotted) and a PI controller (solid) adjusted to give reasonable robustness. (b) Gain curves for the Gang of Four for the system.

the frequency range in which the lead compensation is applied. In Exercise 12.15 it is shown that the gain increases exponentially with the amount of phase lead. We can also think of the lead compensator as changing the slope of the transfer function and thus shaping the loop transfer function in the crossover region (although it can be applied elsewhere as well).

Example 12.8 Roll control for a vectored thrust aircraft

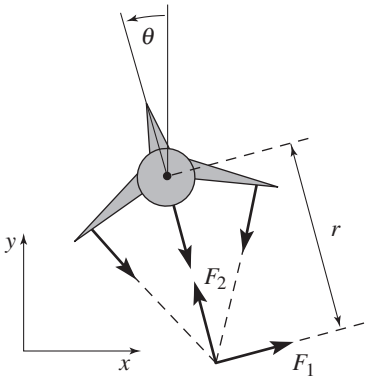
Consider the control of the roll of a vectored thrust aircraft such as the one illustrated in Figure 12.13. Following Exercise 9.10, we model the system with a second-order transfer function of the form

$$P(s) = \frac{r}{Js^2},$$

with the parameters given in Figure 12.13b. We take as our performance specification that we would like less than 1% error in steady state for and less than 10% tracking error up to 10 rad/s.

The open loop transfer function from F_1 to θ is shown in Figure 12.14a. To achieve our performance specification, we would like to have a gain of at least 10 at a frequency of 10 rad/s, requiring the gain crossover frequency to be at a higher frequency. We see from the loop shape that in order to achieve the desired performance we cannot simply increase the gain since this would give a very low phase margin. Instead, we must increase the phase at the desired crossover frequency.

To accomplish this, we use a lead compensator (12.18) with $a = 2$ and $b = 50$. We then set the gain of the system to provide a large loop gain up to the desired bandwidth, as shown in Figure 12.14b. We see that this system has a gain of greater



(a) Simplified model

Symbol	Description	Value
m	Vehicle mass	4.0 kg
J	Vehicle inertia, φ_3 axis	0.0475 kg m^2
r	Force moment arm	25.0 cm
c	Damping coefficient	0.05 kg m/s
g	Gravitational constant	9.8 m/s^2

(b) Parameter values

Figure 12.13: Roll control of a vectored thrust aircraft. (a) The roll angle θ is controlled by applying maneuvering thrusters, resulting in a moment generated by F_1. (b) The table lists the parameter values for a laboratory version of the system.

than 10 at all frequencies up to 10 rad/s and that it has more than 60° of phase margin. ∇

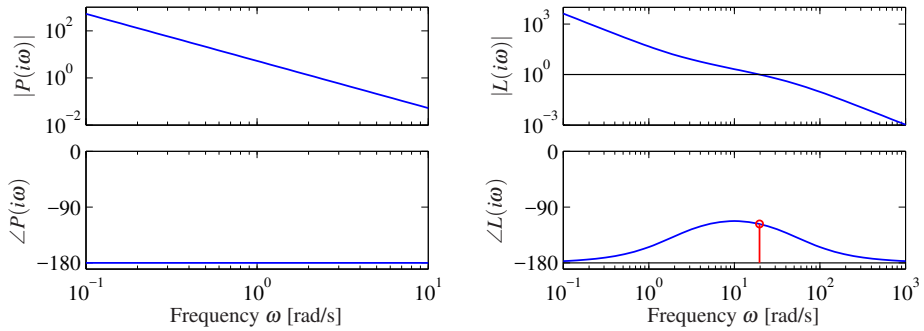
The action of a lead compensator is essentially the same as that of the derivative portion of a PID controller. As described in Section 11.5, we often use a filter for the derivative action of a PID controller to limit the high-frequency gain. This same effect is present in a lead compensator through the pole at $s = b$.

Equation (12.18) is a first-order compensator and can provide up to 90° of phase lead. Larger phase lead can be obtained by using a higher-order lead compensator (Exercise 12.15):

$$C(s) = k\frac{(s+a)^n}{(s+b)^n}, \qquad a < b.$$

12.5 The Root-Locus Method

In design methods such as eigenvalue assignment, discussed in Sections 7.2 and 8.3, we designed controllers that give desired closed loop poles. The controllers were sufficiently complex so that all closed loop poles could be specified. The complexity of the controller is thus related to the complexity of the process. In a practice we may have to use a simple controller for a complex process and it is then not possible to find a controller that gives all closed poles their desired values. It is interesting to explore what can be done with a controller having restricted complexity as was the case for loop shaping in Section 12.4. The simplest case with only one selectable controller parameter can be investigated with the *root locus method*. The *root locus* is a graph of the roots of the characteristic equation as a function of the parameter. The method gives insight into the effects of the controller parameter. It is straight forward to obtain the root locus by finding the roots



(a) Process dynamics (b) Lead compensator

Figure 12.14: Control design for a vectored thrust aircraft using lead compensation. The Bode plot for the open loop process P is shown in (a) and for the loop transfer function $L = PC$ using a lead compensator C in (b). Note the phase lead in the crossover region near $\omega = 100$ rad/s.

of of the closed-loop characteristic polynomial for different values of the parameter. There are also good computer tools for generating the root locus. Of greater interest is that the general shape of the root locus can be obtained with very little effort. The root locus also gives considerable insight.

Proportional Control

To illustrate the root locus method we consider a process with the transfer function

$$P(s) = \frac{b(s)}{a(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \cdots b_m}{s^n + a_1 s^{n-1} + \cdots a_n} = b_0 \frac{(s-z_1)(s-z_2)\ldots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}.$$

The polynomial $a(s)$ has degree n and the polynomial $b(s)$ has degree m. We assume that the integer $n_{pe} = n - m$, which is called the *pole excess* is positive or zero. The controller is assumed to be a proportional controller with the transfer function $C(s) = k$. We will explore the poles of the closed loop system when the gain k of the proportional controller ranges from 0 to ∞.

The closed loop characteristic polynomial is

$$a_{cl}(s) = a(s) + kb(s), \tag{12.19}$$

and the closed loop poles are the roots of $a_{cl}(s)$. The root-locus is a graph of the roots of $a_{cl}(s)$ as the gain k is varies from 0 to ∞. Since the polynomial $a_{cl}(s)$ has degree n the plot will have n branches.

The branches start from the open loop poles. When the gain is k zero we have $a_{cl}(s) = a(s)$ and the closed loop poles are equal to the open loop poles. When there are open loop poles with multiplicity n^* the characteristic equation can be written as

$$(s-p_l)^{n^*}\tilde{a}(s) + kb(s) \approx (s-p_1)^{n^*}\tilde{a}(p_l) + kb(p_l) = 0.$$

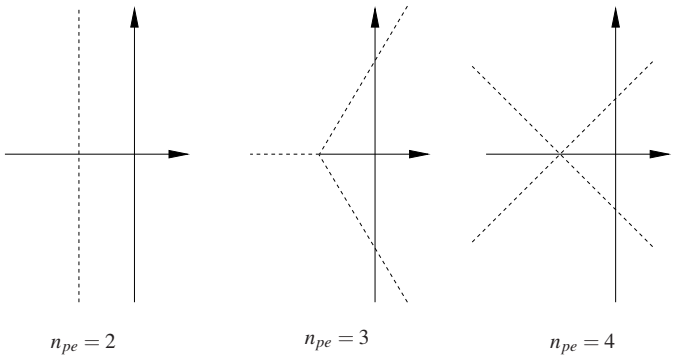


Figure 12.15: Asymptotes of of root locus for systems with pole excess $n_{pe} = 2$, 3 and 4. There are n_{pe} asymptotes radiate from the point given by equation 12.21, and the angles between the asymptotes are $2\pi/n_{pe}$.

For small values of k the roots are $s = p_l + \sqrt[n^*]{-kb(p_l)/\tilde{a}(p_l)}$. The root locus has a star pattern with n^* branches emanating from the open loop pole $s = p_l$. The angle between two neighboring brances is $2\pi/n^*$.

To explore what happens for large gain we approximate the characteristic equation (12.19) for large s and k, hence

$$a_{cl}(s) = b(s)\Big(\frac{a(s)}{b(s)} + k\Big) \approx b(s)\Big(\frac{s^{n_{pe}}}{b_0} + k\Big). \tag{12.20}$$

For large k the closed loop poles are approximately the roots of $b(s)$ and $\sqrt[n_{pe}]{-b_0 k}$. A better approximation of (12.20) is

$$s = s_0 + \sqrt[n_{pe}]{-kb_0}, \qquad s_0 = \frac{1}{n_{pe}}\left(\sum_{k=1}^{n} p_k - \sum_{k=1}^{m} z_k\right), \tag{12.21}$$

see Exercise 12.12. The asymptotes are thus n_{pe} lines that radiate from $s = s_0$, the center of mass of poles and zeros. When $b_0 k > 0$ the lines have the angles $(\pi + 2l\pi)/n_{pe}, l = 1, \cdots, n_{pe}$ with the real line. Figure 12.15 shows the asymptotes of the root locus for large gain for different values of the pole excess n_{pe}.

Summarizing we find that he root locus thus has n branches that start at the loop poles and end either at the open loop poles or at infinity. The branches that ends at infinity have the star-patterned asymptotes given by (12.21). An immediate consequence is that and open loop systems with right half plane zeros or a pole excess larger than 2 will always be unstable for sufficiently large gains.

There are simple rules for sketching root loci. Let it suffice to mention a few of them. The root locus has locally the symmetric star pattern at points where there are multiple roots; the number of branches depend on the multiplicity of the roots. For systems with $kb_0 > 0$ the root locus has segments on the real line where there are odd numbers of real poles and zeros to the right of the segment, see Exercise 12.13. It is also straight forward to find direction where a branch of the

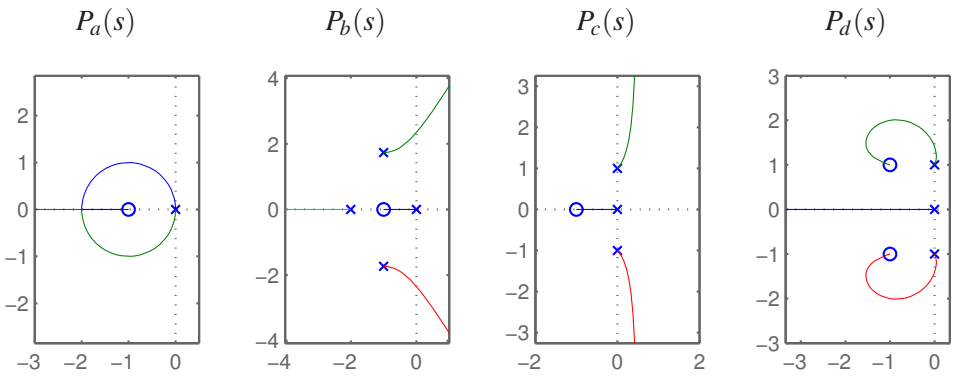


Figure 12.16: Examples of root loci for processes with the transfer functions $P_a(s)$, $P_b(s)$, $P_c(s)$ and $P_d(s)$ given by equation (12.22).

rool locus leaves a pole, see Exercise 12.14.

Figure 12.16 show root loci for systems with $k > 0$ and the transfer functions.

$$\begin{aligned} P_a(s) &= k\frac{s+1}{s^2}, & P_b(s) &= k\frac{s+1}{s(s+2)(s^2+2s+4)} \\ P_c(s) &= k\frac{s+2}{s(s^2+1)}, & P_d(s) &= k\frac{s^2+2s+2}{s(s^2+1)}. \end{aligned} \tag{12.22}$$

The locus of $P_a(s)$ in Figure 12.16a starts with two roots at the origin and the pattern locally has the star configuration with $n^* = 2$. As the gain increases the locus bends because of the attraction of the zero. In this particular case the locus is actually a circle around the zero $s = -1$. Two roots meet at the real axis, and there is the typical star pattern. One root goes towards the zero and the other one goes to infinity along the negative real axis as the gain k increases. The root locus has the segment $(-\infty, -1]$ on the real axis. The locus in Figure 12.16b start at the open loop poles $s = -2, 0$ and $-1 \pm i\sqrt{3}$. The pole excess is $n_{pe} = 3$ and the asymptotes which originate from $s_0 = -1$, have the corresponding pattern. The locus in Figure 12.16c has vertical asymptotes since $n_{pe} = 2$, see Figure 12.15. The asymptotes originate from $s_0 = 0.5$. The root locus has the segment $[-10])$ on the real line. The locus in Figure 12.16d has three brances, one is the segment $(-\infty, 0]$ on the real line the other segment originate in the complex open loop poles and end at the open loop zeros.

The root locus is useful for qualitative arguments. For example, it follows from Figure Figure 12.15 that the closed loop system will always be unstable for sufficiently large gains if the pole excess is larger than $n_{pe} = 2$ or if the process transfer function has zeros in the right half plane.

The root locus can also be used for design. Consider for example the system in

Figure 12.16c which can represent PI control of a system with the transfer function

$$P(s) = \frac{1}{s^2+1}, \qquad C(s) = k\,\frac{s+2}{s}$$

The root locus in Figure 12.16c shows that the system is unstable for all values of the controller gain and we can immediately conclude that the process cannot be stabilized with a PI dontroller. To obtain a stable closed loop system we can attempt to choose a PID controller with zeros to the left of the undamped poles, for example

$$C(s) = k\,\frac{s^2+2s+2}{s}.$$

The root locus obtained with this controller is shown in Figure 12.16d.

We have illustrated the root locus with a closed loop system with a proportional controller where the parameter is the gain. The root locus also be used to find the effects of other parameters as was illustrated in Example 5.15.

12.6 Fundamental Limitations

Although loop shaping gives us a great deal of flexibility in designing the closed loop response of a system, there are certain fundamental limits on what can be achieved. We consider here some of the primary performance limitations that can occur because of difficult dynamics; additional limitations related to robustness are considered in the next chapter.

Right Half-Plane Poles and Zeros and Time Delays

There are linear systems that are inherently difficult to control. The limitations are related to poles and zeros in the right half-plane and time delays. To explore the limitations caused by poles and zeros in the right half-plane we factor the process transfer function as

$$P(s) = P_{mp}(s)P_{ap}(s), \tag{12.23}$$

where P_{mp} is the minimum phase part and P_{ap} is the nonminimum phase part, we require that P_{mp} has all its zeros in the open left half plane. The factorization is normalized so that $|P_{ap}(i\omega)| = 1$, and the sign is chosen so that P_{ap} has negative phase. The transfer function P_{ap} is called an *all-pass system* because it has unit gain for all frequencies. We have for example

$$P(s) = \frac{s-2}{(s+1)(s-1)} = \frac{s+2}{(s+1)^2}\,\frac{(s-2)(s+1)}{(s+2)(s-1)}. \tag{12.24}$$

The transfer function P_{ap} does not influence the gain curve in the Bode plot but it does influence the phase curve. Requiring that the phase margin be φ_m, we get

$$\arg L(i\omega_{\rm gc}) = \arg P_{ap}(i\omega_{\rm gc}) + \arg P_{mp}(i\omega_{\rm gc}) + \arg C(i\omega_{\rm gc}) \geq -\pi + \varphi_m, \tag{12.25}$$

where C is the controller transfer function. Let n_{gc} be the slope of the gain curve of the loop transfer function $L(s)$ at the crossover frequency. Since $|P_{ap}(i\omega)| = 1$, it follows that

$$n_{\text{gc}} = \left.\frac{d\log|L(i\omega)|}{d\log\omega}\right|_{\omega=\omega_{\text{gc}}} = \left.\frac{d\log|P_{mp}(i\omega)C(i\omega)|}{d\log\omega}\right|_{\omega=\omega_{\text{gc}}}.$$

If the slope n_{gc} is negative, it has to be larger than -2 for the closed loop system to be stable. It follows from Bode's relations, equation (10.8), that

$$\arg P_{mp}(i\omega) + \arg C(i\omega) \approx n_{\text{gc}}\frac{\pi}{2}.$$

Combining this with equation (12.25) gives the following inequality for the allowable phase lag of the all-pass part at the gain crossover frequency:

$$-\arg P_{ap}(i\omega_{\text{gc}}) \leq \pi - \varphi_m + n_{\text{gc}}\frac{\pi}{2} =: \varphi_l. \tag{12.26}$$

In addition we must require that the encirclement condition of the Nyquist theorem is satisfied.

The condition (12.26), which we call the *gain crossover frequency inequality*, shows that the gain crossover frequency must be chosen so that the phase lag of the nonminimum phase component is not too large. For systems with high robustness requirements we may choose a phase margin of 60° ($\varphi_m = \pi/3$). It follows from the approximate version of Bode's relations (10.8) that the slope n_{gc} cannot be steeper than $n_{\text{gc}} = -1$. To have a reasonable flexibility in choosing the gain crossover frequency we choose $n_{\text{gc}} = -1$, which gives an admissible phase lag $\varphi_l = \pi/6 = 0.52$ rad (30°) of the allpass component. For systems where we can accept a lower robustness we may choose a phase margin of 45° ($\varphi_m = \pi/4$) and the slope $n_{\text{gc}} = -1/2$, which gives an admissible phase lag $\varphi_l = \pi/2 = 1.57$ rad (90°).

The crossover frequency inequality (12.26), shows that nonminimum phase components impose severe restrictions on possible crossover frequencies. It also means that there are systems that cannot be controlled with sufficient stability margins. We illustrate the limitations in a number of commonly encountered situations.

Example 12.9 Zero in the right half-plane
The nonminimum phase part of the process transfer function for a system with a right half-plane zero is

$$P_{ap}(s) = \frac{z-s}{z+s},$$

where $z > 0$. Notice that we have $z-s$ in the numerator instead of $s-z$ to satisfy the condition that P_{ap} should have negative phase. The phase lag of the nonminimum phase part is

$$-\arg P_{ap}(i\omega) = 2\arctan\frac{\omega}{z}.$$

Since the phase lag of P_{ap} increases with frequency, the inequality (12.26) gives

the following bound on the crossover frequency:

$$\omega_{gc} < z \tan(\varphi_l/2). \tag{12.27}$$

With $\varphi_l = \pi/3$ we get $\omega_{gc} < 0.6z$. Slow right half-plane zeros (z small) therefore give tighter restrictions on possible gain crossover frequencies than fast right half-plane zeros. ∇

Time delays also impose limitations similar to those given by zeros in the right half-plane. We can understand this intuitively from the Padé approximation

$$e^{-s\tau} \approx \frac{1-0.5s\tau}{1+0.5s\tau} = \frac{2/\tau - s}{2/\tau + s}.$$

A long time delay is thus approximately equivalent to a slow right half-plane zero $z = 2/\tau$.

Example 12.10 Pole in the right half-plane
The nonminimum phase part of the transfer function for a system with a pole in the right half-plane is

$$P_{ap}(s) = \frac{s+p}{s-p},$$

where $p > 0$. The sign of P_{ap} is dictated by the condition that it should have negative pahse. The phase lag of the nonminimum phase part is

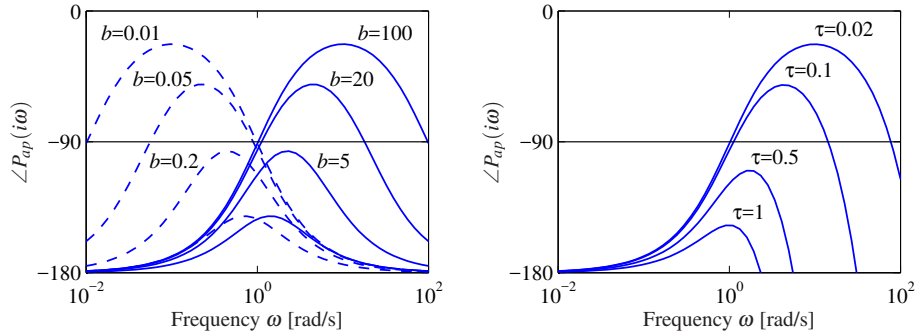
$$-\arg P_{ap}(i\omega) = 2\arctan\frac{p}{\omega},$$

and the crossover frequency inequality becomes

$$\omega_{gc} > \frac{p}{\tan(\varphi_l/2)}. \tag{12.28}$$

Right half-plane poles thus require that the closed loop system have a sufficiently high gain crossover frequency, a consequence is that the actuators must be fast. With $\varphi_l = \pi/3$ we get $\omega_{gc} > 1.7p$. Fast right half-plane poles (p large) therefore a larger gain crossover frequency than slower right half-plane poles. The control of unstable systems imposes minimum bandwidth requirements for process actuators and sensors. ∇

We will now consider systems with a right half-plane zero z and a right half-plane pole p. If $p = z$, there will be cancellation of an unstable system mode and the system cannot be stabilized. A cancellation of an unstable pole means that the system has an unstable mode that is not reachable and observable, see Section 8.3. We can therefore expect that the system is difficult to control if the right half-plane pole and zero are close. A straightforward way to use the crossover frequency inequality is to plot the phase of the nonminimum phase factor P_{ap} of the process transfer function. Such a plot, which can be incorporated in an ordinary Bode plot, will immediately show the permissible gain crossover frequencies. An illustration is given in Figure 12.17, which shows the phase of the transfer functions for systems with a right half-plane pole/zero pair and systems with a right half-plane pole



(a) RHP pole/zero pair (b) RHP pole and time delay

Figure 12.17: Example limitations due to the gain crossover frequency inequality. The figures show the phase lag of the all-pass factor P_{ap} as a function of frequency for the systems (12.29). Since the phase lag of P_{ap} at the gain crossover frequency cannot be too large, it is necessary to choose the gain crossover frequency properly. All systems have a right half-plane pole at $s = 1$. The system in (a) has zeros at $s = 2$, 5, 20 and 100 (solid lines) and at $s = 0.5$, 0.2, 0.05 and 0.01 (dashed lines). The system in (b) has time delays $\tau = 0.02$ 0.1, 0.5 and 1.

and a time delay. The transfer functions of the systems are

$$P_{ap}(s) = \frac{(bs-1)(s+1)}{(bs+1)(s-1)}, \qquad P_{ap}(s) = \frac{s+1}{s-1}e^{-\tau s} \tag{12.29}$$

To illustrate the limitations we will introduce numerical values. If we require that the phase lag φ_l of the nonminimum phase factor be less than 90°, we must require that the ratio z/p be larger than 6 or smaller than 1/6 for systems with right half-plane poles and zeros and that the product $p\tau$ be less than 0.3 for systems with a time delay and a right half-plane pole. Notice the symmetry in the problem for $z > p$ and $z < p$: in either case the zeros and the poles must be sufficiently far apart (Exercise 12.16). Also notice that possible values of the gain crossover frequency ω_{gc} are quite restricted.

Using the theory of functions of complex variables, it can be shown that for systems with a right half-plane pole p and a right half-plane zero z (or a time delay τ), any stabilizing controller gives sensitivity functions with the property

$$\sup_{\omega} |S(i\omega)| \geq \frac{p+z}{|p-z|}, \qquad \sup_{\omega} |T(i\omega)| \geq e^{p\tau}. \tag{12.30}$$

This result is proven in Exercise 12.17.

As the examples above show, right half-plane poles and zeros significantly limit the achievable performance of a system, hence one would like to avoid these whenever possible. The poles of a system depend on the intrinsic dynamics of the system and are given by the eigenvalues of the dynamics matrix A of a linear system. Sensors and actuators have no effect on the poles; the only way to change poles is to redesign the system. Notice that this does not imply that unstable systems

should be avoided. Unstable system may actually have advantages; one example is high-performance supersonic aircraft.

The zeros of a system depend on how the sensors and actuators are coupled to the states. The zeros depend on all the matrices A, B, C and D in a linear system. The zeros can thus be influenced by moving the sensors and actuators or by adding sensors and actuators.

Example 12.11 Balance system
As an example of a system with both right half-plane poles and zeros, consider the balance system with zero damping shown in Example 3.1, whose dynamics are given by

$$H_{\theta F} = \frac{ml}{-(M_t J_t - m^2 l^2)s^2 + mglM_t},$$
$$H_{pF} = \frac{-J_t s^2 + mgl}{s^2\left(-(M_t J_t - m^2 l^2)s^2 + mglM_t\right)}.$$

Assume that we want to stabilize the pendulum by using the cart position as the measured signal. The transfer function from the input force F to the cart position p has poles $\{0, 0, \pm\sqrt{mglM_t/(M_t J_t - m^2 l^2)}\}$ and zeros $\{\pm\sqrt{mgl/J_t}\}$. Using the parameters in Example 7.7, the right half-plane pole is at $p = 2.68$ and the zero is at $z = 2.09$. Equation (12.30) then gives $|S(i\omega)| \geq 8$, which shows that it is not possible to control the system robustly.

The right half-plane zero of the system can be eliminated by changing the output of the system. For example, if we choose the output to correspond to a position at a distance r along the pendulum, we have $y = p - r\sin\theta$ and the transfer function for the linearized output becomes

$$H_{y,F} = H_{pF} - rH_{\theta F} = \frac{(mlr - J_t)s^2 + mgl}{s^2\left(-(M_t J_t - m^2 l^2)s^2 + mglM_t\right)}.$$

If we choose r sufficiently large, then $mlr - J_t > 0$ and we eliminate the right half-plane zero, obtaining instead two pure imaginary zeros. The gain crossover frequency inequality is then based just on the right half-plane pole (Example 12.10). If our admissible phase lag for the nonminimum phase part is $\varphi_l = 45^\circ$, then our gain crossover must satisfy

$$\omega_{gc} > \frac{p}{\tan(\varphi_l/2)} = 6.48 \text{ rad/s}.$$

If the actuators have sufficiently high bandwidth, e.g., a factor of 10 above ω_{gc} or roughly 10 Hz, then we can provide robust tracking up to this frequency. ∇

Bode's Integral Formula

In addition to providing adequate phase margin for robust stability, a typical control design will have to satisfy performance conditions on the sensitivity functions (Gang of Four). In particular, the sensitivity function $S = 1/(1+PC)$ represents the disturbance attenuation and also relates the tracking error e to the reference signal

r: we usually want the sensitivity to be small over the range of frequencies where we want small tracking error and good disturbance attenuation. A basic problem is to investigate if S can be made small over a large frequency range. We will start by investigating an example.

Example 12.12 System that admits small sensitivities
Consider a closed loop system consisting of a first-order process and a proportional controller. Let the loop transfer function be

$$L(s) = PC = \frac{k}{s+1},$$

where parameter k is the controller gain. The sensitivity function is

$$S(s) = \frac{s+1}{s+1+k}$$

and we have

$$|S(i\omega)| = \sqrt{\frac{1+\omega^2}{1+2k+k^2+\omega^2}}.$$

This implies that $|S(i\omega)| < 1$ for all finite frequencies and that the sensitivity can be made arbitrarily small for any finite frequency by making k sufficiently large. ∇

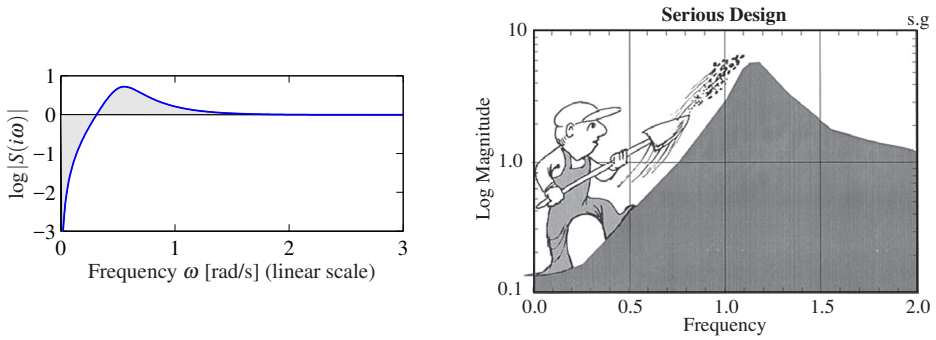
The system in Example 12.12 is unfortunately an exception. The key feature of the system is that the Nyquist curve of the process is completely contained in the right half-plane. Such systems are called *passive*, their transfer functions are *positive real* and their physics is associated with energy dissipation. For typical control systems there are severe constraints on the sensitivity function. The following theorem, due to Bode, provides insights into the limits of performance under feedback.

Theorem 12.2 (Bode's integral formula)**.** *Let $S(s)$ be the sensitivity function of an internally stable system with loop transfer function $L(s)$. Assume that the loop transfer function $L(s)$ is such that $sL(s)$ goes to zero as $s \to \infty$, then the sensitivity function satisfies the following integral:*

$$\int_0^\infty \log|S(i\omega)|\,d\omega = \int_0^\infty \log\frac{1}{|1+L(i\omega)|}\,d\omega = \pi\sum p_k. \tag{12.31}$$

where the sum is over the right half plane poles p_k of $L(s)$.

Equation (12.31) implies that there are fundamental limitations to what can be achieved by control and that control design can be viewed as a redistribution of disturbance attenuation over different frequencies. In particular, this equation shows that if the sensitivity function is made smaller for some frequencies, it must increase at other frequencies so that the integral of $\log|S(i\omega)|$ remains constant. This means that if disturbance attenuation is improved in one frequency range, it will be worse in another, a property sometime referred to as the *waterbed effect*. It also follows that systems with open loop poles in the right half-plane have larger overall sensitivity than stable systems.



(a) Bode integral formula (b) Control design process

Figure 12.18: Interpretation of the *waterbed effect*. The function $\log|S(i\omega)|$ is plotted versus ω in linear scales in (a). According to Bode's integral formula (12.31), the area of $\log|S(i\omega)|$ above zero must be equal to the area below zero. Gunter Stein's interpretation of design as a trade-off of sensitivities at different frequencies is shown in (b) (from [Ste03]).

Equation (12.31) can be regarded as a *conservation law*: if the loop transfer function has no poles in the right half-plane, the equation simplifies to

$$\int_0^\infty \log|S(i\omega)|d\omega = 0.$$

This formula can be given a nice geometric interpretation as illustrated in Figure 12.18, which shows $\log|S(i\omega)|$ as a function of ω. The area over the horizontal axis must be equal to the area under the axis when the frequency is plotted on a *linear* scale. Thus if we wish to make the sensitivity smaller up to some frequency ω_{sc}, we must balance this by increased sensitivity above ω_{sc}. Control system design can be viewed as trading the disturbance attenuation at some frequencies for disturbance amplification at other frequencies. Notice that the system in Example 12.12 violates the condition that $\lim_{s\to\infty} sL(s) = 0$ and hence the integral formula does not apply.

There is a result analogous to equation (12.31) for the complementary sensitivity function:

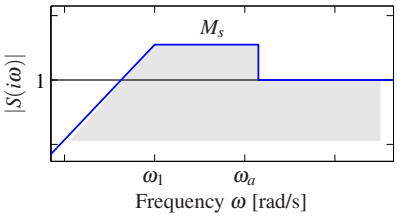
$$\int_0^\infty \frac{\log|T(i\omega)|}{\omega^2}\,d\omega = \pi\sum\frac{1}{z_i}, \tag{12.32}$$

where the summation is over all right half-plane zeros. Notice that slow right half-plane zeros are worse than fast ones and that fast right half-plane poles are worse than slow ones.

Example 12.13 X-29 aircraft

As an example of the application of Bode's integral formula, we present an analysis of the control system for the X-29 aircraft (see Figure 12.19a), which has an unusual configuration of aerodynamic surfaces that are designed to enhance its maneuverability. This analysis was originally carried out by Gunter Stein in his article "Respect the Unstable" [Ste03], which is also the source of the quote at the beginning of this chapter.

(a) X-29 aircraft



(b) Sensitivity analysis

Figure 12.19: X-29 flight control system. The aircraft makes use of forward swept wings and a set of canards on the fuselage to achieve high maneuverability (a). The desired sensitivity for the closed loop system is shown in (b). We seek to use our control authority to shape the sensitivity curve so that we have low sensitivity (good performance) up to frequency ω_1 by creating higher sensitivity up to our actuator bandwidth ω_a.

To analyze this system, we make use of a small set of parameters that describe the key properties of the system. The X-29 has longitudinal dynamics that are very similar to inverted pendulum dynamics (Exercise 9.3) and, in particular, have a pair of poles at approximately $p = \pm 6$ and a zero at $z = 26$. The actuators that stabilize the pitch have a bandwidth of $\omega_a = 40$ rad/s and the desired bandwidth of the pitch control loop is $\omega_1 = 3$ rad/s. Since the ratio of the zero to the pole is only 4.3, we may expect that it may be difficult to achieve the specifications.

To evaluate the achievable performance, we search for a control law such that the sensitivity function is small up to the desired bandwidth and not greater than M_s beyond that frequency. Because of the Bode integral formula, we know that M_s must be greater than 1 at high frequencies to balance the small sensitivity at low frequency. We thus ask if we can find a controller that has the shape shown in Figure 12.19b with the smallest value of M_s. Note that the sensitivity above the frequency ω_a is not specified since we have no actuator authority at that frequency. However, assuming that the process dynamics fall off at high frequency, the sensitivity at high frequency will approach 1. Thus, we desire to design a closed loop system that has low sensitivity at frequencies below ω_1 and sensitivity that is not too large between ω_1 and ω_a.

From Bode's integral formula, we know that whatever controller we choose, equation (12.31) must hold. We will assume that the sensitivity function is given by

$$|S(i\omega)| = \begin{cases} \frac{\omega M_s}{\omega_1} & \omega \leq \omega_1 \\ M_s & \omega_1 \leq \omega \leq \omega_a, \end{cases}$$

corresponding to Figure 12.19b. If we further assume that $|L(s)| \leq \delta/\omega^2$ for frequencies larger than the actuator bandwidth, Bode's integral becomes

$$\begin{aligned} \int_0^\infty \log|S(i\omega)|\,d\omega &= \int_0^{\omega_a} \log|S(i\omega)|\,d\omega \\ &= \int_0^{\omega_1} \log\frac{\omega M_s}{\omega_1}\,d\omega + (\omega_a - \omega_1)\log M_s = \pi p. \end{aligned}$$

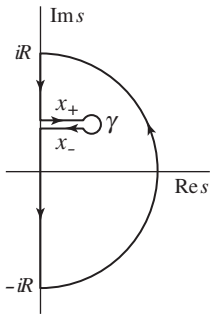


Figure 12.20: Contour used to prove Bode's theorem. For each right half-plane pole we create a path from the imaginary axis that encircles the pole as shown. To avoid clutter we have shown only one of the paths that enclose one right half-plane.

Evaluation of the integral gives $-\omega_1 + \omega_a \log M_s = \pi p$ or

$$M_s = e^{(\pi p + \omega_1)/\omega_a}.$$

This formula tells us what the achievable value of M_s will be for the given control specifications. In particular, using $p = 6$, $\omega_1 = 3$ and $\omega_a = 40$ rad/s, we find that $M_s = 1.75$, which means that in the range of frequencies between ω_1 and ω_a, disturbances at the input to the process dynamics (such as wind) will be amplified by a factor of 1.75 in terms of their effect on the aircraft.

Another way to view these results is to compute the phase margin that corresponds to the given level of sensitivity. Since the peak sensitivity normally occurs at or near the crossover frequency, we can compute the phase margin corresponding to $M_s = 1.75$. As shown in Exercise 12.18, the maximum achievable phase margin for this system is approximately 35°, which is below the usual design limit of 45° in aerospace systems. The zero at $s = 26$ limits the maximum gain crossover that can be achieved. ∇

Derivation of Bode's Formula

We now derive Bode's integral formula (Theorem 12.2). This is a technical section that requires some knowledge of the theory of complex variables, in particular contour integration. Assume that the loop transfer function has distinct poles at $s = p_k$ in the right half-plane and that $L(s)$ goes to zero faster than $1/s$ for large values of s.

Consider the integral of the logarithm of the sensitivity function $S(s) = 1/(1 + L(s))$ over the contour shown in Figure 12.20. The contour encloses the right half-plane except for the points $s = p_k$ where the loop transfer function $L(s) = P(s)C(s)$ has poles and the sensitivity function $S(s)$ has zeros. The direction of the contour is counterclockwise.

The integral of the log of the sensitivity function around this contour is given

by

$$\begin{aligned}\int_\Gamma \log(S(s))\,ds &= \int_{iR}^{-iR} \log(S(s))\,ds + \int_R \log(S(s))\,ds + \sum_k \int_\gamma \log(S(s))\,ds \\ &= I_1 + I_2 + I_3 = 0,\end{aligned}$$

where R is a large semicircle on the right and γ_k is the contour starting on the imaginary axis at $s = \operatorname{Im} p_k$ and a small circle enclosing the pole p_k. The integral is zero because the function $\log S(s)$ is analytic inside the contour. We have

$$I_1 = -i\int_{-R}^{R} \log(S(i\omega))d\omega = -2i\int_0^R \log(|S(i\omega)|)d\omega$$

because the real part of $\log S(i\omega)$ is an even function and the imaginary part is an odd function. Furthermore we have

$$I_2 = \int_R \log(S(s))\,ds = -\int_R \log(1+L(s))\,ds \approx -\int_R L(s)\,ds.$$

Since $L(s)$ goes to zero faster than $1/s$ for large s, the integral goes to zero when the radius of the circle goes to infinity.

Next we consider the integral I_3. For this purpose we split the contour into three parts X_+, γ and X_-, as indicated in Figure 12.20. We can then write the integral as

$$I_3 = \int_{X_+} \log S(s)\,ds + \int_\gamma \log S(s)\,ds + \int_{X-} \log S(s)\,ds.$$

The contour γ is a small circle with radius r around the pole p_k. The magnitude of the integrand is of the order $\log r$, and the length of the path is $2\pi r$. The integral thus goes to zero as the radius r goes to zero. Since $S(s) \approx k/(s-p_k)$ close to the pole, the argument of $S(s)$ decreases by 2π as the contour encircles the pole. On the contours X_+ and X_- we therefore have

$$|S_{X_+}| = |S_{X_-}|, \qquad \arg S_{X_-} = \arg S_{X_+} - 2\pi.$$

Hence

$$\log(S_{X_+}) - \log(S_{X_-}) = 2\pi i,$$

and we get

$$\int_{X_+} \log S(s)\,ds + \int_{X_-} \log S(s)\,ds = 2\pi\, i \operatorname{Re} p_k.$$

Repeating the argument for all poles p_k in the right half plane, letting the small circles go to zero gives

$$I_1 + I_2 + I_3 = -2i\int_0^\infty \log|S(i\omega)|d\omega + i\sum_k 2\pi \operatorname{Re} p_k = 0.$$

Since complex poles appear as complex conjugate pairs, $\sum_k \operatorname{Re} p_k = \sum_k p_k$, which gives Bode's formula (12.31).

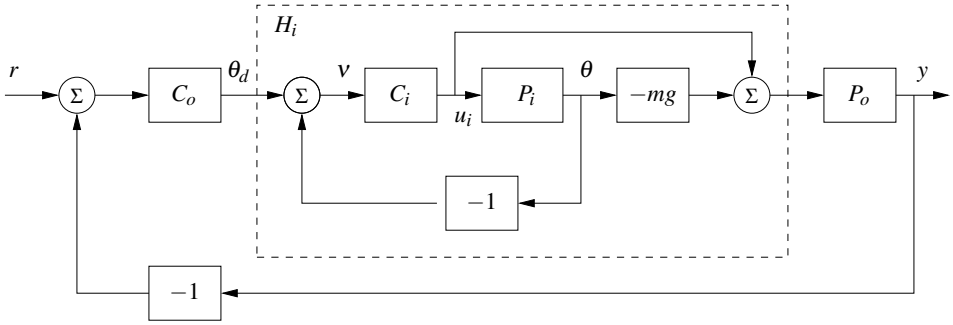


Figure 12.21: Inner/outer control design for a vectored thrust aircraft. The inner loop H_i controls the roll angle of the aircraft using the vectored thrust. The outer loop controller C_o commands the roll angle to regulate the lateral position. The process dynamics are decomposed into inner loop (P_i) and outer loop (P_o) dynamics, which combine to form the full dynamics for the aircraft.

.

12.7 Design Example

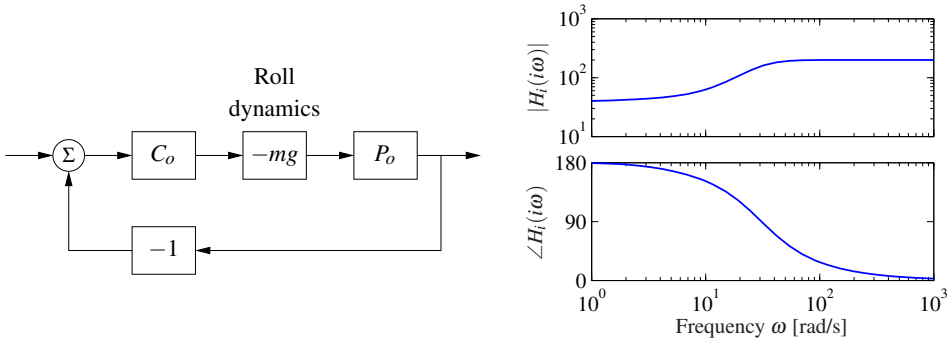
In this section we present a detailed example that illustrates the main design techniques described in this chapter.

Example 12.14 Lateral control of a vectored thrust aircraft
The problem of controlling the motion of a vertical takeoff and landing (VTOL) aircraft was introduced in Example 3.11 and in Example 12.8, where we designed a controller for the roll dynamics. We now wish to control the position of the aircraft, a problem that requires stabilization of the attitude. The system thus has two control loops.

To control the lateral dynamics of the vectored thrust aircraft, we make use of a "inner/outer" loop design methodology, as illustrated in Figure 12.21. This diagram shows the process dynamics and controller divided into two components: an *inner loop* consisting of the roll dynamics and control and an *outer loop* consisting of the lateral position dynamics and controller. This decomposition follows the block diagram representation of the dynamics given in Exercise 9.10.

The approach that we take is to design a controller C_i for the inner loop so that the resulting closed loop system H_i assures that the roll angle θ follows its reference θ_r fast and accurately. We then design a controller for the lateral position y that uses the approximation that we can directly control the roll angle as an input θ to the dynamics controlling the position. Under the assumption that the dynamics of the roll controller are fast relative to the desired bandwidth of the lateral position control, we can then combine the inner and outer loop controllers to get a single controller for the entire system. As a performance specification for the entire system, we would like to have zero steady-state error in the lateral position, a bandwidth of approximately 1 rad/s and a phase margin of 45°.

For the inner loop, we choose our design specification to provide the outer loop



(a) Outer loop approximation (b) Actual roll dynamics

Figure 12.22: Outer loop control design for a vectored thrust aircraft. (a) The outer loop approximates the roll dynamics as a state gain $-mg$. (b) The Bode plot for the roll dynamics, indicating that this approximation is accurate up to approximately 10 rad/s.

with accurate and fast control of the roll. The inner loop dynamics are given by

$$P_i = H_{\theta u_1} = \frac{r}{Js^2}.$$

We choose the desired bandwidth to be 10 rad/s (10 times that of the outer loop) and the low-frequency error to be no more than 5%. This specification is satisfied using the lead compensator of Example 12.8 designed previously, so we choose

$$C_i(s) = k\frac{s+a}{s+b}, \qquad a = 2, \quad b = 50, \quad k = 1.$$

The closed loop dynamics for the system satisfy

$$H_i = \frac{C_i}{1+C_iP_i} - mg\frac{C_iP_i}{1+C_iP_i} = \frac{C_i(1-mgP_i)}{1+C_iP_i}.$$

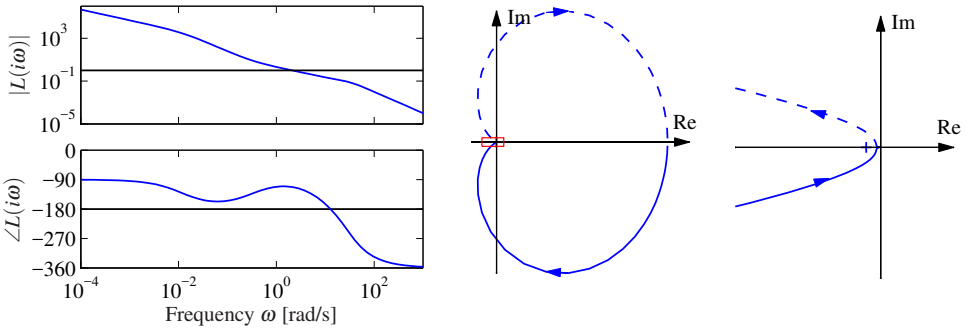
A plot of the magnitude of this transfer function is shown in Figure 12.22, and we see that $H_i \approx -mg = -39.2$ is a good approximation up to 10 rad/s.

To design the outer loop controller, we assume the inner loop roll control is perfect, so that we can take θ_d as the input to our lateral dynamics. Following the diagram shown in Exercise 9.10, the outer loop dynamics can be written as

$$P(s) = H_i(0)P_o(s) = \frac{H_i(0)}{ms^2+cs},$$

where we replace $H_i(s)$ with $H_i(0)$ to reflect our approximation that the inner loop will eventually track our commanded input. Of course, this approximation may not be valid, and so we must verify this when we complete our design.

Our control goal is now to design a controller that gives zero steady-state error in y for a step input and has a bandwidth of 1 rad/s. The outer loop process dynamics are given by a double integrator, and we can again use a simple lead compensator to satisfy the specifications. We also choose the design such that the loop transfer function for the outer loop has $|L_o| < 0.1$ for $\omega > 10$ rad/s, so that



(a) Bode plot (b) Nyquist plot

Figure 12.23: Inner/outer loop controller for a vectored thrust aircraft. Bode plot (a) and Nyquist plot (b) for the loop transfer function cut at θ_d, for the complete system. The system has a phase margin of 68° and a gain margin of 6.2.

the H_i high frequency dynamics can be neglected. We choose the controller to be of the form

$$C_o(s) = -k_o \frac{s+a_o}{s+b_o},$$

with the negative sign to cancel the negative sign in the process dynamics. To find the location of the poles, we note that the phase lead flattens out at approximately $b_o/10$. We desire phase lead at crossover, and we desire the crossover at $\omega_{gc} = 1$ rad/s, so this gives $b_o = 10$. To ensure that we have adequate phase lead, we must choose a_o such that $b_o/10 < 10a_o < b_o$, which implies that a_o should be between 0.1 and 1. We choose $a_o = 0.3$. Finally, we need to set the gain of the system such that at crossover the loop gain has magnitude 1. A simple calculation shows that $k_o = 2$ satisfies this objective. Thus, the final outer loop controller becomes

$$C_o(s) = -2\frac{s+0.3}{s+10}.$$

Finally, we can combine the inner and outer loop controllers and verify that the system has the desired closed loop performance. The Bode and Nyquist plots corresponding to Figure 12.21 with inner and outer loop controllers are shown in Figure 12.23, and we see that the specifications are satisfied. In addition, we show the Gang of Four in Figure 12.24, and we see that the transfer functions between all inputs and outputs are reasonable. The sensitivity to load disturbances PS is large at low frequency because the controller does not have integral action.

The approach of splitting the dynamics into an inner and an outer loop is common in many control applications and can lead to simpler designs for complex systems. Indeed, for the aircraft dynamics studied in this example, it is very challenging to directly design a controller from the lateral position y to the input u_1. The use of the additional measurement of θ greatly simplifies the design because it can be broken up into simpler pieces. ∇

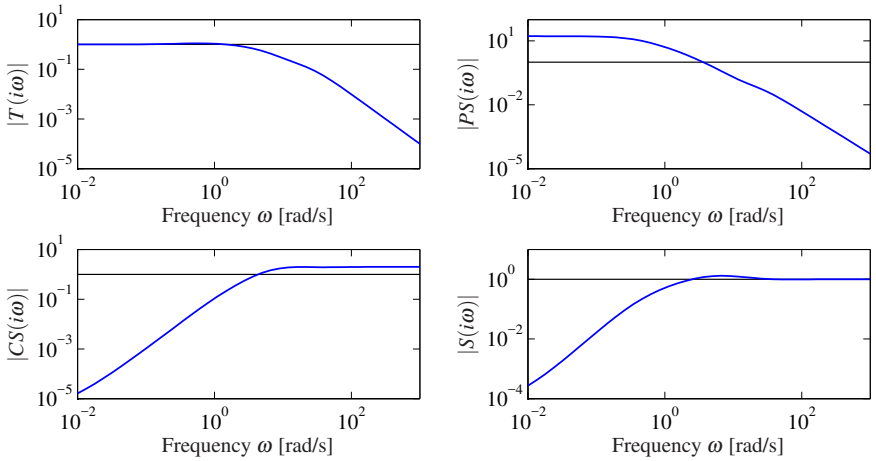


Figure 12.24: Gang of Four for vectored thrust aircraft system.

12.8 Further Reading

Design by loop shaping was a key element in the early development of control, and systematic design methods were developed; see James, Nichols and Phillips [JNP47], Chestnut and Mayer [CM51], Truxal [Tru55] and Thaler [Tha89]. Loop shaping is also treated in standard textbooks such as Franklin, Powell and Emami-Naeini [FPEN05], Dorf and Bishop [DB04], Kuo and Golnaraghi [KG02] and Ogata [Oga01]. Systems with two degrees of freedom were developed by Horowitz [Hor63], who also discussed the limitations of poles and zeros in the right half-plane. Fundamental results on limitations are given in Bode [Bod45]; more recent presentations are found in Goodwin, Graebe and Salgado [GGS01]. The treatment in Section 12.6 is based on [Åst00]. Much of the early work was based on the loop transfer function; the importance of the sensitivity functions appeared in connection with the development in the 1980s that resulted in H_∞ design methods. A compact presentation is given in the texts by Doyle, Francis and Tannenbaum [DFT92] and Zhou, Doyle and Glover [ZDG96]. Loop shaping was integrated with the robust control theory in McFarlane and Glover [MG90] and Vinnicombe [Vin01]. Comprehensive treatments of control system design are given in Maciejowski [Mac89] and Goodwin, Graebe and Salgado [GGS01].

Exercises

12.1 Consider the system in Figure 12.1. Give all signal pairs that are related by the transfer functions $1/(1+PC)$, $P/(1+PC)$, $C/(1+PC)$ and $PC/(1+PC)$.

12.2 Consider the system in Example 12.1, where the process and controller transfer fucctions are given by

$$P(s) = 1/(s-a), \qquad C(s) = k(s-a)/s.$$

Choose the parameters $a = -1$ and compute the time (step) and frequency responses for all the transfer functions in the Gang of Four for controllers with $k = 0.2$ and $k = 5$.

12.3 (Equivalence of Figures 12.1 and 12.2) Consider the system in Figure 12.1 and let the outputs of interest be $z = (\eta, \nu)$ and the major disturbances be $w = (n, d)$. Show that the system can be represented by Figure 12.2 and give the matrix transfer functions $\mathscr{P}$ and $\mathscr{C}$. Verify that the elements of the closed loop transfer function H_{zw} are the Gang of Four.

12.4 Consider the spring–mass system given by (3.17), which has the transfer function

$$P(s) = \frac{1}{ms^2 + cs + k}.$$

Design a feedforward compensator that gives a response with critical damping ($\zeta = 1$).

12.5 (Sensitivity of feedback and feedforward) Consider the system in Figure 12.1 and let G_{yr} be the transfer function relating the measured signal y to the reference r. Show that the sensitivities of G_{yr} with respect to the feedforward and feedback transfer functions F and C are given by $dG_{yr}/dF = CP/(1+PC)$ and $dG_{yr}/dC = FP/(1+PC)^2 = G_{yr}S/C$.

12.6 (Equivalence of controllers with two degrees of freedom) Show that the systems in Figures 12.1 and 12.3 give the same responses to command signals if $F_m C + F_u = CF$.

12.7 (Disturbance attenuation) Consider the feedback system shown in Figure 12.1. Assume that the reference signal is constant. Let y_{ol} be the measured output when there is no feedback and y_{cl} be the output with feedback. Show that $Y_{cl}(s) = S(s)Y_{ol}(s)$, where S is the sensitivity function.

12.8 (Disturbance reduction through feedback) Consider a problem in which an output variable has been measured to estimate the potential for disturbance attenuation by feedback. Suppose an analysis shows that it is possible to design a closed loop system with the sensitivity function

$$S(s) = \frac{s}{s^2 + s + 1}.$$

Estimate the possible disturbance reduction when the measured disturbance is

$$y(t) = 5\sin(0.1t) + 3\sin(0.17t) + 0.5\cos(0.9t) + 0.1t.$$

12.9 Show that the effect of high frequency measurement noise on the control signal for the system in Example 12.6 can be approximated by

$$CS \approx C = \frac{k_d s}{(sT_f)^2/2 + sT_f + 1},$$

and that the largest value of $|CS(i\omega)|$ is k_d/T_f which occurs for $\omega = \sqrt{2}/T_f$.

12.10 (Attenuation of low-frequency sinusoidal disturbances) Integral action eliminates constant disturbances and reduces low-frequency disturbances because the controller gain is infinite at zero frequency. A similar idea can be used to reduce the effects of sinusoidal disturbances of known frequency ω_0 by using the controller

$$C(s) = k_p + \frac{k_s s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}.$$

This controller has the gain $C_s(i\omega_0) = k_p + k_s/(2\zeta)$ for the frequency ω_0, which can be large by choosing a small value of ζ. Assume that the process has the transfer function $P(s) = 1/s$. Determine the Bode plot of the loop transfer function and simulate the system. Compare the results with PI control.

12.11 (Lead-lag compensation) Lead and lag compensators can be combined into a lead-lag compensator which has the transfer function

$$C(s) = k\frac{(s+a_1)(s+a_2)}{(s+b_1)(s+b_2)}$$

Show that the controller reduces to a PID controller with special choice of parameters and give the relations between the parameters.

12.12 (Asymptotes of root locus) Consider proportional control of a system with the transfer function

$$P(s) = \frac{b(s)}{a(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \cdots b_m}{s^n + a_1 s^{n-1} + \cdots a_n} = b_0 \frac{(s-z_1)(s-z_2)\ldots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}.$$

Show that the root locus has asymptotes that are straight line that emerge from the point

$$s_0 = \frac{1}{n_e}\Big(\sum_{k=1}^{n} p_k - \sum_{k=1}^{m} z_k\Big),$$

where $n_e = n - m$ is the pole excess of the transfer function.

12.13 (Real line segments of root locus) Consider proportional control a process with a rational transfer function. Assume that $b_o k > 0$, show that the root locus has segments on the real line there are an odd number of real poles and zeros to the right the segment.

12.14 (Initial direction of root locus) Consider proportional control of a system with the transfer function

$$P(s) = \frac{b(s)}{a(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \cdots b_m}{s^n + a_1 s^{n-1} + \cdots a_n} = b_0 \frac{(s-z_1)(s-z_2)\ldots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}.$$

Let p_j be an isolated pole and assume that $kb_p > 0$. Show that the root locus starting at p_j has the initial direction.

$$\angle(s-p_j) = \pi + \Sigma_{k=1}^{m}\angle(p_j - s_k) - \Sigma_{k\neq j}\angle(p_j - p_k).$$

Give a geometric interpretation of the result.

12.15 Consider a lead compensator with the transfer function

$$C_n(s) = \left(\frac{s\sqrt[n]{k}+a}{s+a}\right)^n,$$

which has zero frequency gain $C(0) = 1$ and high-frequency gain $C(\infty) = k$. Show that the gain required to give a given phase lead φ is

$$k = \left(1 + 2\tan^2(\varphi/n) + 2\tan(\varphi/n)\sqrt{1+\tan^2(\varphi/n)}\right)^n,$$

and that $\lim_{n\to\infty} k = e^{2\varphi}$.

12.16 Consider a process with the loop transfer function

$$L(s) = k\frac{z-s}{s-p},$$

with positive z and p. Show that the system is stable if $p/z < k < 1$ or $1 < k < p/z$, and that the largest stability margin is $s_m = |p-z|/(p+z)$ is obtained for $k = 2p/(p+z)$. Determine the pole/zero ratios that gives the stability margin $s_m = 2/3$.

12.17 Prove the inequalities given by equation (12.30). (Hint: Use the maximum modulus theorem.)

12.18 (Phase margin formulas) Show that the relationship between the phase margin and the values of the sensitivity functions at gain crossover is given by

$$|S(i\omega_{gc})| = |T(i\omega_{gc})| = \frac{1}{2\sin(\varphi_m/2)}.$$

12.19 (Stabilization of an inverted pendulum with visual feedback) Consider stabilization of an inverted pendulum based on visual feedback using a video camera with a 50-Hz frame rate. Let the effective pendulum length be l. Assume that we want the loop transfer function to have a slope of $n_{gc} = -1/2$ at the crossover frequency. Use the gain crossover frequency inequality to determine the minimum length of the pendulum that can be stabilized if we desire a phase margin of 45°.

12.20 (Rear-steered bicycle) Consider the simple model of a bicycle in Equation (4.5), which has one pole in the right half-plane. The model is also valid for a bicycle with rear wheel steering, but the sign of the velocity is then reversed and the system also has a zero in the right half-plane. Use the results of Exercise 12.16 to give a condition on the physical parameters that admits a controller with the stability margin s_m.

12.21 Prove the formula (12.32) for the complementary sensitivity.

Chapter Thirteen
Robust Performance

However, by building an amplifier whose gain is deliberately made, say 40 decibels higher than necessary (10000 fold excess on energy basis), and then feeding the output back on the input in such a way as to throw away that excess gain, it has been found possible to effect extraordinary improvement in constancy of amplification and freedom from non-linearity.

Harold S. Black, "Stabilized Feedback Amplifiers," 1934 [Bla34].

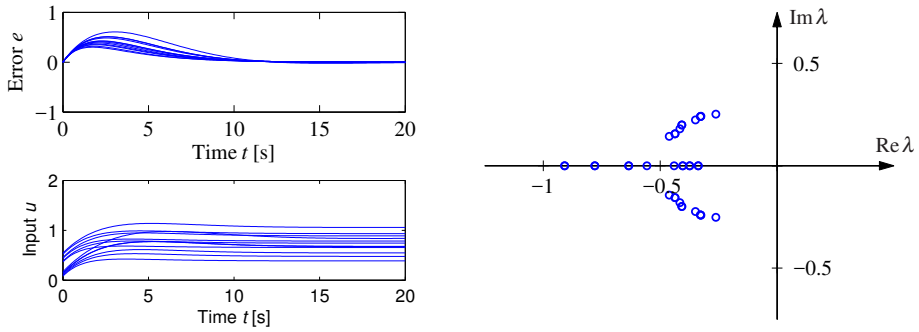
This chapter focuses on the analysis of robustness of feedback systems, a vast topic for which we provide only an introduction to some of the key concepts. We consider the stability and performance of systems whose process dynamics are uncertain and derive fundamental limits for robust stability and performance. To do this we develop ways to describe uncertainty, both in the form of parameter variations and in the form of neglected dynamics. We also briefly mention some methods for designing controllers to achieve robust performance.

13.1 Modeling Uncertainty

Harold Black's quote above illustrates that one of the key uses of feedback is to provide robustness to uncertainty ("constancy of amplification"). It is one of the most useful properties of feedback and is what makes it possible to design feedback systems based on strongly simplified models.

One form of uncertainty in dynamical systems is *parametric uncertainty* in which the parameters describing the system are not precisely known. A typical example is the variation of the mass of a car, which changes with the number of passengers and the weight of the baggage. When linearizing a nonlinear system, the parameters of the linearized model also depend on the operating conditions. It is straightforward to investigate the effects of parametric uncertainty simply by evaluating the performance criteria for a range of parameters. Such a calculation reveals the consequences of parameter variations. We illustrate by a simple example.

Example 13.1 Cruise control
The cruise control problem is described in Section 4.1, and a PI controller was designed in Example 11.3. To investigate the effect of parameter variations, we will choose a controller designed for a nominal operating condition corresponding to mass $m = 1600$ kg, fourth gear ($\alpha = 12$) and speed $v_e = 25$ m/s; the controller gains are $k_p = 0.72$ and $k_i = 0.18$. Figure 13.1a shows the velocity error e and the throttle u when encountering a hill with a 3° slope with masses in the range



(a) Disturbance response (b) Closed loop eigenvalues

Figure 13.1: Responses of the cruise control system to a slope increase of 3° (a) and the eigenvalues of the closed loop system (b). Model parameters are swept over a wide range. The closed loop system is of second order.

$1600 < m < 2000$ kg, gear ratios 3–5 ($\alpha = 10$, 12 and 16) and velocity $10 \leq v \leq 40$ m/s. The simulations were done using models that were linearized around the different operating conditions. The figure shows that there are variations in the response but that they are quite reasonable. The largest velocity error is in the range of 0.2–0.6 m/s, and the settling time is about 15 s. The control signal is marginally larger than 1 in some cases, which implies that the throttle is fully open. A full nonlinear simulation using a controller with windup protection is required if we want to explore these cases in more detail. The closed loop system has two eigenvalues Figure 13.1bb shows these eigenvalues for the different operating conditions. The figure shows that the closed loop system is well damped in all cases. ∇

This example indicates that at least as far as parametric variations are concerned, the design based on a simple nominal model will give satisfactory control. The example also indicates that a controller with fixed parameters can be used in all cases. Notice that we have not considered operating conditions in low gear and at low speed, but cruise controllers are not typically used in these cases.

Unmodeled Dynamics

It is generally easy to investigate the effects of parametric variations. However, there are other uncertainties that also are important, as discussed at the end of Section 3.3. The simple model of the cruise control system captures only the dynamics of the forward motion of the vehicle and the torque characteristics of the engine and transmission. It does not, for example, include a detailed model of the engine dynamics (whose combustion processes are extremely complex) or the slight delays that can occur in modern electronically controlled engines (as a result of the processing time of the embedded computers). These neglected mechanisms are called *unmodeled dynamics*.

Unmodeled dynamics can be accounted for by developing a more complex model. Such models are commonly used for controller development, but substan-

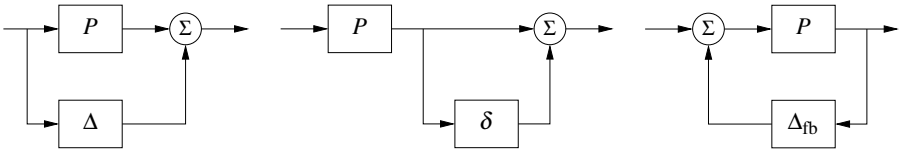


Figure 13.2: Unmodeled dynamics in linear systems. Uncertainty can be represented using additive perturbations (left), multiplicative perturbations (middle) or feedback perturbations (right). The nominal system is P, and Δ, δ and Δ_{fb} represent unmodeled dynamics.

tial effort is required to develop them. An alternative is to investigate if the closed loop system is sensitive to generic forms of unmodeled dynamics. The basic idea is to describe the unmodeled dynamics by including a transfer function in the system description whose frequency response is bounded but otherwise unspecified. For example, we might model the engine dynamics in the cruise control example as a system that quickly provides the torque that is requested through the throttle, giving a small deviation from the simplified model, which assumed the torque response was instantaneous. This technique can also be used in many instances to model parameter variations, allowing a quite general approach to uncertainty management.

In particular, we wish to explore if additional linear dynamics may cause difficulties. A simple way is to assume that the transfer function of the process is $P(s)+\Delta$, where $P(s)$ is the nominal simplified transfer function and Δ represents the unmodeled dynamics in terms of *additive uncertainty*. Different representations of uncertainty are shown in Figure 13.2. The relations between the different representations of unmodeled dynamics are

$$\delta = \frac{\Delta}{P}, \quad \Delta_{fb} = -\frac{\Delta}{P(P+\Delta)} = -\frac{\delta}{P(1+\delta)}.$$

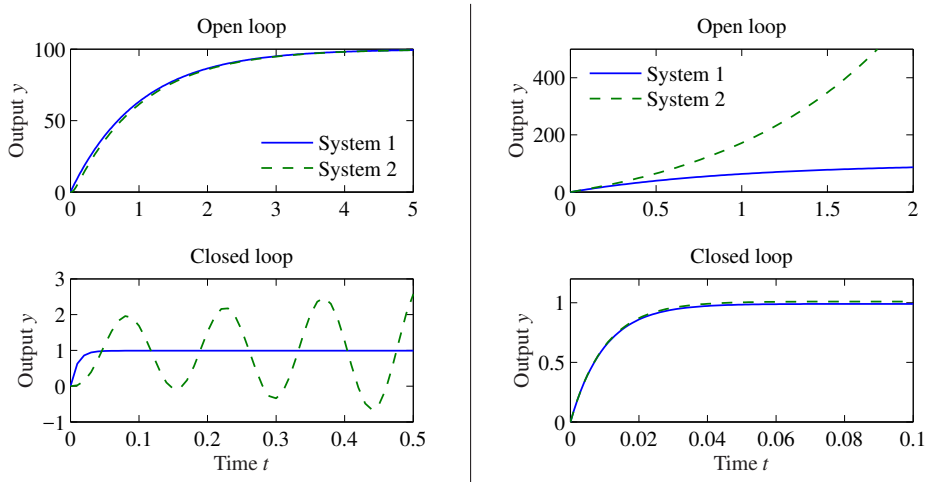
When Are Two Systems Similar? The Vinnicombe Metric

A fundamental issue in describing robustness is to determine when two systems are close. Given such a characterization, we can then attempt to describe robustness according to how close the actual system must be to the model in order to still achieve the desired levels of performance. This seemingly innocent problem is not as simple as it may appear. A naive approach is to say that two systems are close if their open loop responses are close. Even if this appears natural, there are complications, as illustrated by the following examples.

Example 13.2 Similar in open loop but large differences in closed loop
The systems with the transfer functions

$$P_1(s) = \frac{k}{s+1}, \qquad P_2(s) = \frac{k}{(s+1)(sT+1)^2} \tag{13.1}$$

have very similar open loop step responses for small values of T, as illustrated in the top plot in Figure 13.3a, which is plotted for $T = 0.025$ and $k = 100$. The dif-



(a) Example 13.2 (b) Example 13.3

Figure 13.3: Determining when two systems are close. The plots in (a) show a situation when the open loop responses are almost identical, but the closed loop responses are very different. The processes are given by equation (13.1) with $k = 100$ and $T = 0.025$. The plots in (b) show the opposite situation: the systems are different in open loop but similar in closed loop. The processes are given by equation (13.2) with $k = 100$.

ferences between the step responses are barely noticeable in the figure. The step responses with unit gain error feedback are shown in the bottom plot in Figure 13.3a. Notice that one closed loop system is stable and the other one is unstable. ∇

Example 13.3 Different in open loop but similar in closed loop
Consider the systems

$$P_1(s) = \frac{k}{s+1}, \qquad P_2(s) = \frac{k}{s-1}. \tag{13.2}$$

The open loop responses are very different because P_1 is stable and P_2 is unstable, as shown in the top plot in Figure 13.3b. Closing a feedback loop with unit gain around the systems, we find that the closed loop transfer functions are

$$T_1(s) = \frac{k}{s+k+1}, \qquad T_2(s) = \frac{k}{s+k-1},$$

which are very close for large k, as shown in Figure 13.3b. ∇

These examples show that if our goal is to close a feedback loop, it may be very misleading to compare the open loop responses of the system.

Inspired by these examples we introduce the *Vinnicombe metric*, which is a distance measure that is appropriate for closed loop systems. Consider two systems with the transfer functions P_1 and P_2, and define

$$d(P_1, P_2) = \sup_{\omega} \frac{|P_1(i\omega) - P_2(i\omega)|}{\sqrt{(1+|P_1(i\omega)|^2)(1+|P_2(i\omega)|^2)}}, \tag{13.3}$$

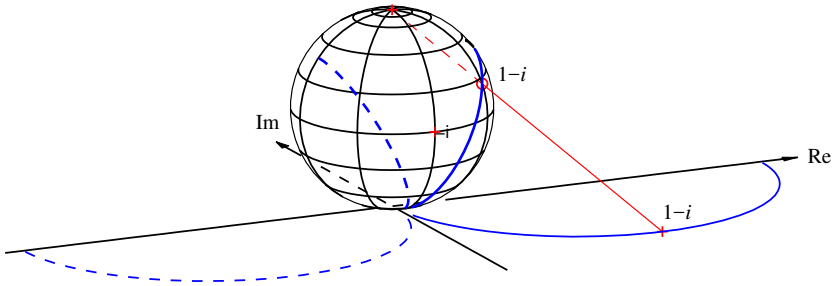


Figure 13.4: Geometric interpretation of $d(P_1,P_2)$. At each frequency, the points on the Nyquist curve for P_1 (solid) and P_2 (dashed) are projected onto a sphere of radius 1 sitting at the origin of the complex plane. The projection of the point $1-i$ is shown. The distance between the two systems is defined as the maximum distance between the projections of $P_1(i\omega)$ and $P_2(i\omega)$ over all frequencies ω. The figure is plotted for the transfer functions $P_1(s) = 2/(s+1)$ and $P_2(s) = 2/(s-1)$. (Diagram courtesy G. Vinnicombe.)

which is a metric with the property $0 \le d(P_1,P_2) \le 1$. The number $d(P_1,P_2)$ can be interpreted as the difference between the complementary sensitivity functions for the closed loop systems that are obtained with unit feedback around P_1 and P_2; see Exercise 13.3. The metric also has a nice geometric interpretation, as shown in Figure 13.4, where the Nyquist plots of P_1 and P_2 are projected onto a sphere with radius 1 sitting at the origin of the complex plane (called the *Riemann sphere*). Points in the complex plane are projected onto the sphere by a line through the point and the north pole (Figure 13.4). The distance $d(P_1,P_2)$ is the longest chordal distance between the projections of $P_1(i\omega)$ and $P_2(i\omega)$. The distance is small when $P_1(i\omega)$ and $P_2(i\omega)$ are both small or both large.

The distance $d(P_1,P_2)$ has one drawback for the purpose of comparing the behavior of systems under feedback. If P_2 is perturbed continuously from P_1 to P_2, there can be intermediate transfer functions P where $d(P_1,P)$ is 1 even if $d(P_1,P_2)$ is small (see Exercise 13.4). To explore when this could happen, we observe that

$$1 - d^2(P_1,P) = \frac{(1+P(i\omega)P_1(-i\omega))(1+P(-i\omega)P_1(i\omega))}{(1+|P_1(i\omega)|^2)(1+|P(i\omega)|^2)}.$$

The right-hand side is zero, and hence $d(P_1,P) = 1$ if $1 + P(i\omega)P_1(-i\omega) = 0$ for some ω.

Some technical conditions are required to avoid the difficulty. Vinnicombe [Vin01] introduced the set $\mathscr{C}$ of all pairs (P_1,P_2) such that the functions $f_1 = 1+P_1(s)P_1(-s)$ and $f_2 = 1+P_2(s)P_1(-s)$ have the same number of zeros in the right half-plane. He then defined

$$\delta_\nu(P_1,P_2) = \begin{cases} d(P_1,P_2), & \text{if } (P_1,P_2) \in \mathscr{C} \\ 1, & \text{otherwise.} \end{cases}, \tag{13.4}$$

which we call the *Vinnicombe metric* or *ν-gap metric*. Vinnicombe showed that $\delta_\nu(P_1,P_2)$ is a metric, gave strong robustness results based on the metric, and de-

veloped the theory for systems with many inputs and many outputs. We illustrate its use by computing the metric for the systems in the previous examples.

Example 13.4 Vinnicombe metric for Examples 13.2 and 13.3
For the systems in Example 13.2 we have

$$f_1(s) = 1 + P_1(s)P_1(-s) = \frac{1+k^2-s^2}{1-s^2},$$
$$f_2(s) = 1 + P_2(s)P_1(-s) = \frac{1+k^2+2sT+(T^2-1)s^2-2s^3T-s^4T^2}{(1-s^2)(1+2sT+s^2T^2)}.$$

The function f_1 has one zero in the right half-plane. A numerical calculation for $k = 100$ and $T = 0.025$ shows that the function f_2 has the roots 46.3, -86.3, $-20.0 \pm 60.0i$. Both functions have one zero in the right half-plane, allowing us to compute the norm (13.4). For $T = 0.025$ this gives $\delta_v(P_1, P_2) = 0.98$, which is a quite large value. To have reasonable robustness Vinnicombe recommended values less than 1/3.

For the system in Example 13.3 we have

$$1 + P_1(s)P_1(-s) = \frac{1+k^2-s^2}{1-s^2}, \qquad 1 + P_2(s)P_1(-s) = \frac{1-k^2-2s+s^2}{(s+1)^2}$$

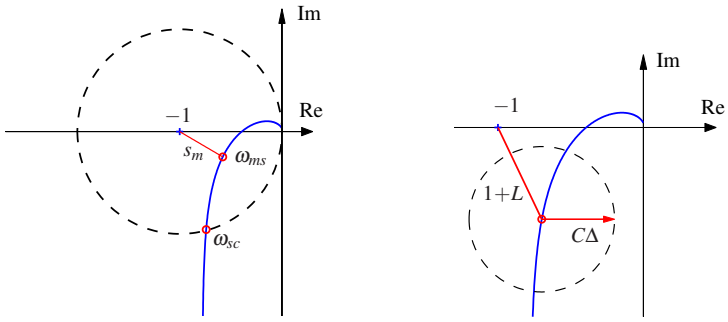
These functions have the same number of zeros in the right half-plane if $k > 1$. In this particular case the Vinnicombe metric is $d(P_1, P_2) = 2k/(1+k^2)$ (Exercise 13.4) and with $k = 100$ we get $\delta_v(P_1, P_2) = 0.02$. Figure 13.4 shows the Nyquist curves and their projections for $k = 2$. Notice that $d(P_1, P_2)$ is very small for small k even though the closed loop systems are very different. It is therefore essential to consider the condition $(P_1, P_2) \in \mathscr{C}$, as discussed in Exercise 13.4. ∇

13.2 Stability in the Presence of Uncertainty

Having discussed how to describe uncertainty and the similarity between two systems, we now consider the problem of robust stability: When can we show that the stability of a system is robust with respect to process variations? This is an important question since the potential for instability is one of the main drawbacks of feedback. Hence we want to ensure that even if we have small inaccuracies in our model, we can still guarantee stability and performance of the closed loop system.

Robust Stability Using Nyquist's Criterion

The Nyquist criterion provides a powerful and elegant way to study the effects of uncertainty for linear systems. A simple criterion is that the Nyquist curve be sufficiently far from the critical point -1. Recall that the shortest distance from the Nyquist curve to the critical point is $s_m = 1/M_s$, where M_s is the maximum of the sensitivity function and s_m is the stability margin introduced in Section 10.3.



(a) Nyquist plot (b) Additive uncertainty

Figure 13.5: Robust stability using the Nyquist criterion. The plot (a) shows that the shortest distance to the critical point s_m is a robustness measure. The plot (b) shows the Nyquist curve of the nominal loop transfer function L, the circle shows its uncertainty due to additive process variations Δ.

The maximum sensitivity M_s or the stability margin s_m is thus a good robustness measure, as illustrated in Figure 13.5a.

We will now derive explicit conditions on the controller C such that stability i guaranteed for process perturbations where $|Delta|$ is less than a given bound. Consider a stable feedback system with a process P and a controller C. If the process is changed from P to $P+\Delta$, the loop transfer function changes from PC to $PC+C\Delta$, as illustrated in Figure 13.5b. If we have a bound on the size of Δ (represented by the dashed circle in the figure), then the system remains stable as long the perturbed loop tansfer funcion $|1+(P+\Delta)C|$ never reaches the critical point -1 point, since the number of encirclements of -1 remain unchanged.

Some additional assumptions are required for the analysis to hold. Most importantly, we require that the process perturbations Δ be stable so that we do not introduce any new right half-plane poles that would require additional encirclements in the Nyquist criterion.

We will now compute an analytical bound on the allowable process disturbances. The distance from the critical point -1 to the loop transfer function L is $|1+L|$. This means that the perturbed Nyquist curve will not reach the critical point -1 provided that $|C\Delta| < |1+L|$, which is guaranteed if

$$|\Delta| < \left|\frac{1+PC}{C}\right| \qquad \text{or} \qquad |\delta| < \frac{1}{|T|}, \quad \text{where} \quad \delta := \left|\frac{\Delta}{P}\right|. \tag{13.5}$$

This condition must be valid for all points on the Nyquist curve, i.e, pointwise for all frequencies. The condition for robust stability can thus be written as

$$|\delta(i\omega)| = \left|\frac{\Delta(i\omega)}{P(i\omega)}\right| < \frac{1}{|T(i\omega)|} \qquad \text{for all } \omega \geq 0. \tag{13.6}$$

Notice that the condition is conservative because it follows from Figure 13.5 that the critical perturbation is in the direction toward the critical point -1. Larger

perturbations can be permitted in the other directions.

The condition in equation (13.6) allows us to reason about uncertainty without exact knowledge of the process perturbations. Namely, we can verify stability for *any* uncertainty Δ that satisfies the given bound. From an analysis perspective, this gives us a measure of the robustness for a given design. Conversely, if we require robustness of a given level, we can attempt to choose our controller C such that the desired level of robustness is available (by asking that T be small) in the appropriate frequency bands.

Equation (13.6) is one of the reasons why feedback systems work so well in practice. The mathematical models used to design control systems are often simplified, and the properties of a process may change during operation. Equation (13.6) implies that the closed loop system will at least be stable for substantial variations in the process dynamics.

It follows from equation (13.6) that the variations can be large for those frequencies where T is small and that smaller variations are allowed for frequencies where T is large. A conservative estimate of permissible process variations that will not cause instability is given by

$$|\delta(i\omega)| = \left|\frac{\Delta(i\omega)}{P(i\omega)}\right| < \frac{1}{M_t},$$

where M_t is the largest value of the complementary sensitivity

$$M_t = \sup_{\omega} |T(i\omega)| = \left\| \frac{PC}{1+PC} \right\|_{\infty}. \tag{13.7}$$

Reasonable values of M_t are in the range of 1.2 to 2. It is shown in Exercise 13.5 that if $M_t = 2$ then pure gain variations of 50% or pure phase variations of 30° are permitted without making the closed loop system unstable.

Example 13.5 Cruise control

Consider the cruise control system discussed in Section 4.1. The model of the car in fourth gear at speed 25 m/s is

$$P(s) = \frac{1.38}{s+0.0142},$$

and the controller is a PI controller with gains $k_p = 0.72$ and $k_i = 0.18$. Figure 13.6 plots the allowable size of the process uncertainty using the bound in equation (13.6). At low frequencies, $T(0) = 1$ and so the perturbations can be as large as the original process ($|\delta| = |\Delta/P| < 1$). The complementary sensitivity has its maximum $M_t = 1.14$ at $\omega_{mt} = 0.35$, and hence this gives the minimum allowable process uncertainty, with $|\delta| < 0.87$ or $|\Delta| < 3.47$. Finally, at high frequencies, $T \to 0$ and hence the relative error can get very large. For example, at $\omega = 5$ we have $|T(i\omega)| = 0.195$, which means that the stability requirement is $|\delta| < 5.1$. The analysis clearly indicates that the system has good robustness and that the high-frequency properties of the transmission system are not important for the design of the cruise controller.

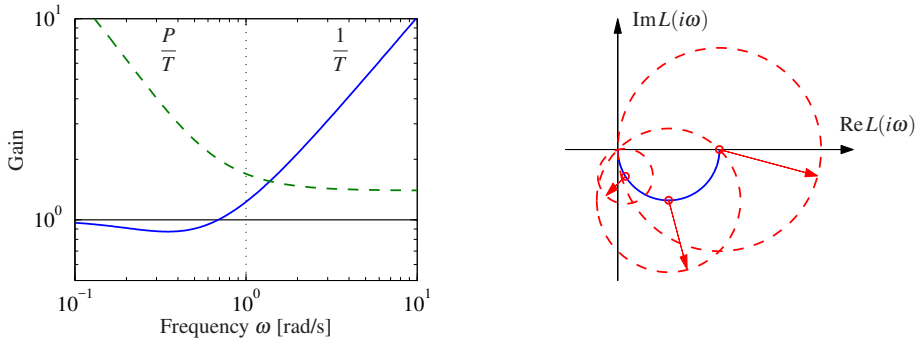


Figure 13.6: Robustness for a cruise controller. On the left the maximum relative error $1/|T|$ (solid) and the absolute error $|P|/|T|$ (dashed) for the process uncertainty Δ. The Nyquist curve is shown on the right as a solid line. The dashed circles show permissible perturbations in the process dynamics, $|\Delta| = |P|/|T|$, at the frequencies $\omega = 0$, 0.0142 and 0.05.

Another illustration of the robustness of the system is given in the right diagram in Figure 13.6, which shows the Nyquist curve of the transfer function of the process and the uncertainty bounds $\Delta = |P|/|T|$ for a few frequencies. Note that the controller can tolerate large amounts of uncertainty and still maintain stability of the closed loop. ∇

The situation illustrated in the previous example is typical of many processes: moderately small uncertainties are required only around the gain crossover frequencies, but large uncertainties can be permitted at higher and lower frequencies. A consequence of this is that a simple model that describes the process dynamics well around the crossover frequency is often sufficient for design. Systems with many resonant peaks are an exception to this rule because the process transfer function for such systems may have large gains for higher frequencies also, as shown for instance in Example 10.9.

The robustness condition given by equation (13.6) can be given another interpretation by using the small gain theorem (Theorem 10.4). To apply the theorem we start with block diagrams of a closed loop system with a perturbed process and make a sequence of transformations of the block diagram that isolate the block representing the uncertainty, as shown in Figure 13.7. The result is the two-block interconnection shown in Figure 13.7c, which has the loop transfer function

$$L = \frac{PC}{1+PC}\frac{\Delta}{P} = T\delta.$$

Equation (13.6) implies that the largest loop gain is less than 1 and hence the system is stable via the small gain theorem.

The small gain theorem can be used to check robust stability for uncertainty in a variety of other situations. Table 13.1 summarizes a few of the common cases; the proofs (all via the small gain theorem) are left as exercises.

The following example illustrates that it is possible to design systems that are

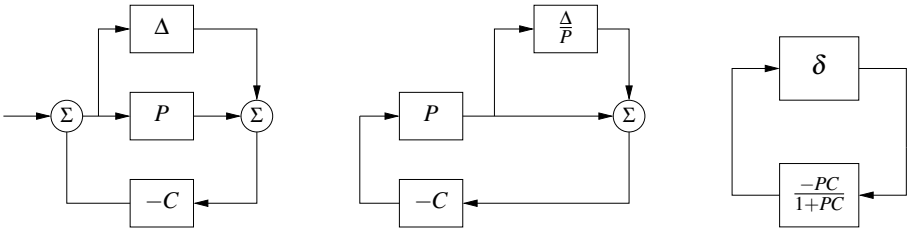


Figure 13.7: Illustration of robustness to process perturbations. A system with additive uncertainty (left) can be manipulated via block diagram algebra to one with multiplicative uncertainty $\delta = \Delta/P$ (center). Additional manipulations isolate the uncertainty in a manner that allows application of the small gain theorem (right)

robust to parameter variations.

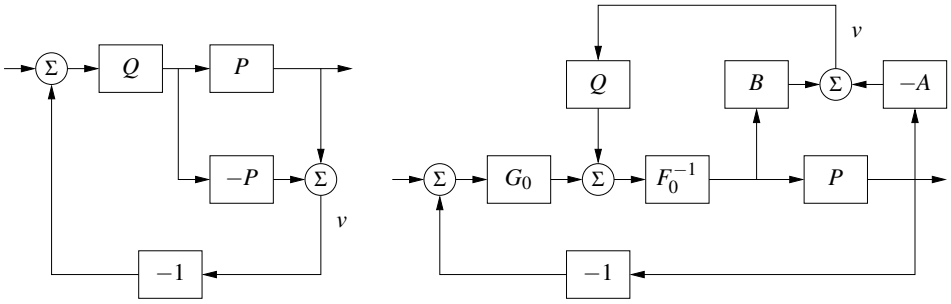
Example 13.6 Bode's ideal loop transfer function
A major problem in the design of electronic amplifiers is to obtain a closed loop system that is insensitive to changes in the gain of the electronic components. Bode found that the loop transfer function $L(s) = ks^{-n}$, with $1 \leq n \leq 5/3$, was an ideal loop transfer function. The gain curve of the Bode plot is a straight line with slope $-n$ and the phase is constant $\arg L(i\omega) = -n\pi/2$. The phase margin is thus $\varphi_m = 90(2-n)^\circ$ for all values of the gain k and the stability margin is $s_m = \sin\pi(1-n/2)$. This transfer function cannot be realized with physical components unless n is an integer, but it can be approximated over a given frequency range with a proper rational function (Exercise 13.7) for any n. An operational amplifier circuit that has the approximate transfer function $G(s) = k/(s+a)$ is a realization of Bode's ideal transfer function with $n = 1$, as described in Example 9.3. Designers of operational amplifiers go to great efforts to make the approximation valid over a wide frequency range. ∇

Youla Parameterization

Since stability is such an essential property, it is useful to characterize all controllers that stabilize a given process. Such a representation, which is called a *Youla parameterization*, is very useful when solving design problems because it makes it possible to search over all stabilizing controllers without the need to test stability explicitly.

Table 13.1: Conditions for robust stability for different types of uncertainty

Process	Uncertainty Type	Robust Stability
$P+\Delta$	Additive	$\|CS\Delta\|_\infty < 1$
$P(1+\delta)$	Multiplicative	$\|T\delta\|_\infty < 1$
$P/(1+\Delta_{\text{fb}}\cdot P)$	Feedback	$\|PS\Delta_{\text{fb}}\|_\infty < 1$



(a) Stable process (b) Unstable process

Figure 13.8: Youla parameterization. Block diagrams of Youla parameterizations for a stable system (a) and an unstable system (b). Notice that the signal v is zero in steady state.

We will first derive Youla's parameterization for a stable process with a rational transfer function P. A system with the complementary sensitivity function T can be obtained by feedforward control with the stable transfer function Q if $T = PQ$. Notice that T must have the same right half-plane zeros as P since Q is stable. Now assume that we want to implement the complementary transfer function T by using unit feedback with the controller C. Since $T = PC/(1+PC) = PQ$, it follows that the controller transfer function is

$$C = \frac{Q}{1-PQ}. \tag{13.8}$$

A straightforward calculation gives

$$S = 1-PQ, \qquad PS = P(1-PQ), \qquad CS = Q, \qquad T = PQ.$$

These transfer functions are all stable if P and Q are stable and the controller given by equation (13.8) is thus stabilizing. Indeed, it can be shown that all stabilizing controllers are in the form given by equation (13.8) for some choice of Q. The parameterization is illustrated by the block diagrams in Figure 13.8a. Notice that the signal v is always zero.

A similar characterization can be obtained for unstable systems. Consider a process with a rational transfer function $P(s) = a(s)/b(s)$, where $a(s)$ and $b(s)$ are polynomials. By introducing a stable polynomial $c(s)$, we can write

$$P(s) = \frac{b(s)}{a(s)} = \frac{B(s)}{A(s)},$$

where $A(s) = a(s)/c(s)$ and $B(s) = b(s)/c(s)$ are stable rational functions. Similarly we introduce the controller $C_0(s) = G_0(s)/F_0(s)$, where $F_0(s)$ and $G_0(s)$ are stable rational functions. We have

$$S_0 = \frac{1}{1+P_0C_0} = \frac{AF_0}{AF_0+BG_0}, \qquad PS_0 = \frac{BF_0}{AF_0+BG_0},$$
$$C_0S_0 = \frac{AG_0}{AF_0+BG_0}, \qquad T_0 = \frac{BG_0}{AF_0+BG_0}.$$

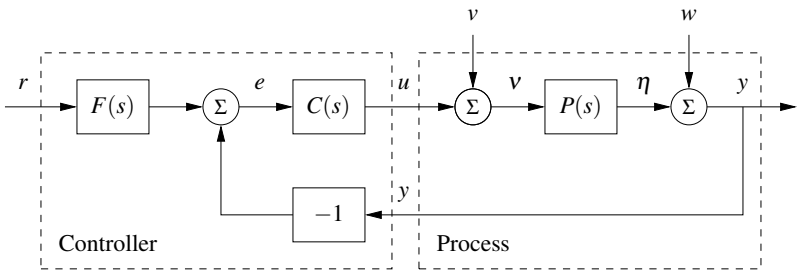


Figure 13.9: Block diagram of a basic feedback loop. The external signals are the reference signal r, the load disturbance v and the measurement noise w. The process output is y, and the control signal is u. The process P may include unmodeled dynamics, such as additive perturbations.

The controller C_0 is stabilizing if and only if the rational function $AF_0 + BG_0$ does not have any zeros in the right half plane. Let Q be a stable rational function and consider the controller

$$C = \frac{G_0 + QA}{F_0 - QB}. \tag{13.9}$$

The Gang of Four for P and C is

$$S = \frac{A(F_0 - QB)}{AF_0 + BG_0}, \qquad PS = \frac{B(F_0 - QB)}{AF_0 + BG_0},$$
$$CS = \frac{A(G_0 + QA)}{AF_0 + BG_0}, \qquad T = \frac{B(G_0 + QA)}{AF_0 + BG_0}.$$

All these transfer functions are stable because A, B, F_0 and G_0 are stable and $AF_0 + BG_0$ does not have any zeros in the right half plane. The controller C given by (13.9) thus stabilizes the closed loop system for any stable Q. A block diagram of the closed loop system with the controller C is shown in Figure 13.8b. Notice that the signal ν is zero.

13.3 Performance in the Presence of Uncertainty

So far we have investigated the risk for instability and robustness to process uncertainty. We will now explore how responses to load disturbances, measurement noise and reference signals are influenced by process variations. To do this we will analyze the system in Figure 13.9, which is identical to the basic feedback loop analyzed in Chapter 12.

Disturbance Attenuation

The sensitivity function S gives a rough characterization of the effect of feedback on disturbances, as was discussed in Section 12.3. A more detailed characterization

is given by the transfer function from load disturbances to process output:

$$G_{yv} = \frac{P}{1+PC} = PS. \tag{13.10}$$

Load disturbances typically have low frequencies, and it is therefore important that the transfer function G_{yv} is small for low frequencies. For processes P with constant low-frequency gain and a controller with integral action it follows from(13.10) that $G_{yv} \approx s/k_i$. The integral gain k_i is thus a simple measure of the attenuation of low frequency load disturbances.

To find out how the transfer function G_{yv} is influenced by small variations in the process transfer function we differentiate equation (13.10) with respect to P yielding

$$\frac{dG_{yv}}{dP} = \frac{1}{(1+PC)^2} = \frac{SP}{P(1+PC)} = S\frac{G_{yv}}{P},$$

and it follows that

$$\frac{dG_{yv}}{G_{yv}} = S\frac{dP}{P}. \tag{13.11}$$

The response to load disturbances is thus insensitive to process variations for frequencies where $|S(i\omega)|$ is small.

A drawback with feedback is that the controller feeds measurement noise into the system. It is thus also important that the control actions generated by measurement noise are not too large. It follows from Figure 13.9 that the transfer function G_{uw} from measurement noise to controller output is given by

$$G_{uw} = -\frac{C}{1+PC} = -\frac{T}{P}. \tag{13.12}$$

Since measurement noise typically has high frequencies, the transfer function G_{uw} should not be too large for high frequencies. The loop transfer function PC is typically small for high frequencies, which implies that $G_{uw} \approx C$ for large s. To avoid injecting too much measurement noise the high frequency gain of the controller transfer function $C(s)$ should thus be small. This property is called *high-frequency roll-off*. In PID control is is common practice to low-pass filter the measured signal; see Section 11.5.

To determine how the transfer function G_{uw} is influenced by small variations in the process transfer, we differentiate equation (13.12):

$$\frac{dG_{uw}}{dP} = \frac{d}{dP}\left(-\frac{C}{1+PC}\right) = \frac{C}{(1+PC)^2}C = -T\frac{G_{uw}}{P}.$$

Rearranging the terms gives

$$\frac{dG_{uw}}{G_{uw}} = -T\frac{dP}{P}. \tag{13.13}$$

Since the complementary sensitivity function is also small for high frequencies, we find that process uncertainty has little influence on the transfer function G_{uw} for frequencies where measurements are important.

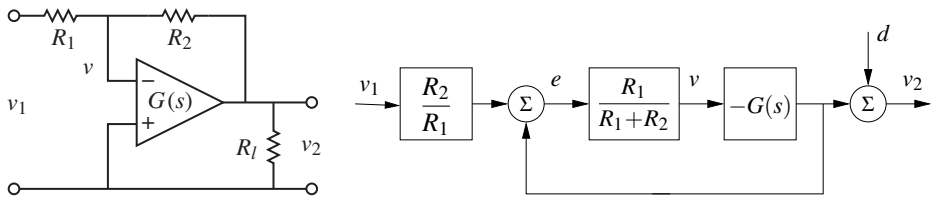


Figure 13.10: Operational amplifier with uncertain dynamics. The circuit on the left is modeled using the transfer function $G(s)$ to capture its dynamic properties and it has a load at the output. The block diagram on the right shows the input/output relationships. The load is represented as a disturbance d applied at the output of $G(s)$.

Reference Signal Tracking

The transfer function from reference to output is given by

$$G_{yr} = \frac{PCF}{1+PC} = TF, \tag{13.14}$$

which contains the complementary sensitivity function. To see how variations in P affect the performance of the system, we differentiate equation (13.14) with respect to the process transfer function:

$$\frac{dG_{yr}}{dP} = \frac{CF}{1+PC} - \frac{PCFC}{(1+PC)^2} = \frac{CF}{(1+PC)^2} = S\frac{G_{yr}}{P},$$

and it follows that

$$\frac{dG_{yr}}{G_{yr}} = S\frac{dP}{P}. \tag{13.15}$$

The relative error in the closed loop transfer function thus equals the product of the sensitivity function and the relative error in the process. In particular, it follows from equation (13.15) that the relative error in the closed loop transfer function is small when the sensitivity is small. This is one of the useful properties of feedback.

As in the previous section, there are some mathematical assumptions that are required for the analysis presented here to hold. As already stated, we require that the perturbations Δ be small (as indicated by writing dP). Second, we require that the perturbations be stable, so that we do not introduce any new right half-plane poles that would require additional encirclements in the Nyquist criterion. Also, as before, this condition is conservative: it allows for any perturbation that satisfies the given bounds, while in practice the perturbations may be more restricted.

Example 13.7 Operational amplifier circuit

To illustrate the use of these tools, consider the performance of an op amp-based amplifier, as shown in Figure 13.10. We wish to analyze the performance of the amplifier in the presence of uncertainty in the dynamic response of the op amp and changes in the loading on the output. We model the system using the block diagram in Figure 13.10b, which is based on the derivation in Example 10.1.

Consider first the effect of unknown dynamics for the operational amplifier. Let the dynamics of the op amp be modeled as $v_2 = -G(s)v$, it then follows from the

block diagram in Figure 13.10b that the transfer function for the overall circuit is

$$G_{v_2v_1} = -\frac{R_2}{R_1}\frac{G(s)}{G(s)+R_2/R_1+1}.$$

We see that if $G(s)$ is large over the desired frequency range, then the closed loop system is very close to the ideal response $\alpha = R_2/R_1$. Assuming $G(s) = b/(s+a)$, where b is the gain-bandwidth product of the amplifier, as discussed in Example 9.3, the sensitivity function and the complementary sensitivity function become

$$S = \frac{s+a}{s+a+\alpha b}, \qquad T = \frac{\alpha b}{s+a+\alpha b}.$$

The sensitivity function around the nominal values tells us how the tracking response response varies as a function of process perturbations:

$$\frac{dG_{yr}}{G_{yr}} = S\frac{dP}{P}.$$

We see that for low frequencies, where S is small, variations in the bandwidth a or the gain-bandwidth product b will have relatively little effect on the performance of the amplifier (under the assumption that b is sufficiently large).

To model the effects of an unknown load, we consider the addition of a disturbance at the output of the system, as shown in Figure 13.10b. This disturbance represents changes in the output voltage due to loading effects. The transfer function $G_{yd} = S$ gives the response of the output to the load disturbance, and we see that if S is small, then we are able to reject such disturbances. The sensitivity of G_{yd} to perturbations in the process dynamics can be computed by taking the derivative of G_{yd} with respect to P:

$$\frac{dG_{yd}}{dP} = \frac{-C}{(1+PC)^2} = -\frac{T}{P}G_{yd} \qquad \Longrightarrow \qquad \frac{dG_{yd}}{G_{yd}} = -T\frac{dP}{P}.$$

Thus we see that the relative changes in the disturbance rejection are roughly the same as the process perturbations at low frequency (when T is approximately 1) and drop off at higher frequencies. However, it is important to remember that G_{yd} itself is small at low frequency, and so these variations in relative performance may not be an issue in many applications. ∇

13.4 Robust Pole Placement

In Chapters 7 and 8 we saw how to design controllers by setting the locations of the eigenvalues of the closed loop system. If we analyze the resulting system in the frequency domain, the closed loop eigenvalues correspond to the poles of the closed loop transfer function and hence these methods are often referred to as design by *pole placement*.

State space design methods, like many methods developed for control system design, do not explicitly take robustness into account. In such cases it is essen-

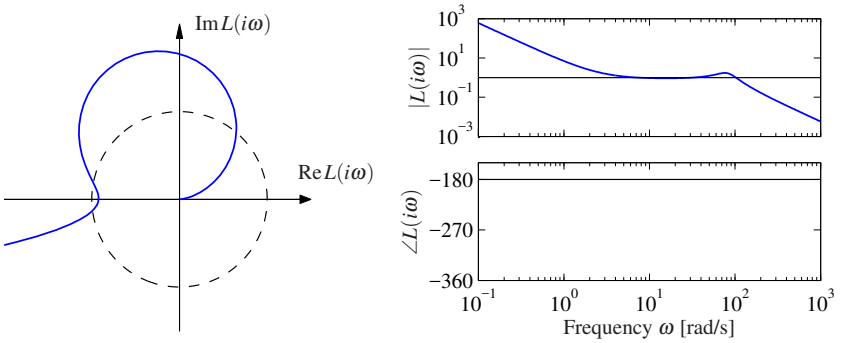


Figure 13.11: Observer-based control of steering. The Nyquist plot (left) and Bode plot (right) of the loop transfer function for vehicle steering with a controller based on state feedback and an observer. The controller provides stable operation, but with very low gain and phase margin.

tial to always investigate the robustness because there are seemingly reasonable designs that give controllers with poor robustness. We illustrate this by analyzing controllers designed by state feedback and observers. The closed loop poles can be assigned to arbitrary locations if the system is observable and reachable. However, if we want to have a robust closed loop system, the poles and zeros of the process impose severe restrictions on the location of the closed loop poles. Some examples are first given; based on the analysis of these examples we then present design rules for robust pole (eigenvalue) placement.

Slow Stable Process Zeros

We will first explore the effects of slow stable zeros, and we begin with a simple example.

Example 13.8 Vehicle steering
Consider the linearized model for vehicle steering in Example 9.10, which has the transfer function

$$P(s) = \frac{0.5s+1}{s^2}.$$

A controller based on state feedback was designed in Example 7.4, and state feedback was combined with an observer in Example 8.4. The system simulated in Figure 8.8 has closed loop poles specified by $\omega_c = 0.3$, $\zeta_c = 0.707$, $\omega_o = 7$ and $\zeta_o = 9$. Assume that we want a faster closed loop system and choose $\omega_c = 10$, $\zeta_c = 0.707$, $\omega_o = 20$ and $\zeta_o = 0.707$. Using the state representation in Example 8.3, a pole placement design gives state feedback gains $k_1 = 100$ and $k_2 = -35.86$ and observer gains $l_1 = 28.28$ and $l_2 = 400$. The controller transfer function is

$$C(s) = \frac{-11516s+40000}{s^2+42.4s+6657.9}.$$

Figure 13.11 shows Nyquist and Bode plots of the loop transfer function. The

Nyquist plot indicates that the robustness is poor since the loop transfer function is very close to the critical point -1. The phase margin is 7° and the stability margin is $s_m = 0.077$. The poor robustness shows up in the Bode plot, where the gain curve hovers around the value 1 and the phase curve is close to -180° for a wide frequency range. More insight is obtained by analyzing the sensitivity functions, shown by solid lines in Figure 13.12. The maximum sensitivities are $M_s = 13$ and $M_t = 12$, indicating that the system has poor robustness.

At first sight it may be surprising that a controller where the nominal closed system has well damped poles and zeros is so sensitive to process variations. We have an indication that something is unusual because the controller has a slow zero at $s = 3.5$, recall that the observer and controller poles have $\omega_c = 10$ and $\omega_o = 20$. To understand what happens, we will investigate the reason for the peaks of the sensitivity functions.

Let the transfer functions of the process and the controller be

$$P(s) = \frac{n_p(s)}{d_p(s)}, \qquad C(s) = \frac{n_c(s)}{d_c(s)},$$

where $n_p(s)$, $n_c(s)$, $d_p(s)$ and $d_c(s)$ are the numerator and denominator polynomials. The complementary sensitivity function is

$$T(s) = \frac{PC}{1+PC} = \frac{n_p(s)n_c(s)}{d_p(s)d_c(s)+n_p(s)n_c(s)}.$$

The poles of $T(s)$ are the poles of the closed loop system and the zeros are the zeros of the process and of the controller. A Bode plot of the gain curve of T is shown in Figure 13.12a. We have $T(0) = 1$, because $L(0) = P(0)C(0) = \infty$, due to the double integrator of P. The gain $|T(i\omega)|$ increases for increasing ω due to the process zero at $\omega = 2$. It increases further at the controller zero at $\omega = 3.5$, and it does not start to decrease until the closed loop poles appear at $\omega = 10$ and $\omega = 20$. The gain of the complementary sensitivity function has a peak indicating poor sensitivity of the closed loop system.

The peak of the complementary sensitivity function can be avoided by assigning a closed loop pole at the slow process zero or close to it. We can achieve this by choosing $\omega_c = 10$ and $\zeta_c = 2.6$, which gives closed loop poles at $s = -2$ and $s = -50$. The controller transfer function then becomes

$$C(s) = \frac{3628s+40000}{s^2+80.28s+156.56} = 3628\,\frac{s+11.02}{(s+2)(s+78.28)}.$$

The sensitivity functions are shown by dashed lines in Figure 13.12b. The closed loop system has the maximum sensitivities $M_s = 1.34$ and $M_t = 1.41$, which indicate good robustness. Notice that the controller has a pole at $s = -2$ that cancels the slow process zero. The design can be done simply by first canceling the slow stable process zero and then designing the controller. ∇

One lesson from the example is that it is necessary to choose closed loop poles that are equal to or close to slow stable process zeros. Another lesson is that slow

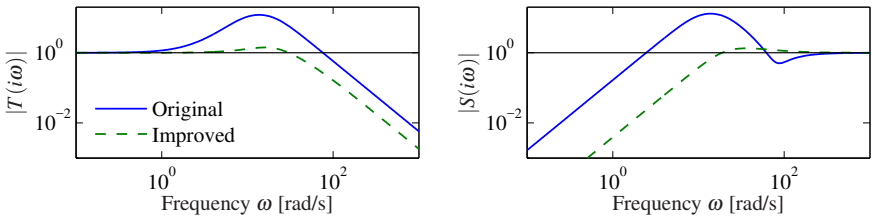


Figure 13.12: Gain curves of Bode plots for the sensitivity functions of observer-based control of vehicle steering. The complementary sensitivity function is shown in (a) and the sensitivity function in (b). The plots for the original controller with $\omega_c = 10$, $\zeta_c = 0.707$, $\omega_o = 20$, $\zeta_o = 0.707$ is shown in solid lines and the improved controller with $\omega_c = 10$, $\zeta_c = 2.6$ is shown in dashed lines.

unstable process zeros impose limitations on the achievable bandwidth, as already noted in Section 12.6.

Fast Stable Process Poles

The next example shows the effect of fast stable poles.

Example 13.9 Fast system poles
Consider a PI controller for a first-order system, where the process and the controller have the transfer functions $P(s) = b/(s+a)$, with a ¿ 0, and $C(s) = k_p + k_i/s$. The loop transfer function is

$$L(s) = \frac{b(k_p s + k_i)}{s(s+a)},$$

and the closed loop characteristic polynomial is.

$$s(s+a) + b(k_p s + k_i) = s^2 + (a + bk_p)s + k_i b$$

If we specify the desired closed loop poles should be $-p_1$ and $-p_2$, we find that the controller parameters are given by

$$k_p = \frac{p_1 + p_2 - a}{b}, \qquad k_i = \frac{p_1 p_2}{b}.$$

The sensitivity functions are then

$$S(s) = \frac{s(s+a)}{(s+p_1)(s+p_2)}, \qquad T(s) = \frac{(p_1 + p_2 - a)s + p_1 p_2}{(s+p_1)(s+p_2)}.$$

Assume that the process pole $-a$ is much more negative than the closed loop poles $-p_1$ and $-p_2$, say, $p_1 < p_2 \ll a$. Notice that the proportional gain k_p is negative and that the controller has a zero in the right half-plane if $a > p_1 + p_2$, an indication that the system has bad properties.

Next consider the sensitivity function, which is 1 for high frequencies. Figure 13.13a shows that the increases for $\omega = a$ corresponding to the process pole. The sensitivity does not decrease until the breakpoints of closed loop poles are

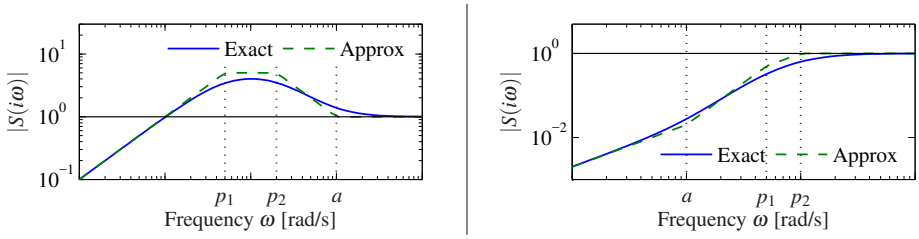


Figure 13.13: Gain curves of Bode plots of the sensitivity function S for designs with $p_1 < p_2 < a$ (left) and $a < p_1 < p_2$ (right). The solid lines are the true sensitivities, and the dashed lines are the asymptotes.

reached, resulting in a large sensitivity peak that is approximately a/p_2. The gain of the sensitivity function is shown in Figure 13.13 for $a = b = 1$, $p_1 = 0.05$ and $p_2 = 0.2$. Notice the high-sensitivity peak. For comparison we also show the gain curve for the case when the closed loop poles ($p_1 = 5$, $p_2 = 20$) are faster than the process pole ($a = 1$).

The problem with poor robustness can be avoided by choosing one closed loop pole equal to the process pole, i.e., $p_2 = a$. The controller gains then become

$$k_p = \frac{p_1}{b}, \qquad k_i = \frac{ap_1}{b},$$

which means that the fast process pole is canceled by a controller zero at $s = -a$. The loop transfer function and the sensitivity functions are

$$L(s) = \frac{bk_p}{s}, \qquad S(s) = \frac{s}{s+bk_p}, \qquad T(s) = \frac{bk_p}{s+bk_p}.$$

The maximum sensitivities are now less than 1 for all frequencies. Notice that this is possible because the process transfer function goes to zero as s^{-1}. ∇

Design Rules for Pole Placement

Based on the insight gained from the examples, it is now possible to obtain design rules that give designs with good robustness. Consider the expression (13.7) for maximum complementary sensitivity, repeated here:

$$M_t = \sup_\omega |T(i\omega)| = \left\| \frac{PC}{1+PC} \right\|_\infty.$$

Let ω_{gc} be the desired gain crossover frequency. Assume that the process has zeros that are slower than ω_{gc}. The complementary sensitivity function is 1 for low frequencies, and it increases for frequencies close to the process zeros unless there is a closed loop pole in the neighborhood. To avoid large values of the complementary sensitivity function we find that the closed loop system should therefore have poles close to or equal to the slow stable zeros. This means that slow stable zeros should be canceled by controller poles. Since unstable zeros cannot be canceled, the presence of slow unstable zeros means that achievable gain crossover

frequency must be smaller than the slowest unstable process zero.

Now consider process poles that are faster than the desired gain crossover frequency. Consider the expression for the maximum of the sensitivity function:

$$M_s = \sup_{\omega} |S(i\omega)| = \left\| \frac{1}{1+PC} \right\|_{\infty}.$$

The sensitivity function is 1 for high frequencies. Moving from high to low frequencies, the sensitivity function increases at the fast process poles. Large peaks can result unless there are closed loop poles close to the fast process poles. To avoid large peaks in the sensitivity the closed loop system should therefore have poles that match the fast process poles. This means that the controller should cancel the fast process poles by controller zeros. Since unstable modes cannot be canceled, the presence of a fast unstable pole implies that the gain crossover frequency must be sufficiently large.

To summarize, we obtain the following simple rule for choosing closed loop poles: slow stable process zeros should be matched by slow closed loop poles, and fast stable process poles should be matched by fast closed loop poles. Slow unstable process zeros and fast unstable process poles impose severe limitations.

Example 13.10 Nanopositioning system for an atomic force microscope
A simple nanopositioner with the process transfer function

$$P(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

was explored in Example 10.9. It was shown that the system could be controlled using an integral controller. The closed-loop performance was poor because the gain crossover frequency was limited to $\omega_{gc} < 2\zeta\omega_0(1 - s_m)$ to have good robustness whith the integrating controller. It can be shown that little improvement is obtained by using a PI controller. We will explore if better performance can be obtained with PID control. For a modest performance increase, we will use the design rule derived in Example 13.9 that fast stable process poles should be canceled by controller zeros. The controller transfer function should thus be chosen as

$$C(s) = \frac{k_d s^2 + k_p s + k_i}{s} = \frac{k_i}{s} \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{\omega_0^2} \tag{13.16}$$

which gives $k_p = 2\zeta k_i/\omega_0$ and $k_d = k_i/\omega_0^2$. The loop transfer function becomes $L(s) = k_i/s$.

Figure 13.14 shows, in dashed lines, the gain curves for the Gang of Four for a system designed with $k_i = 0.5$. A comparison with Figure 10.12 shows that the bandwidth is increased significantly from $\omega_{gc} = 0.01$ to $\omega_{gc} = k_i = 0.5$. Since the process pole is canceled, the system will, however, still be very sensitive to load disturbances with frequencies close to the resonant frequency. The gain curve of CS has a dip or a notch at the resonant frequency ω_0, which implies that the controller gain is very low for frequencies around the resonance. The gain curve

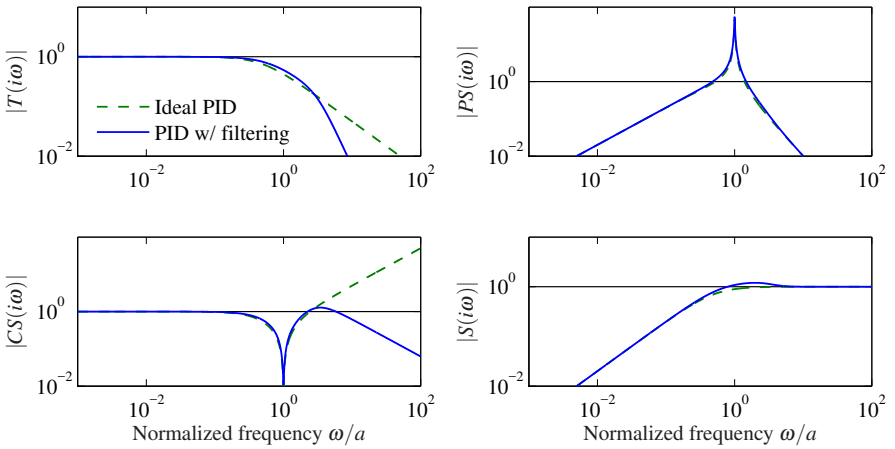


Figure 13.14: Nanopositioning system control via cancellation of the fast process pole. Gain plots for the Gang of Four for PID control with second-order filtering (13.17) are shown by solid lines, and the dashed lines show results for an ideal PID controller without filtering (13.16).

also shows that the system is very sensitive to high-frequency noise. The system will likely be unusable because the gain goes to infinity for high frequencies.

The sensitivity to high frequency noise can be remedied by modifying the controller to be

$$C(s) = \frac{k_i}{s} \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{a^2(1 + sT_f + (sT_f)^2/2)}, \tag{13.17}$$

which has high-frequency roll-off. Selection of the constant T_f for the filter is a compromise between attenuation of high-frequency measurement noise and robustness. A large value of T_f reduces the effects of sensor noise significantly, but it also reduces the stability margin. Since the gain crossover frequency without filtering is k_i, a reasonable choice is $T_f = 0.2/k_i$, as shown by the solid curves in Figure 13.14. The plots of $|CS(i\omega)|$ and $|S(i\omega)|$ show that the sensitivity to high-frequency measurement noise is reduced dramatically at the cost of a marginal increase of sensitivity. Notice that the poor attenuation of disturbances with frequencies close to the resonance is not visible in the sensitivity function because of the cancellation of the resonant poles.

The designs thus far have the drawback that load disturbances with frequencies close to the resonance are not attenuated, since $|S(i\omega_0)|$ is close to one. We will now consider a design that actively attenuates the poorly damped modes. We start with an ideal PID controller where the design can be done analytically, and we add high-frequency roll-off. The loop transfer function obtained with this controller is

$$L(s) = \frac{a^2(k_d s^2 + k_p s + k_i)}{s(s^2 + 2\zeta as + a^2)}. \tag{13.18}$$

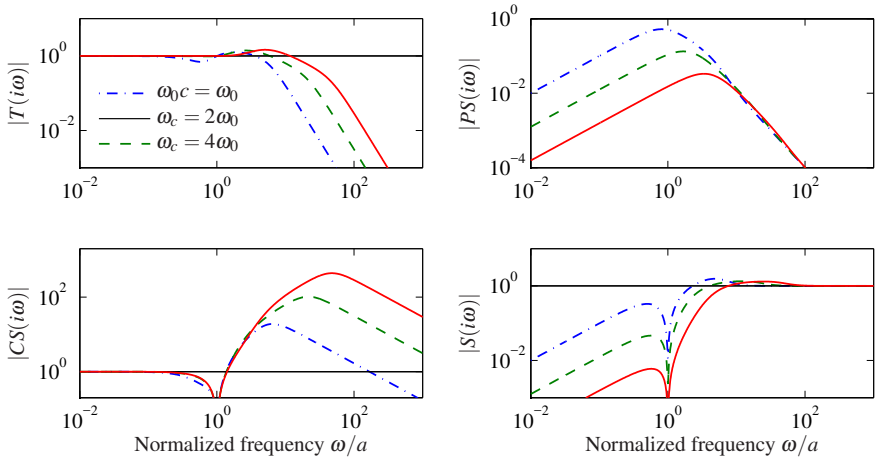


Figure 13.15: Nanopositioner control using active damping. Gain curves for the Gang of Four for PID control of the nanopositioner designed for $\omega_c = \omega_0$ (dash-dotted), $2\omega_0$ (dashed), and $4\omega_0$ (solid). The controller has high-frequency roll-off and has been designed to give active damping of the oscillatory mode. The different curves correspond to different choices of magnitudes of the poles, parameterized by ω_c in equation (13.19).

The closed loop system is of third order, and its characteristic polynomial is

$$s^3 + (k_d a^2 + 2\zeta a)s^2 + (k_p + 1)a^2 s + k_i a^2. \tag{13.19}$$

A general third-order polynomial can be parameterized as

$$s^3 + (\alpha_c + 2\zeta_c)\omega_c s^2 + (1 + 2\alpha_c\zeta_c)\omega_c^2 s + \alpha_c\omega_c^3. \tag{13.20}$$

The parameters α_c and ζ_c give the relative configuration of the poles, and the parameter ω_c gives their magnitudes, and therefore also the bandwidth of the system.

The identification of coefficients of equal powers of s with equation (13.19) gives a linear equation for the controller parameters, which has the solution

$$k_p = \frac{(1 + 2\alpha_c\zeta_c)\omega_c^2}{a^2} - 1, \qquad k_i = \frac{\alpha_c\omega_c^3}{\omega_0^2}, \qquad k_d = \frac{(\alpha_c + 2\zeta_c)\omega_c}{a^2} - \frac{2\zeta_c}{\omega_0}. \tag{13.21}$$

Adding high-frequency roll-off, the controller becomes

$$C(s) = \frac{k_d s^2 + k_p s + k}{s(1 + sT_f + (sT_f)^2/2)}. \tag{13.22}$$

If the PID controller is designed without the filter the filter timeconstant must be significantly smaller than T_d to avoid introducing extra phase lag, a reasonable value is $T_f = T_d/10 = 0.1\,k_d/k$. If more filtering is desired it is necessary to account for the filter dynamics in the design.

Figure 13.15 shows the gain curves of the Gang of Four for designs with $\zeta_c = 0.707$, $\alpha_c = 1$ and $\omega_c = \omega_0$, $2\omega_0$ and $4\omega_0$. The figure shows that the largest values of the sensitivity function and the complementary sensitivity function are

small. The gain curve for *PS* shows that the load disturbances are now well attenuated over the whole frequency range, and attenuation increases with increasing ω_0. The gain curve for *CS* shows that large control signals are required to provide active damping. The high gain of *CS* for high frequencies also shows that low-noise sensors and actuators with a wide range are required. The largest gains for *CS* are 19, 103 and 434 for $\omega_0 = a$, $2a$ and $4a$, respectively. There is clearly a trade-off between disturbance attenuation and controller gain. A comparison of Figures 13.14 and 13.15 illustrates the trade-offs between control action and disturbance attenuation for the designs with cancellation of the fast process pole and active damping. ∇

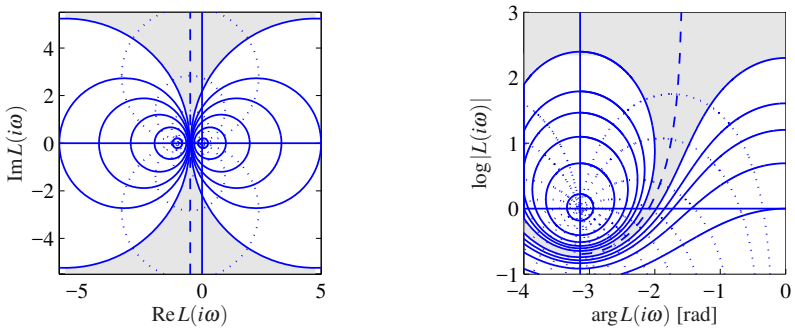
13.5 Design for Robust Performance

Control design is a rich problem where many factors have to be taken into account. Typical requirements are that load disturbances should be attenuated, the controller should inject only a moderate amount of measurement noise, the output should follow variations in the command signal well and the closed loop system should be insensitive to process variations. For the system in Figure 13.9 these requirements can be captured by specifications on the sensitivity functions S and T and the transfer functions G_{yv}, G_{uw}, G_{yr} and G_{ur}. Notice that it is necessary to consider at least seven transfer functions, as discussed Section 12.1. The requirements are mutually conflicting, and may be necessary to make trade-offs. The attenuation of load disturbances will be improved if the bandwidth is increased, but so will the noise injection.

It is highly desirable to have design methods that can guarantee robust performance. Such design methods did not appear until the late 1980s. Many of these design methods result in controllers having the same structure as the controller based on state feedback and an observer. In this section we provide a brief review of some of the techniques as a preview for those interested in more specialized study.

Quantitative Feedback Theory

Quantitative feedback theory (QFT) is a graphical design method for robust loop shaping that was developed by I. M. Horowitz [Hor91]. The idea is to first determine a controller that gives a complementary sensitivity that is robust to process variations and then to shape the response to reference signals by feedforward. The idea is illustrated in Figure 13.16a, which shows the level curves of the gain $|T(i\omega)|$ of thecomplementary sensitivity function on a Nyquist plot. The complementary sensitivity function has unit gain on the line $\mathrm{Re}\, L(i\omega) = -0.5$. In the neighborhood of this line, significant variations in process dynamics only give moderate changes in the complementary transfer function. The shaded part of the figure corresponds to the region $0.9 < |T(i\omega)| < 1.1$. To use the design method, we represent the uncertainty for each frequency by a region and attempt to shape



(a) Hall chart (b) Nichols chart

Figure 13.16: Hall and Nichols charts. The Hall chart is a Nyquist plot with curves for constant gain and phase of the complementary sensitivity function T. The Nichols chart is the conformal map of the Hall chart under the transformation $N = \log L$ (with the scale flipped). The dashed curve is the line where $|T(i\omega)| = 1$, and the shaded region corresponding to loop transfer functions whose complementary sensitivity changes by no more than $\pm 10\%$ is shaded.

the loop transfer function so that the variation in T is as small as possible. The design is often performed using the Nichols chart shown in Figure 13.16b.

Linear Quadratic Control

One way to make the trade-off between the attenuation of load disturbances and the injection of measurement noise is to design a controller that minimizes the loss function

$$J = \int_0^\infty \left(y^2(t) + \rho u^2(t)\right) dt,$$

where ρ is a weighting parameter as discussed in Section 7.3. This loss function gives a compromise between load disturbance attenuation and disturbance injection because it balances control actions against deviations in the output. If all state variables are measured, the controller is a state feedback $u = -Kx$ and it has the same form as the controller obtained by eigenvalue assignment (pole placement) in Section 7.2. However, the controller gain is obtained by solving an optimization problem. It has been shown that this controller is very robust. It has a phase margin of at least 60° and an infinite gain margin. The controller is called a *linear quadratic control* or *LQ control* because the process model is linear and the criterion is quadratic.

When all state variables are not measured, the state can be reconstructed using an observer, as discussed in Section 8.3. It is also possible to introduce process disturbances and measurement noise explicitly in the model and to reconstruct the states using a Kalman filter, as discussed briefly in Section 8.4. The Kalman filter has the same structure as the observer designed by eigenvalue assignment in Section 8.3, but the observer gains L are now obtained by solving an optimization problem. The control law obtained by combining linear quadratic control with a

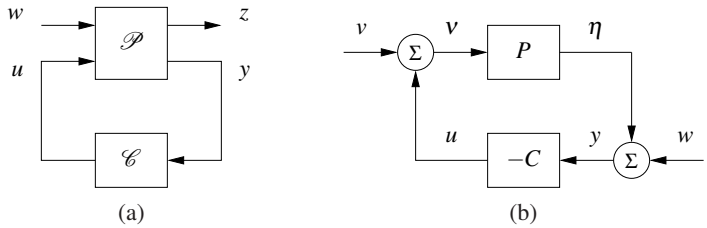


Figure 13.17: H_∞ robust control formulation. (a) General representation of a control problem used in robust control. The input u represents the control signal, the input w represents the external influences on the system, the output z is the generalized error and the output y is the measured signal. (b) Special case of the basic feedback loop in Figure 13.9 where the reference signal is zero. In this case we have $w = (n,d)$ and $z = (y,-u)$.

Kalman filter is called *linear quadratic Gaussian control* or *LQG control*. The Kalman filter is optimal when the models for load disturbances and measurement noise are Gaussian.

It is interesting that the solution to the optimization problem leads to a controller having the structure of a state feedback and an observer. The state feedback gains depend on the parameter ρ, and the filter gains depend on the parameters in the model that characterize process noise and measurement noise (see Section 8.4). There are efficient programs to compute these feedback and observer gains.

The nice robustness properties of state feedback are unfortunately lost when the observer is added. There are parameters that give closed loop systems with poor robustness, similar to Example 13.8. We can thus conclude that there is a fundamental difference between using sensors for all states and reconstructing the states using an observer.

H_∞ Control

Robust control design is often called H_∞ control for reasons that will be explained shortly. The basic ideas are simple, but the details are complicated and we will therefore just give the flavor of the results. A key idea is illustrated in Figure 13.17a, where the closed loop system is represented by two blocks, the process $\mathscr{P}$ and the controller $\mathscr{C}$ as discussed in Section 12.1. The process $\mathscr{P}$ has two inputs, the control signal u, which can be manipulated by the controller, and the generalized disturbance w, which represents all external influences, e.g., command signals and disturbances. The process has two outputs, the generalized error z, which is a vector of error signals representing the deviation of signals from their desired values, and the measured signal y, which can be used by the controller to compute u. For a linear system and a linear controller the closed loop system can be represented by the linear system

$$z = H(P(s),C(s))w, \tag{13.23}$$

which tells how the generalized error z depends on the generalized disturbances w. The control design problem is to find a controller C such that the gain of the transfer function H is small even when the process has uncertainties. There are many

different ways to specify uncertainty and gain, giving rise to different designs. The names H_2 and H_∞ control correspond to the norms $\|H\|_2$ and $\|H\|_\infty$.

To illustrate the ideas we will consider a regulation problem for a system where the reference signal is assumed to be zero and the external signals are the load disturbance v and the measurement noise w, as shown in Figure 13.17b. The generalized input is $w = (-n, d)$. (The negative sign of w is not essential but is chosen to obtain somewhat nicer equations.) The generalized error is chosen as $z = (\eta, \nu)$, where η is the process output and ν is the part of the load disturbance that is not compensated by the controller. The closed loop system is thus modeled by

$$z = \begin{pmatrix} y \\ -u \end{pmatrix} = \begin{pmatrix} \dfrac{1}{1+PC} & \dfrac{P}{1+PC} \\ \dfrac{C}{1+PC} & \dfrac{PC}{1+PC} \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix} = H(P,C) \begin{pmatrix} w \\ v \end{pmatrix}, \tag{13.24}$$

which is the same as equation (13.23). A straightforward calculation shows that

$$\|H(P,C))\|_\infty = \sup_\omega \frac{\sqrt{(1+|P(i\omega)|^2)(1+|C(i\omega)|^2)}}{|1+P(i\omega)C(i\omega)|}. \tag{13.25}$$

There are numerical methods for finding a stabilizing controller such that $\|H(P,C)\|_\infty < \gamma$, if such a controller exists. The best controller can then be found by iterating on γ. The calculations can be made by solving *algebraic Riccati* equations, e.g., by using the command `hinfsyn` in MATLAB. The controller has the same order as the process and the same structure as the controller based on state feedback and an observer; see Figure 8.7 and Theorem 8.3.

Notice that if we minimize $\|H(P,C)\|_\infty$, we make sure that the transfer functions $G_{yv} = P/(1+PC)$, representing the transmission of load disturbances to the output, and $G_{uw} = -C/(1+PC)$, representing how measurement noise is transmitted to the control signal, are small. Since the sensitivity and the complementary sensitivity functions are also elements of $H(P,C)$, we have also guaranteed that the sensitivities are less than γ. The design methods thus balance performance and robustness.

There are strong robustness results associated with the H_∞ controller. It follows from equations (13.4) and (13.25) that

$$\inf_\omega \frac{\|P+1/C\|}{\sqrt{(1+\|P\|^2)(1+\|(1/C)|^2\|}} = \frac{1}{\|H(P,C)\|_\infty}. \tag{13.26}$$

The inverse of $\|H(P,C)\|_\infty$ is thus be interpreted as the shortest distance between P and $-1/C$ and can therefore be interpreted as a *generalized stability margin*. Compare with s_m, which we defined as the shortest distance between the Nyquist curve of the loop transfer function and the critical point -1. It also follows that if we find a controller C with $\|H(P,C)\|_\infty < \gamma$, then this controller will stabilize any process P_* such that $\delta_\nu(P,P_*) < 1/\gamma$.

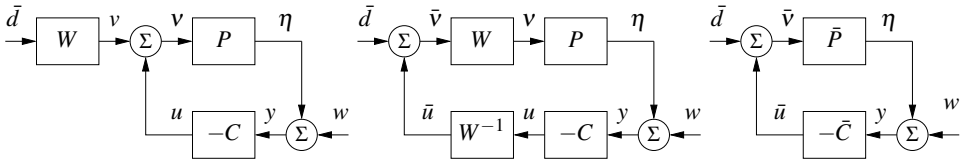


Figure 13.18: Block diagrams of a system with disturbance weighting. The left figure provides a frequency weight on processes disturbances. Through block diagram manipulation, this can be converted to the standard problem on the right.

Disturbance Weighting

Minimizing the gain $\|H(P,C)\|_\infty$ means that the gains of all individual signal transmissions from disturbances to outputs are less than γ for all frequencies of the input signals. The assumption that the disturbances are equally important and that all frequencies are also equally important is not very realistic; recall that load disturbances typically have low frequencies and measurement noise is typically dominated by high frequencies. It is straightforward to modify the problem so that disturbances of different frequencies are given different emphasis, by introducing a weighting filter on the load disturbance as shown in Figure 13.18. For example, low-frequency load disturbances will be enhanced by choosing W as a low-pass filter because the actual load disturbance is $W\bar{d}$.

By using block diagram manipulation as shown in Figure 13.18, we find that the system with frequency weighting is equivalent to the system with no frequency weighting in Figure 13.18 and the signals are related through

$$\bar{z} = \begin{pmatrix} y \\ \bar{u} \end{pmatrix} \begin{pmatrix} \dfrac{1}{1+\bar{P}\bar{C}} & \dfrac{\bar{P}}{1+\bar{P}\bar{C}} \\ \dfrac{\bar{C}}{1+\bar{P}\bar{C}} & \dfrac{\bar{P}\bar{C}}{1+\bar{P}\bar{C}} \end{pmatrix} \begin{pmatrix} w \\ \bar{v} \end{pmatrix} = H(\bar{P},\bar{C})\bar{w}, \tag{13.27}$$

where $\bar{P} = PW$ and $\bar{C} = W^{-1}C$. The problem of finding a controller $\bar{C}$ that minimizes the gain of $H(\bar{P},\bar{C})$ is thus equivalent to the problem without disturbance weighting; having obtained $\bar{C}$, the controller for the original system is then $C = W\bar{C}$. Notice that if we introduce the frequency weighting $W = k/s$, we will automatically get a controller with integral action.

Limits of Robust Design

There is a limit to what can be achieved by robust design. In spite of the nice properties of feedback, there are situations where the process variations are so large that it is not possible to find a linear controller that gives a robust system with good performance. It is then necessary to use other types of controllers. In some cases it is possible to measure a variable that is well correlated with the process variations. Controllers for different parameter values can then be designed and the corresponding controller can be chosen based on the measured signal. This type of

control design is called *gain scheduling*. The cruise controller is a typical example where the measured signal could be gear position and velocity. Gain scheduling is the common solution for high-performance aircraft where scheduling is done based on Mach number and dynamic pressure. When using gain scheduling, it is important to make sure that switches between the controllers do not create undesirable transients (often referred to as *bumpless transfer*).

If it is not possible to measure variables related to the parameters, *automatic tuning* and *adaptive control* can be used. In automatic tuning the process dynamics are measured by perturbing the system, and a controller is then designed automatically. Automatic tuning requires that parameters remain constant, and it has been widely applied for PID control. It is a reasonable guess that in the future many controllers will have features for automatic tuning. If parameters are changing, it is possible to use adaptive methods where process dynamics are measured online.

13.6 Further Reading

The topic of robust control is a large one, with many articles and textbooks devoted to the subject. Robustness was a central issue in classical control as described in Bode's classical book [Bod45]. Robustness was deemphasized in the euphoria of the development of design methods based on optimization. The strong robustness of controllers based on state feedback, shown by Anderson and Moore [AM90], contributed to the optimism. The poor robustness of output feedback was pointed out by Rosenbrock [RM71], Horowitz [Hor75] and Doyle [Doy78] and resulted in a renewed interest in robustness. A major step forward was the development of design methods where robustness was explicitly taken into account, such as the seminal work of Zames [Zam81]. Robust control was originally developed using powerful results from the theory of complex variables, which gave controllers of high order. A major breakthrough was made by Doyle, Glover, Khargonekar and Francis [DGKF89], who showed that the solution to the problem could be obtained using Riccati equations and that a controller of low order could be found. This paper led to an extensive treatment of H_∞ control, including books by Francis [Fra87], McFarlane and Glover [MG90], Doyle, Francis and Tannenbaum [DFT92], Green and Limebeer [GL95], Zhou, Doyle and Glover [ZDG96], Skogestand and Postlethwaite [SP05] and Vinnicombe [Vin01]. A major advantage of the theory is that it combines much of the intuition from servomechanism theory with sound numerical algorithms based on numerical linear algebra and optimization. The results have been extended to nonlinear systems by treating the design problem as a game where the disturbances are generated by an adversary, as described in the book by Basar and Bernhard [BB91]. Gain scheduling and adaptation are discussed in the book by Åström and Wittenmark [ÅW08].

Exercises

13.1 Consider systems with the transfer functions $P_1 = 1/(s+1)$ and $P_2 = 1/(s+a)$. Show that P_1 can be changed continuously to P_2 with bounded additive and multiplicative uncertainty if $a > 0$ but not if $a < 0$. Also show that no restriction on a is required for feedback uncertainty.

13.2 Consider systems with the transfer functions $P_1 = (s+1)/(s+1)^2$ and $P_2 = (s+a)/(s+1)^2$. Show that P_1 can be changed continuously to P_2 with bounded feedback uncertainty if $a > 0$ but not if $a < 0$. Also show that no restriction on a is required for additive and multiplicative uncertainties.

13.3 (Difference in sensitivity functions) Let $T(P,C)$ be the complementary sensitivity function for a system with process P and controller C. Show that

$$T(P_1,C) - T(P_2,C) = \frac{(P_1 - P_2)C}{(1+P_1C)(1+P_2C)},$$

and derive a similar formula for the sensitivity function.

13.4 (The Riemann sphere) Consider systems with the transfer functions $P_1 = k/(s+1)$ and $P_2 = k/(s-1)$. Show that

$$d(P_1,P_2) = \frac{2k}{1+k^2}, \qquad \delta_v(P_1,P_2) = \begin{cases} 1, & \text{if } k < 1 \\ \dfrac{2k}{1+k^2} & \text{otherwise.} \end{cases}$$

Use the Riemann sphere to show geometrically that $\delta_v(P_1,P_2) = 1$ if $k < 1$. (Hint: It is sufficient to evaluate the transfer function for $\omega = 0$.)

13.5 (Stability margins) Consider a feedback loop with a process and a controller having transfer functions P and C. Assume that the maximum sensitivity is $M_s = 2$. Show that the phase margin is at least 30° and that the closed loop system will be stable if the gain is changed by 50%.

13.6 (Bode's ideal loop transfer function) Make Bode and Nyquist plots of Bode's ideal loop transfer function. Show that the phase margin is $\varphi_m = 180° – 90°n$ and that the stability margin is $s_m = \arcsin \pi(1 - n/2)$.

13.7 Consider a process with the transfer function $P(s) = k/(s(s+1))$, where the gain can vary between 0.1 and 10. A controller that is robust to these gain variations can be obtained by finding a controller that gives the loop transfer function $L(s) = 1/(s\sqrt{s})$. Suggest how the transfer function can be implemented by approximating it by a rational function.

13.8 (Smith predictor) The *Smith predictor*, a controller for systems with time delays, is a special version of Figure 13.8a with $P(s) = e^{-s\tau}P_0(s)$ and $C(s) = C_0(s)/(1+C_0(s)P(s))$. The controller $C_0(s)$ is designed to give good performance for the process $P_0(s)$. Show that the sensitivity functions are

$$S(s) = \frac{1 + (1 - e^{-s\tau})P_0(s)C_0(s)}{1 + P_0(s)C_0(s)}, \qquad T(s) = \frac{P_0(s)C_0(s)}{1 + P_0(s)C_0(s)}e^{-s\tau}.$$

13.9 (Ideal delay compensator) Consider a process whose dynamics are a pure time delay with transfer function $P(s) = e^{-s}$. The ideal delay compensator is a controller with the transfer function $C(s) = 1/(1-e^{-s})$. Show that the sensitivity functions are $T(s) = e^{-s}$ and $S(s) = 1-e^{-s}$ and that the closed loop system will be unstable for arbitrarily small changes in the delay.

13.10 (Vehicle steering) Consider the Nyquist curve in Figure 13.11. Explain why part of the curve is approximately a circle. Derive a formula for the center and the radius and compare with the actual Nyquist curve.

13.11 Consider a process with the transfer function

$$P(s) = \frac{(s+3)(s+200)}{(s+1)(s^2+10s+40)(s+40)}.$$

Discuss suitable choices of closed loop poles for a design that gives dominant poles with undamped natural frequency 1 and 10.

13.12 (AFM nanopositioning system) Consider the design in Example 13.10 and explore the effects of changing parameters α_0 and ζ_0.

13.13 (H_∞ control) Consider the matrix $H(P,C)$ in equation (13.24). Show that it has the singular values

$$\sigma_1 = 0, \qquad \sigma_2 = \bar{\sigma} = \sup_\omega \frac{\sqrt{(1+|P(i\omega)|^2)(1+|C(i\omega)|^2)}}{|1+P(i\omega)C(i\omega)|} = \|H(P,C))\|_\infty.$$

Also show that $\bar{\sigma} = 1/d_\nu(P,-1/C)$, which implies that $1/\bar{\sigma}$ is a generalization of the closest distance of the Nyquist plot to the critical point.

13.14 Show that

$$\delta_\nu(P,-1/C) = \inf_\omega \frac{|P(i\omega)+1/C(i\omega)|}{\sqrt{(1+|P(i\omega)|^2)(1+1/|C(i\omega)|^2)}} = \frac{1}{\|H(P,C))\|_\infty}.$$

13.15 Consider the system

$$\frac{dx}{dt} = Ax+Bu = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} x + \begin{pmatrix} a-1 \\ 1 \end{pmatrix} u, \qquad y = Cx = \begin{pmatrix} 0 & 1 \end{pmatrix} y.$$

Design a state feedback that gives $\det(sI-BK) = s^2+2\zeta_c\omega_c s+\omega_c^2$, and an observer with $\det(sI-LC) = s^2+2\zeta_o\omega_o s+\omega_o^2$ and combine them using the separation principle to get an output feedback. Choose the numerical values $a = 1.5$, $\omega_c = 5$, $\zeta_c = 0.6$ and $\omega_o = 10$, $\zeta_o = 0.6$. Compute the eigenvalues of the perturbed system when the process gain is increased by 2%. Also compute the loop transfer function and the sensitivity functions. Is there a way to know beforehand that the system will be highly sensitive?

13.16 (Robustness using the Nyquist criterion) Another view of robust performance can be obtained through appeal to the Nyquist criterion. Let $S_{\max}(i\omega)$ represent a desired upper bound on our sensitivity function. Show that the system provides this level of performance subject to additive uncertainty Δ if the following inequality is satisfied:

$$|1+\tilde{L}| = |1+L+C\Delta| > \frac{1}{|S_{\max}(i\omega)|} \qquad \text{for all } \omega \geq 0. \tag{13.28}$$

Describe how to check this condition using a Nyquist plot.

Chapter Fourteen
Architecture and System Design

An architecture is the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and the principles governing its design. and evolution

ISO/IEC/IEEE 42010:2011

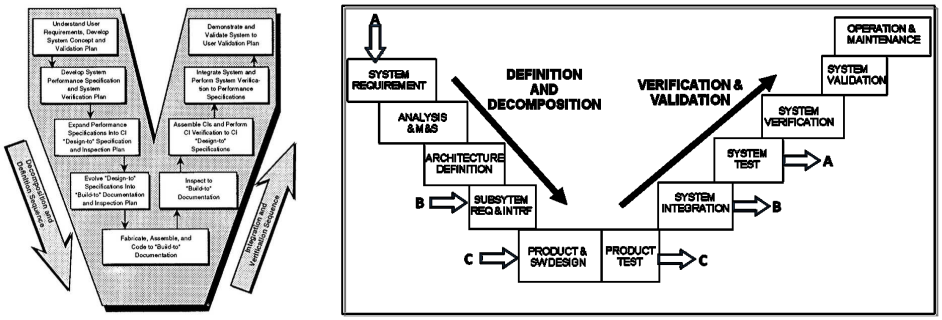
In this chapter we will place the simple feedback loops in the context of system design. First we outline a typical design process. We discuss the role of architecture and and how it can be approached from top-down and bottom-up perspectives. Interaction, adaptation and autonomy are then reviewed and the section ends with a brief overview of control design in some major industrial fields.

14.1 Introduction

So far we have dealt with relatively simple feedback systems. We will now give a glimpse of how they appear as components of real systems and how they are designed. All control systems have sensors, actuators, communication, computers and operator interfaces, but they can have dramatically different sizes and shapes and very different user communities. It is surprising that such a variety of systems can be analyzed and designed using the same framwork.

Many advantage can be obtained by codesign of a process and its control system. Care can be taken to ensure that the system is easy to control, by avoiding non-minimum phase dynamics, use of feedback gives an extra degree of freedom to the designer; a system can be made manoevrable by making it unstable and stabilizing it with a controller. Section 14.2 gives an overview of system design. The system itself and its physical and operational environment are key elemenst together with requirements, analysis and testing.

Architecture from the Greek arkhitekton (from - ”chief” and ”builder, carpenter, mason”) is both the process and the product of planning, designing, and constructing buildings and other physical structures. In the context of control systems, the physical structure consists of the process, sensors, actuators, computers, communication devices, human machine interfaces, algorithms and software. The control system interacts with the operational environment, it observes the process by sensors and it interacts with the process through actuators and with the users through a range of interfaces. Architecture describes how the system components are connected and how they interact. There is a growing awareness that architecture is important in all engineering fields and today we haved software, hardware and systems architects.



(a) 1992 NASA Design V [SC92] (b) Modern Design V [Boa16]

Figure 14.1: Design V Development Process.

A top-down view of architecture is presented in Section 14.3 and the bottom-up view is given in Section 14.4. Section 14.5 treats interaction, adaptation and cognition and autonomy are discussed in Section 14.6 and Section 14.6e. A perspective on control design in aerospace, automotive, process, power and telecommunication industries is given in Section 14.7.

14.2 System and Control Design

Control system design includes many activities starting with requirements and modeling and ending with implementation, testing, commissioning, operation and upgrading. In between are the important steps of modeling, architecture selection, analysis, design and simulation. Control design is often part of design of a system, both follow the sam pattern which can be illustrated by the so-called Design V in Figure 14.1, which dates back to NASA's Apollo program. The left side of the figure shows how design moves from understanding of the process and the requirements to a simulation of the complete system. The right hand side shows how the system is implemented, tested, commisioned, operated and upgraded. Testing is only indicated at one stage of the process but it is good practice to perform it after each step in the design, this is particularly important when the design is modified. On the right leg of the V we have also shown hardware-in-the-loop (HIL) simulation, which is a simulation where some subsystems are actual hardware and other are simulated. Design always starts by developing an understanding of the system and its environment. This includes sensors, actuators, static and dynamic process characteristics, limitations on signals, bounds for safe operation and characterization of disturbances.

Even if we are focusing on control system design it it obvious that there is an interaction between process a and control design. Early analysis can reveal that there are fundamental limitations such as time delays, right half plane poles and zeros or insufficient actuation authority. Poles are inherent to the system and can only be modified by significant process modifications. The zeros can be influenced by moving or adding sensors. A nice example of dealing with insufficient actuator

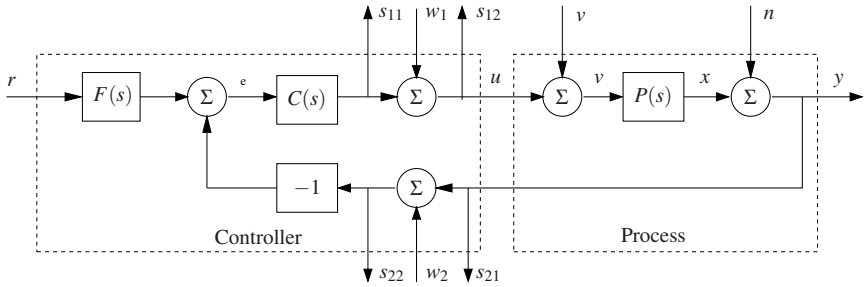


Figure 14.2: Requirements can be tested by injecting signals at test points w_k and measuring responses at s_{ij}. Compare with Figure Figure 12.1

authority is presented in [EHBM08]. It was attempted to reduce risk for rotating stall in a jet engine by feedback but actuators with the required bandwidth were not available. Analysis showed that the problem could instead be alleviated by modifying the process by introducing small asymmetries.

The requirements are given by large and small signal behavior of the closed loop system. Large signal behavior is characterized by limitations in actuation power and its rate, small signal behavior is typically caused by measurement noise, friction, and resolution of AD and DA converters. Requirements for control systems typically include the ability to deal with disturbances, robustness to process variations and uncertatinty and requirements on command signal following. These can all be captured by linear models and they can be expressed in terms of properties of the Gangs of Four and Seven, see equations (12.2) and (12.3)

Refering to the block diagram in Figure 14.2 load disturbance attenuation can be characterized by the transfer function G_{yv} from load disturbance v to process output y. Measurement noise n generates undesired control actions, the effect can be captured by transfer function G_{un} from measurement noise n to control action u. Robustness to parameter variations and process uncertainty can be captured by the sensitivity functions S and T. Command signal response can be shaped independently of response to disturbances and robustness for systems with two degrees of freedom. It is characerized by the transfer functions FT and CSF. For systems with error feedback the response to command signals is characterized by the complementary transfer function T.

For specific systems load disturbances and measurement noise may enter in places that are different from Figure 14.2 and the transfer functions should then be modified accordingly, see Figure 12.2

Since many requirements are expressed in terms of properties of transfer functions of the Gang of Seven, it is important to measure these transfer functions on simulated models and on real equipment. To do this the system must be provided with test points for injecting and measure signals, see Figure 14.2. The transfer function G_{yv}, which characterizes response to load disturbances, can be found by injecting a signal at w_1 and measuring the output s_{21}. Chirp signals are convenient for measuring frequency responses.

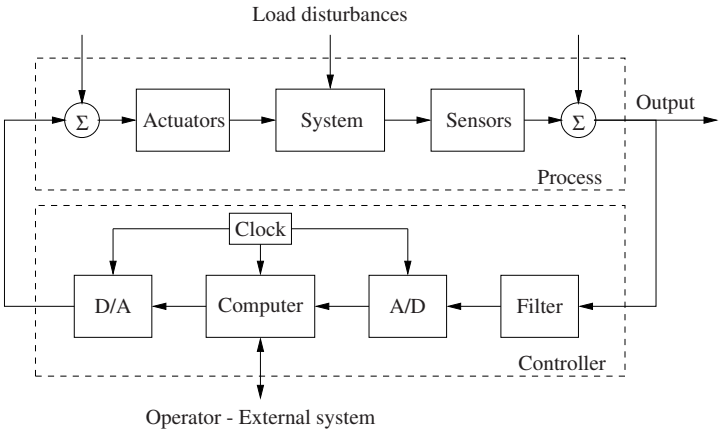


Figure 14.3: Schematic diagram of a control system with sensors, actuators, communications, comuter and interfaces.

Models of the process and its environment can be obtained from physics or from experiments or a combination. Experiments are typically done by changing the control signal and measuring the response. The signals can range from simple step tests to signals that are designed to give optimal information with limited process perturbations. System identification methods and software provide useful tools. The models used typically have different fidelity, cruder in the beginning and more accurate as the design progresses.

A few design methods have been discussed in Chapters 8, 12, and 13, there are many more methods in the literature []. Many design methods are based on linear models, when environmental conditions change significantly it it necessary to use gain scheduling, nonlinear control or adaptation. In many cases there is good software to execute the control design provided that models and criteria are available.

Today most control systems are implemented using computer control. Implementation then involves selection of hardware for signal conversion, communication and computing. A block diagram of a system with computer control is shown in Figure 14.3. The filter before the AD converter is necessary to ensure that high frequency disturbances do not appear as alised low frequency disturbances after sampling. The operations of the system are synchronized by a clock

Real time operating systems that coordinates sensing actuation and computing have to be selected and algorithms that implement the control laws must be generated. The sampling period and the anti-alias filter must be chosen carefully. Since a computer can only do basic arithmetic the control algorithms have to be represented as difference equations. They can be obtained by approximating differential equations as was illustrated in Section 8.5 but there are also design methods that automatically gives controllers in the form of difference equations. Code can be generated automaticallly. It must also be ensured that computational delays and

synchronization of algorithms do not create problems.

When the design is implemented and tested the it must be commissioned, this step may involve adjustment of some parameters, automatic tuning as discussed in Section 11.3 can be very beneficial at this stage. During operation it is important to monitor the behavior of the system to ensure that requirements are still satisfied. It may be necessary to upgrade the system when it has been operating. Requirements may also be modified due to operational experiences.

It is highly desirable to have a suite of test programs that can be used throughout the design and operation stages to ensure that requirements are satisfied.

14.3 Top-Down Architectures

When designed is aproached systematically using the design V as described in Section 14.2 it is natural to use a top-down procedure for control design starting with the requirements and decomposing the overall control design problem into an interlinked set of control problems at different layers of abstraction. At each layer of abstraction we make assumptions about the interactions with the higher and lower layers in order to simplify the control design problem. In this section we give a brief introduction to some of the organizing principles of top-down architectures for control and its connections to some of the techniques described in the text, along with references to the literature for those interested in further details.

There are many other aspects of control architectures that are part control systems design. We mention here a few of these, just to indicate the breadth of applicability of the ideas we have introduced. Among other aspects we can mention: cooperative control [], health monitoring [], fault recovery [] and system reconfiguration [].

14.3a Layered Architectures for Control

For complex control systems, it is often useful to break down the control problem into a hierarchy of control problems, each solved at aquer different layer of abstraction, as illustrated in Figure 14.4. Different types of specifications (shown on the left side of Figure 14.4) are used at each layer to determine the control functionality that will be implemented.

The specific abstractions layers in a control architecture depend on the problem domain. In Figure 14.4, we have used a decomposition that is common in many motion control problems, including robotics and flight control, that be can also appear in application domains ranging from manufacturing to process control to computing systems. At the top layer of abstraction, we care about discrete modes of behavior, which could correspond to different phases of operation (takeoff, cruise, landing) or different environment assumptions (highway driving, city streets, parking lot). The next layer of abstraction reasons about trajectories of the system, often using an optimization-based approach. At this layer, we often take into account the constraints on the system operating state and inputs, as well as

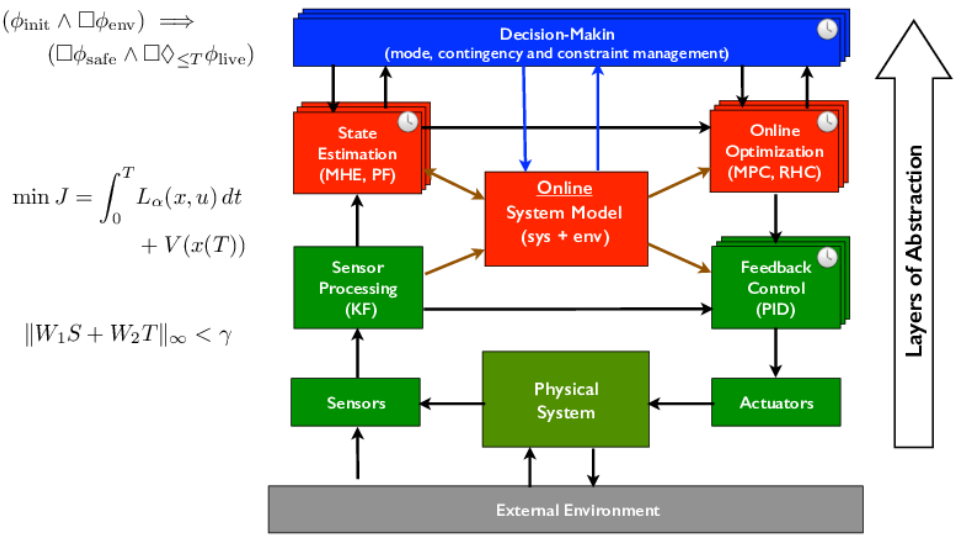


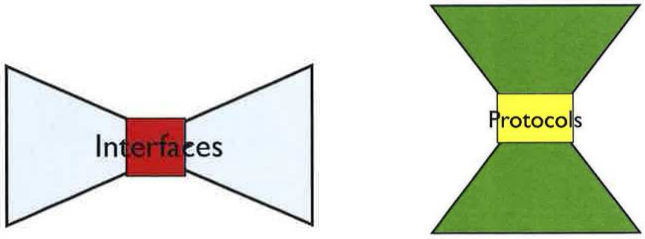
Figure 14.4: Layered decomposition of a control system.

system-level descriptions of performance. Finally, at the lowest layer of the abstration heierarchy we have the feedback control design that has been the main topic in this text thus far, where we may use a linearized model based on the current operating point (along a trajectory).

Note that at each abstraction level we must consider not only the control design but also the way that sensory information is processed and represented. At the lowest levels of abstraction we may use individual sensor signals, perhaps with some filtering. As we move up the abstraction hierarchy, it will be necessary to fuse multiple measurements to obtain a more integrated view of the system state, with the highest level of abstraction often requiring sophisticated methods of reasoning about the state of the environment and the predicted interactions with other entities in that environment.

An important feature of many control systems architectures is the modularity of the control software, enabling parallel development of components and the ability to upgrade components without having to redesign the entire system. Figure 14.5 shows two types of features that are common in architectures: a sensing "bowtie" and a planning "hourglass".

The "bowtie" architectural element refers to the use of a common interface within a layer of abstraction that enables many different subsystems to connect together across the interface. In the context of the sensing system for an autonomous vehicle, a common representation of map data allows many types of sensors to feed information into the map and different layers of controller to extract (fused) information from the map. Through this common interface, new sensors or control functionality can be added on either side of the interface without having to redesign the rest of the system.



(a) Sensing bowtie (b) Planning hourglass

Figure 14.5: Architectural features that support modularity.

The "hourglass" architectural element respresents a hierarchy of control functions that uses a common interface to enable changes above and below that interface to be changed independently of each other. The Open Systems Interconnection model (OSI model) with seven standardized layers, applications at the top and the physical layer at the bottom, has been a key to obtain interoperability in communication systems. For the planning system in an autonomous vehicle, one such hourglass interface is the use of trajectories as a connection between higher and lower levels of the navigation system. Any high-level functions that eventually lead to a trajectory are completely compatible with the lower level controllers that will track that trajectory, allowing the higher levels of decision making to be changed without having to modify the trajctory tracking code. Similarly, the lower level controllers can be changed without having to redesign the high level decision-making, as long as they properly perform the function of tracking a given trajectory.

The bowtie and hourglass patterns shown here can be appear multiple times in a given architecture, so that we obtain appropriate "stacks" of sensing and control functionality.

14.3b Online Optimization

The use of real-time trajectory generation techniques enables a much more sophisticated approach to the design of control systems, especially those in which constraints must be taken into account. The fact that such trajectories can be computed quickly enables us to use a *receding horizon control* technique: a (optimal) feasible trajectory is computed from the current position to the desired position over a finite time T horizon, used for a short period of time $\delta < T$, and then recomputed based on the new system state starting at time $t + \delta$ until time $t + T + \delta$, see Figure 14.6 Development and application of receding horizon control (also called model predictive control, or MPC) originated in process control industries where the processes being controlled are often sufficiently slow to permit its implementation. An overview of the evolution of commercially available MPC technology is given in [QB97] and a survey of the state of stability theory of MPC is given in [MRRS00].

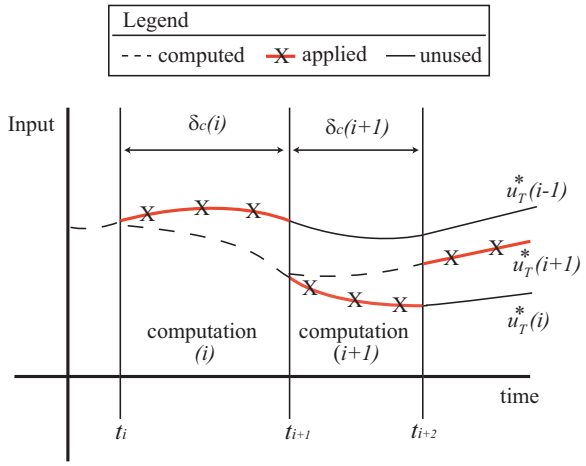


Figure 14.6: The idea of receding horizon control.

Figure 14.7 shows a typical setup for a receding horizon control problem. In this formulation, the trajectory generation block solves the following constrained trajectory generation problem at each time step:

$$\min_{(x,u)} = \int_t^{t+T} L(x,u)\,d\tau + V(x(t+T)) \quad \text{subject to} \quad \begin{cases} x(t) = \text{current state} \\ \dot{x} = f(x) + g(x)u, \\ lb \le c(x,u) \le ub. \end{cases} \tag{14.1}$$

One of the challenges of properly implementing receding horizon control is that instabilities can result if the problem is not specified correctly. In particular, because we optimize the system dynamics over a finite horizon T, it can happen that choosing the optimal short term behavior can lead us *away* from the long term solution (see [**?**] for a more detailed discussion). To address this problem, the terminal cost $V(x(T))$ must have certain properties to insure stability (see [MRRS00] for details).

One of the chief challenges in implementing receding horizon control is the need for fast computation of feasible trajectories. One class of systems for which this is easier are *differentially flat* systems, defined briefly in Section 8.5.

14.3c Discrete-decision making and supervisory control

Design of control systems involves the analysis and synthesis of feedback controllers at multiple levels of abstraction, from fast feedback loops around actuators and subsystems, to higher level decision-making logic in supervisory controllers and autonomous systems. One of the major challenges in design of complex networked control systems—such as those arising in aerospace, computing, robotics, and critical infrastructure—is insuring that the combination of dynamical behavior and logical decision-making satisfies safety and performance specifications. In

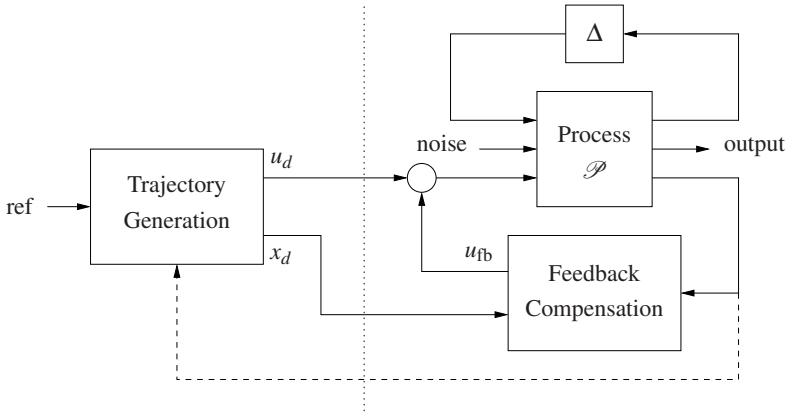


Figure 14.7: Two degree of freedom controller design for a process P with uncertainty Δ. The controller consists of a trajectory generator and feedback controller. The trajectory generation subsystem computes a feedforward command u_d along with the desired state x_d. The state feedback controller uses the measured (or estimated) state and desired state to compute a corrective input u_{fb}. Uncertainty is represented by the block Δ, representing unmodeled dynamics, as well as disturbances and noise.

many of these areas, verification and validation are now dominant drivers of schedule and cost, and the tools available for design of such systems are falling behind the needs of systems and control engineers, particularly in the area of systematic design of the mixed continuous and discrete control laws for networked systems.

We consider systems consisting of subsystems/agents whose dynamics are described by ordinary differential equations of the form

$$\dot{x}^i = f^i(x^i, \alpha^i, u^i), \qquad y^i = h(x^i, \alpha^i),$$

where $x^i \in \mathbb{R}^{n^i}$ is the continuous state of the ith subsystem, $\alpha \in \mathscr{A}$ is the discrete state, $u^i \in \mathbb{R}^{m_i}$ is a control input and $y^i \in \mathbb{R}^{p_i}$ is the measured output of subsystem. The discrete state evolves according to a set of "guarded commands", in which the discrete state α is updated to a new value only if a guard $g^p(x,\alpha)$ is true:

$$g_j^p(x,\alpha) \implies \alpha' = r_j^p(x,\alpha).$$

This specification allows the discrete state to evolve in an asynchronous way (e.g. for modeling failures) or to depend on the system state (e.g. to model nonlinearities or changes in connectivity). A model of this type is called a *discrete transition system*. The overall system, consisting of both continuous dynamics and discrete (state) dynamics, is called a *hybrid system*.

A controller for the system is a combination of a continuous control law and a discrete control *protocol*:

$$u = k(x,\alpha), \qquad g_j^c(x,\alpha) \implies \alpha' = r_j^c(x,\alpha). \tag{14.2}$$

The control protocol is in the form of discrete transition system, which controls some subset of the discrete states. The discrete state is assumed to be updated by a

periodically controlled process that examines the guards and updates appropriate rules. This model for the control allows for the possibility of distributed computation in which different systems (or subsystems) execute on loosely regulated clocks.

The system specification for a hybrid system is often composed of both a continuos performance specification and a discrete performance specification. For the continous portion of the specfication, a typical form is to use a cost function J that is written as a finite horizon cost

$$J = \int_0^T L(x, \alpha, u)dt + V(x(T)). \tag{14.3}$$

This function, which we have already seen in the context of receding horizon control, uses an integral cost over a fixed horizon T along with a terminal cost $V(x(T))$, where V is an appropriate positive function.

For the discrete performance specification, we make use of *temporal logic* formulas. For a discrete transition system, a temporal logic formula describes conditions on the sequence of events. One mathematical language that is widely used is *linear temporal logic* (LTL), which makes use of two useful operators: always ($\Box$) and eventually ($\Diamond$). Given a logical formula $\varphi(\alpha)$ that evaluates to true or false for a given (discrete) state α, we can define a temporal logic formula $\Box\varphi$, which is interpreted as meaning that $\varphi(\alpha)$ should be true at all times in the future. Similarly, the formula $\Diamond\varphi$ represents the temporal logic statement that the logical formula φ is true at some future state. By combining these temporal operators with standard logical operators, we can obtain more complex formulas. For example, the formula

$$\Box(\varphi \implies \Diamond\psi)$$

can be interpreted as saying that at all times, if the formula φ is satisfied (evaluates to true), then eventually (at some point in the future) the formula ψ should be true. This formula is a typical form for specifying that a system should respond to a certain event (captured by φ becoming true) by taking a certain action (captured by ψ becoming true). The always ($\Box$) operator at the outer level of the formula describes the specifiction that this condition should be satisfied at all times, which means that the system should respond over and over again if the event condition occurs repeatedly.

A typical LTL formula for a supervisory control system has the form

$$\varphi_{\text{init}} \wedge \Box\varphi_{\text{env}} \implies \Box\varphi_{\text{safe}} \wedge \Diamond\varphi_{\text{goal}}, \tag{14.4}$$

where φ_{init} represents a proposition describing the initial state of the system, φ_{env} describes the possible actions of the environment, φ_{safe} is a safety requirement and φ_{goal} is a progress requirement. The environmental description φ_{env}, safety requirement φ_{safe} and progress reuqirement φ_{goal} are typically described using LTL or other temporal logics (including computational tree logic, CTL or one of their variants, TCTL, pCTL, etc). These languages allow various specications on sequences of actions, such as requiring that a certain condition hold until another condition is satisfied, or requiring that a certain condition occur on a regular basis.

The control design problem for a supervisory system consists of finding a control law of the form in equation (14.2) that satisfies the system specification (14.4) while minimizing the cost function (14.3). Traditional approaches to this problem involve synthesis of the continuous control law using optimization-based approaches, as described earlier in this section, and manual design of the discrete control protocols. The system is then checked against the specification by running (many) repeated simulations and checking that in each case that the system specification is satisfied.

14.3d Linking Continuous and Discrete Controllers

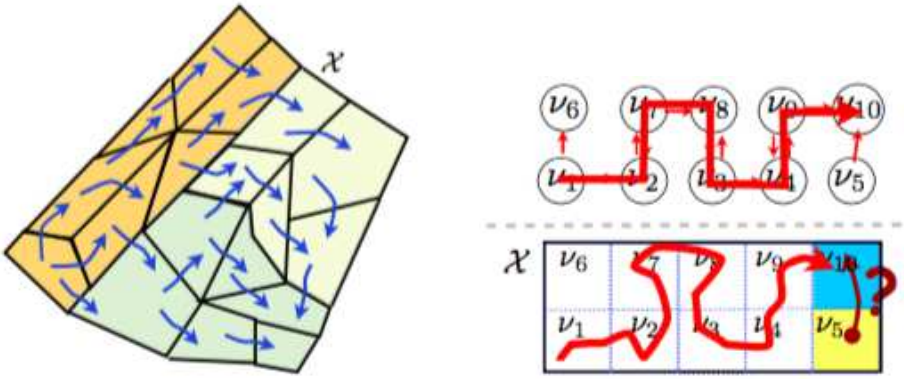
In Section 14.3b we saw techniques for finding (optimal) controllers for continuous systems and in this section we have talked about techniques for synthesis of discrete controllers. Going back to our initial top-down architecture in Figure 14.4, we now consider the problem of how to link these two different layers of the control system.

One approach to linking the two layers is to develop control techniques that can handle both the continuous and discrete dynamics in a common framework. One such approach is to make use of the notion of *mixed logical dynamical* (MLD) systems, introduced by Bemporad and Morari [BM99]. In this formulation, we extend the trajectory generation and receding horizon frameworks to handle discrete variables in the underlying constrained optimization problem. This requires the use of so-scalled *mixed integer solvers*, which allow optimization for problems with both continuous and discrete (integer valued) variables. As computers become increasingly powerful, the size and complexity of problems that we can handle with these tools has increased and these techniques are more and more commonly used.

An alternative, consistent with our philosophy in this section, is to solve the supervisory control problem and the trajectory generation problem separately, with an appropriate simplified representation of the dynamics of the layers above and below the one that for which we are doing the design.

For example, a common approach in doing trajectory generation is to assume that the discrete state (which might represent an operational mode or an environmental context) transitions from one state to another and then remains fixed for a sufficiently long period of time. Under this assumption, we can focus on our attention on the problem of trajectory generation with an initial condition that may represent the state of the system just prior to transition to the new mode and the assume that the mode stays constant for the duration of our planning horizon. We are then required to make sure that our supervisory controller imposes this restriction on the time between mode switches as part of its specification.

This type of linkage between two layers in our abstraction is called a *vertical contract* and can often be written in *assume-guarantee* form. The supervisory layer will assume that the trajectory generation layer maintains the system spefication in a given mode given enough "settling" time, and then guarantees that it does not switch the system mode more quickly than the settling time. Similarly, the trajectory generation will assume that the switching is sufficiently long, but most then



(a) Discretization (b) Simulation relation

Figure 14.8: Representation of continuous dynamics as a discrete transition system

. .

gaurantee that it satisfies the system specification within the prescribed settling time.

Similar to the simplified model of the supervisory controller used at the trajectory generation layer (the mode is essentially constant), the supervisory controller must have an appropriate repesentation of the trajectory generation dynamics. Since the supervisory controller design is done using discrete transition system models, our representation of the dynamics of the lower layers of abstraction must be in terms of a discrete transition system model. One simple approach to such a representation is to break the continuous state space into a collection of discrete regions, such that each continuous state is contained in one discrete region. The dynamics of the trajectory generation layer can then be represented as a set of transitions between adjacent regions in the discretized state space, as illustrated in Figure 14.8a. In setting up the discrete representation, it is important that any trajectory in the discretized state space correspond to a feasible trajectory in the original continuous state space, and *vice versa*. This is required so that when the supervisory controller layer commands the system to move from one region to another then the trajectory generation layer is able to do so, and conversely when the continuous system executes a command and moves from one (continuous) state to another, it corresponds to a valid transition between the regions.

14.3e Model checking and program synthesis

A problem of using simulation to try to check whether the design satisfies the specification is that it can be prohibitively time-consuming to simulate every possible sequence of events. For purely discrete state systems, in which the dynamics are completely specified by a set of discrete variables α, it turns out that there are more efficient techniques for verifying that a system is specified by making use of the structure of the logical formula and the discrete dynamics of the system. These

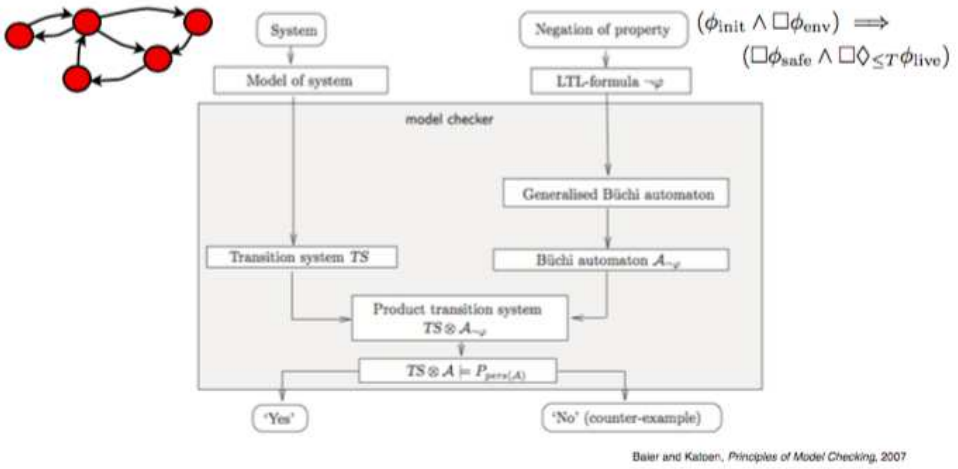


Figure 14.9: Model checking.

verification techniques are referred to as *model checking*.

A block diagram that describes how model checking is performed is shown in Figure 14.9. The main idea of model checking is to make use of a model of the (discrete) system dynamics and the temporal logic specification to create a discrete transition system, known as the product transition system, where finding a path through this transition system represents possible system execution (set of allowable state transitions) that *violates* the system specification. If no such path can be found, then the system specification is satisfied. But if a path is found, it represents a counter-example that can be used to update the controller design.

Model checking tools, such as SPIN [Hol03], nuSMV [CCG+02] and PRISM [KNP11] are capable of handling relatively large discrete state systems with quite general classes of specifications (beyond simple LTL formulas). They are now widely used in industry and are increasingly being applied in mission and safety critical applications, such as planetary exploration and aviation.

Despite their increasing power and applicability, a limitation in the use of model checking is that it does not necessarily provide any insight into how to redesign the system if the specification is not met. Rather, it provides a counterexample indicating what can go wrong, and leaves it to the designer to understand why the controller is not correct and then redesign the system. The updated design can be reverified using the model checker, and this process is iterated until a correct design is obtained.

An alternative to iterative manual design is to make use of results in *program synthesis* or *correct-by-construction* synthesis. The basic idea in synthesis is to create an algorithm that takes a model of the discrete system dynamics along with a temporal logic specification and then *synthesize* a control protocol such that the resulting closed loop system satisfies the specification. This approach is conceptually similar to the LQR design technique outlined in Section 7.3: given a system

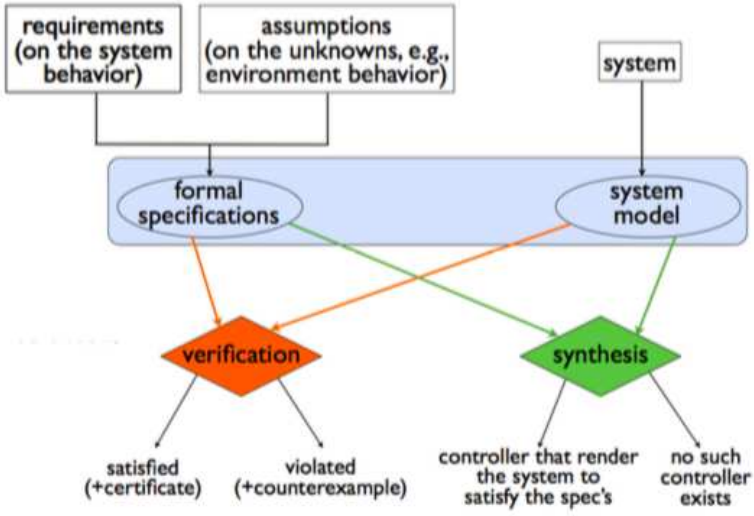


Figure 14.10: Verification versus synthesis.

specification (cost function to minimize) and system model (A, B, C matrices), create a controller ($u = -Kx$) that by construction stabilizes the system and optimizes the performance specification.

Model checking and program synthesis make use of a common set of information (system model and system specification), but solve different problems, as illustrated in Figure 14.10. The advantage of using synthesis is that it provides a control protocol that is guaranteed to satisfy the specification. That is, no matter what sequence of environmental events occur, the system will always respond in a manner that satisfies the specification.

Of course, correct-by-construction synthesis is not a panacea. It can only be used for certain types of specifications and it too can be overwhelmed by complexity. But it is an increasingly important tool in the systematic synthesis of control protocols, especially safety-critical systems. A practical way to obtain a safe system is to design the system in layers with algorithms and guards. The algorithms perform their ordinary tasks and the guards determine if the system is functioning properly. The innermost layer, which for flight control, is called *flying home mode* consists of a simple robust controller that provides the basic properties. If the loop is simple enough it can be designed to be correct-by-construction. The algorithms at the higher layers deliver better performance, and they have guards that move the system to a lower, safer layer, if the system does not perform properly. The safety of the guards must of course also be guaranteed. Systems of this type have been designed by Sha, see [Sha01], examples of guards for adaptive control are found in [THÅ00].

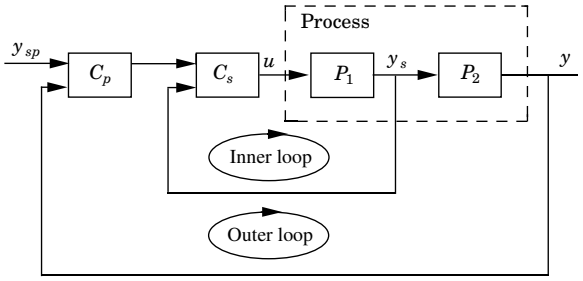


Figure 14.11: Block diagram of a system with cascade control. The system has one control variable u and two measured signals: the primary output y and the secondary or auxiliary output y_s.

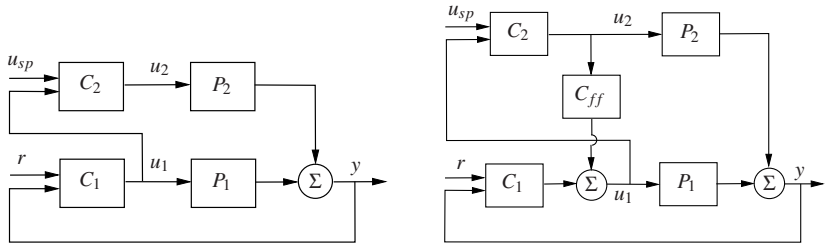
14.4 Bottom-Up Architectures

Building complex systems from standard parts have developed in many branches of engineering. In design of mechanical devices it was found very efficient to standardize nuts an bolts. Design of electronics was similarly done by standardized transistors, circuits, boards and connectors.

In the context of control system design bottom-up desing means that the control system is built loop-by-loop. The elements for blocks are controllers (often PID), filters, nonlinear functions, filters and finite state machines. They can either be separate pieces of hardware or function blocks implemented in software, that can be combined graphically in distributed control systems using cut and paste. The system is built loop by loop by combining control principles such as feedback and feedforward, which have been discussed extensively in Sections 2.6 and 8.5. There are many architectural structures (control principles) that emerged in an ad hoc fashion in many application areas for example *cascade control*, *mid-range control*, *selector control*and *repetitive control*, *model following*, *gain scheduling*, *adaptation* and *extremal control* which will be discussed in this section. An advantage with the bottom-up approach is that the system can be commissioned and tuned loop by loop. There may, however, be difficulties when the loops are interacting. The disadvantage is that it is not easy to judge if additional loops will bring benefits. The system can also be unwieldy when loops are added.

14.4a Cascade Control - Several Sensors

Cascade control is a scheme for using two or more signals and one actuator. The block diagram is shown in Figure 14.11. The system in the figure has two loops. The inner loop is called *the secondary loop* and the outer loop is called *the primary loop*. The reason for this terminology is that the outer loop deals with the primary measured signal. It is also possible to have a cascade control with more nested loops. The ultimate case is state feedback when all states are measured. An example of cascade control is shown in Figure 1.12. Cascade control was also used in Example 12.14 where it was called inner-outer loop design.



(a) Midranging (b) Midranging with feedforward

Figure 14.12: Block diagrams of a two architectures with mid-range control. The system in (a) is the basic architecture and (b) is a refined architecture that also uses feedforward.

Cascade control is useful in when there is significant time delay or dynamics between the input u and the primary output y, but significantly less dynamics between u and the secondary output y_s, and when the major disturbances enter in the block P_1. Cascade control then admits tight feedback in the inner loop, which reduces the effect of disturbances acting on P_1. If integral action is used in both the secondary and primary control loops, it is necessary to have a scheme to avoid integral windup. Anti-windup for the secondary controller can be done in the conventional way since the controller drives the actuator directly. To provide anti-windup for the primary controller it must be told when the secondary controller goes into anti-windup mode. Cascade control is a convenient way to use extra measurements to improve control performance.

14.4b Mid-Range Control - Many Actuators

Midranging is a control architecture when many control signals are used to control one measured output. An example is shown block diagram in Figure 14.12a, where a single variable y is controlled using two subsystes with individual control signals u_1 and u_2 acting through dynamics described by the transfer functions P_1 and P_2. A typical case is that P_1 is fast and accurate but with limited actuation range (low control authority) and that P_2 has slow dynamics but wide actuation range (high control authority). The controller C_1 which drives P_1 is the primary controller that controls the output y to its desired reference r_1. The second controller C_2 drives the subsystem P_2 which has large range. The measured signal to C_2 is the input u_1 to the subsystem P_2, and the controller C_2 attempts to keep the variable u_1 in its mid range. If the signal u_1 is in the middle of its operating range and if only small disturbances are acting on the system, then the controller C_1 manipulates u_1 to reduce the disturbance. For large disturbances, the signal u_1 may reach its actuation limit and the controller C_2 then acts to bring u_1 in range.

The block diagram in Figure 14.12b is an improvement of the basic mid-range control architecture. The output of the controller C_2 is used as a feedforward signal to the controller C_1. The feedforward transfer function is $G_{ff}(s) = -P_2(s)/P_1(s)$. Windup protection must be handled carefully for systems with midranging.

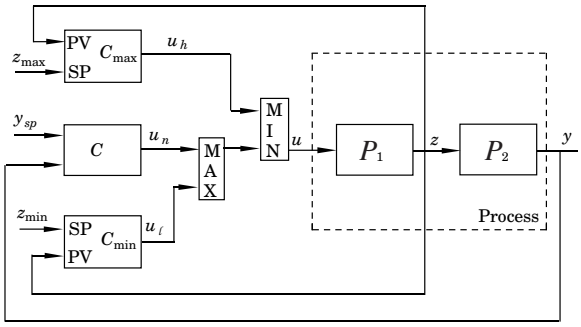


Figure 14.13: Block diagram of a system with selector control. The primary controller is C attempts to keep y close to its reference value. It is typically a PID controller The controllers C_{max} and C_{min} are proportional controller that ensures that the intermediary variable z is in its pemissible range $z_{min} < z < z_{max}$.

14.4c Selector Control - Equipment Protection

Selector control used to control a primary variable while keeping auxiliary variables within given constraints for safety or for equipment protection. A selector is a function with many inputs and one output. The output of a *maximum* selector is the largest of the input signals and the output of a *minimum* is the smallest of the input signals. Other selectors are *median selectors* and *two-out-of-three selectors* that are used for high integrity systems with multiple sensors.

Selector control is commonly used in the power industry to provide safety for example to maintain temperature while ensuring that pressure does not exceed certain limits or to avoid compressor stall.

Selector control is illustrated in Figure 14.13. The primary controlled variable is the process output y. The primary controller C keeps y close to its set-point. To guarantee safe operation the auxiliary variable z that should be kept within the limits $z_{\min}$ and $z_{\max}$. This is accomplished by the scondary controllers C_{max} and C_{min}, which have set-points z_{max} and z_{min}. The secondary controllers connect to the control signal via maximum and minimum selectors.

Under normal conditions the auxiliary variable z is larger than $z_{\min}$ and smaller than $z_{\max}$. The system acts as if the maximum and minimum controller were not present. If the variable z reaches its upper limit, the variable u_h becomes small and is selected by the minimum selector. The control system then attempts to control the variable z and drive it towards z_{max}. A similar situation occurs if the variable z becomes smaller than $z_{\min}$.

A system with selectors has several controllers in parallel, but at each time only one controller is active. The controllers can be tuned in the same way as single-loop controllers. With controllers having integral action, it is necessary to track the integral states of those controllers that are not in operation. Windup protections therefor requires care.

14.4d Complementary Filtering - Sensor Fusion

Complemenatry filtering is a technique that can be used to fuse the information from different sensors, typically one sensor that is slow but accurate and another that is fast but may drift. Consider the situation when we want to give a good estimate of the variable x and that two sensors which gives the signals y_1 and y_2 are available

$$y_1 = x + n_1, \qquad y_2 = x + n_2.$$

The disturbance n_1 has zero mean but the disturbance n_2 may drift indicating that l y_1 is slow but accurate and y_2 fast but drifting. The complementarly filter for recovering the signal x is given by

$$X_f(s) = \frac{1}{s+1} Y_1(s) + \frac{s}{s+1} Y_2(s), \tag{14.5}$$

where $X_r(s)$, $Y_1(s)$ and $Y_2(s)$ are the Laplace transforms of the corresponding signals. Notice that the transfer functions for y_1 and y_2 sums to 1, which explains the name complementary filtering.

Both complementary filtering and Kalman filtering can provide improved estimates by fusing information from several sensors, they can also be obtimized if information about the noise is available. Complementary filtering requires only a model of the sensor system but the Kalman filter requires a model of the complete system dynamics. The Kalman filter can, however, also exploit the control actions. Both methods are widely used for sensor fusion both in simple and advanced systems.

14.4e Extremum Seeking

Among other useful control structures we can mention *extremum seeking* of *self-optimization*. Instead of keeping the process output close to a specified reference value these controller these attempt to change the reference so that an objective function is minimized. To accomplish this it is necessary to change the reference value of the controller and observe the effect of the output. A simple scheme idea is illustrated in Figure 14.14. The reference value of a controller is changed and the behavior of the performance criterion is observed. The performance changes very little close to the optimum. The changes are of performance are in phase with the changes of the reference if the reference is too large and out of phase if the reference is too low. The reference is changes to move towards the smallest value of the objective function. Correlation methods can be used to filter out noise, since the argument is based on a steady–state reasoning the frequency of the perturbation signal must be chosen so low that process dynamics can be neglected. There are many other more sophisticated schemes based on optimization and estimation. There are many other schems for finding optimal operating conditions. They are all characterized by the steps of probing, analysis and action. Many efficient optimization methods can be used if computation power is available.

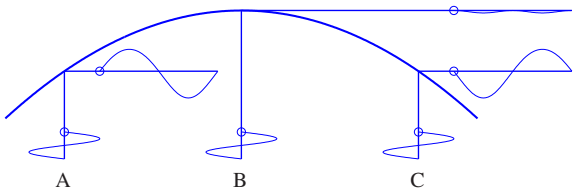


Figure 14.14: Principle for self-optimizing control or extremal seeking. The figure shows the steady state response of the the performance variable v as a funtion of the reference y_r and the effect of sinusoidal variations of the reference.

14.4f Internal Model Control - Disturbance Observer

Figure 14.15 shows a controller architecture that was proposed by Coleman Brosilov. The architecture in is called *internal model control*, *IMC* or *Inferential control*, because the undelayed output was infered from a model. The name reflects that the controller contains a model of the process internally. This model is connected in parallel with the process. The process P is provided with a parallell model $\hat{P}$. The controller is a combination of a filter G_f and an inverse process model $\hat{P}^{-1}$. If $\hat{P}(s) = P(s)$ the signal signal u will not influence the error e which becomes the effective disturbance $e = n + Pv$ at the process output. The signal e is therefore called a *disturbance observer*. Moreover if $G_f = 1$ the control signal u will cancel the disturbance perfectly. Under ideal conditions is thus an ideal controllers.

If $\hat{P} = P$ it follows from Figure 14.15 that the closed loop response to set point changes is given by $G_{yr} = G_f$. The block diagram in Figure **??** can be represented as an standard feedback loop with a process P and a controller C where

$$C = \frac{G_{yr}\hat{P}^{-1}}{1 - G_{yr}\hat{P}^{-1}\hat{P}}. \tag{14.6}$$

It follows from this expression that the controller C cancels all process poles and zeros, which implies that it cannot be used for processes with unstable poles. The same observation can be made simply from the fact that the system has a parallel loop with two identical systems, a prototype for lacking observability and reacha-

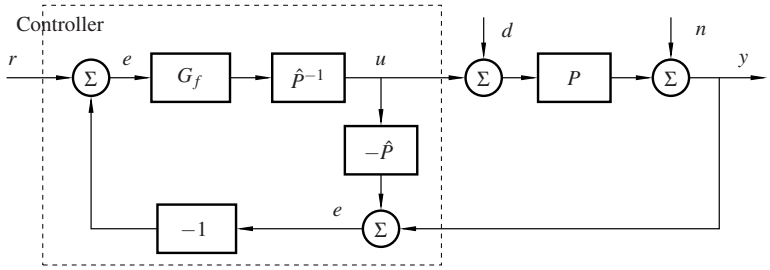
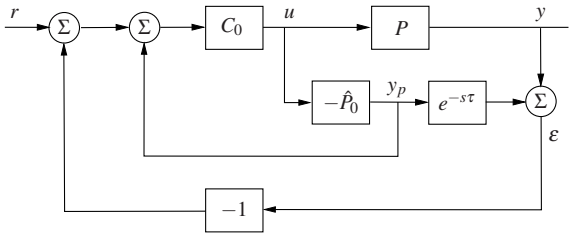
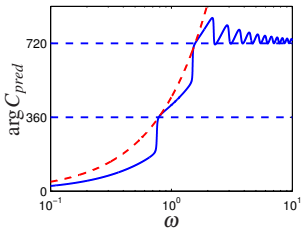


Figure 14.15: Block diagram of a closed-loop system with a controller based on internal model control.



(a) Block diagram for Smith predictor



(b) Bode phase plot of C_{pred}

Figure 14.16: Block diagram of a closed loop system with a Smith predictor (a) and phase plot of predictor transfer function $C_{pred}(s)$ (blue solid) and the ideal predictor e^{sL} (red dashed lines) for $P_0C_0 = \frac{1}{s+1}\left(1+\frac{1}{0.45s}\right)$ (blue full lines) and $L = 8$.

bility.

14.4g The Smith Predictor - Phase Advance

A special controller architecture for systems with time delays have been proposed by Otto Smith. A block diagram of the controller is shown in Figure 14.16a. Let the process have the transfer function $P = P_0 e^{-s\tau}$, where τ is the time delay. The system is provided with a model in parallel with the process, similar to the internal model controller. The parallel model provides the signal y_p which is a proxy for the process output y without the the time delay and the controller C_0 is designed for the delay-free dynamics P_0. Assume that $\hat{P}_0 = P_0$ then the signal ε is zero for all u. Applying block diagram algebra then gives the following transfer function for the closed loop system

$$G_{yr}(s) = \frac{P_0(s)C_0(s)}{1+P_0(s)C_0(s)}e^{-s\tau}. \tag{14.7}$$

To obtain a desired set point response we can just design a controller C_0 for the process P_0 which does not have a time delay, the Smith-predictor architecture thus simplifies the design significantly. The architecture with parallel paths is a prototype or a system that lack reachability and observability. It is particularly serious if the process has unstable modes. The Smith predictor thus cannot be used as is for processes with integrators [ÅHL94].

To get additional insight into the properties of the Smith predictor we observe that if $P = P_0e^{-s\tau}$ and $\hat{P}_0 = P_0$ the block digram Figure 14.16 can be redrawn as a conventional feedback loop with the controller

$$C = \frac{C_0}{1+C_0P_0(1-e^{-s\tau})} = C_0C_{pred}, \qquad C_{pred} = \frac{1}{1+C_0P_0(1-e^{-s\tau})}. \tag{14.8}$$

The controller can thus be viewed as a cascade connection of the conventional controller C_0 with the predictor C_{pred}. Notice that near the gain crossover frequency for P_0C_0 we have $P_0C_0 \approx -1$ and $C_{pred} \approx e^{sL}$ indicating that the transfer function

$C_{pred}(s)$ has a significant phase advance. The Bode phase plot of C_{pred} in Figure 14.16b shows that this is indeed the case. Notice that the phase is very close to the phase of the ideal predictor e^{sl} for the frequencies $\omega = 0.8$, and 1.6.

The Smith predictor gives closed loop systems with good set point responses but load disturbance responses are not much better than with PI control. The predictor C_{pred} is however a useful transfer function to provide phase advance.

14.5 Interaction

A drawback with the bottom up approach when the system is built loop by loop is that there may be unintended interactions. It is therfore important to investigate when interactions arise and what can be done to reduce unintended effects. We will start by introducing *the relative gain array (RGA)* which is a measure of interaction. Next we will discuss *decoupling* which is one way to design a controller that eliminates the effects of the interaction. Parallel systems is a special form of interaction that occurs when sevral subsystems produce the same effect, a typical example is an electric car with motors on each wheel or a power system with may generators.

14.5a The Relative Gain array

The relative gain array is a simple way to determine the degree of interaction in a linear system. To describe the idea we start by investigating a system with two inputs and two outputs, let the transfer function matrix and its inverse be

$$P(s) = \begin{pmatrix} p_{11}(s) & p_{12}(s) \\ p_{21}(s) & p_{22}(s) \end{pmatrix}, \quad P^{-1}(s) = \frac{1}{\det P(s)} \begin{pmatrix} p_{11}(s) & -p_{21}(s) \\ -p_{12}(s) & p_{22}(s) \end{pmatrix}, \quad (14.9)$$

where $\det P(s) = p_{11}(s)p_{22}(s) - p_{12}(s)p_{22}(s)$. Assume that we want to control the system by two single loop controllers. The first problem is to decide if y_1 should be controlled by u_1 or u_2, this is called the *pairing problem* and the second problem is to investigate if there will be interactions between the loops. Sometimes the solution to the problem is clear from the physics of the process. The problem can resolved by trial and error: design controller for the different alternatives and explore their properties.

A clever idea that gives a lot of insight with modest calculations was proposed by Bristol. Assume for simplicity that we are exploring the possibility of controlling y_1 by u_1 and y_2 by u_2. Controllers can be designed based on the transfer functions $p_{11}(s)$ and $p_{22}(s)$. The controller for the first loop will then work well if the second loop is open but the question is how the second loop will influence the second loop. Bristol proposed to look at the ratio

$$\lambda(s) = \frac{p_{11}(s)}{\bar{p}_{11}(s)}, \quad (14.10)$$

where $\bar{p}_{11}(s)$ is the tranfer function from u_1 to y_1 when the *prefectly controlled* in the sense that $y_2 = 0$. Assuming all signals are exponential functions we have for

perfect control of the second loop

$$y_1 = p_{11}u_1 + p_{12}u_2, \qquad 0 = p_{21}u_1 + p_{22}u_2, \qquad y_1, u_1, u_2 \in \mathscr{E}.$$

Eliminating u_2 in these equations gives

$$y_1 = \frac{p_{11}p_{22} - p_{12}p_{21}}{p_{22}} u_1,$$

and we find

$$\bar{p}_{11} = \frac{p_{11}p_{22} - p_{12}p_{21}}{p_{22}}, \quad \text{and} \quad \lambda = \frac{p_{11}p_{22}}{\det P}. \tag{14.11}$$

If $|\lambda(i\omega)| > 1$ is larger than one, the interaction increases the gain and it decreases the gain if $|\lambda(i\omega)| > 1$. There is no interaction if $\lambda(s) = 0$, neither if $\lambda(s) = 1$ but the loops should then be reversed so that y_1 is controlled by u_2. There are rules of thumb for interpreting the relative gain array, interactions can typically be neglected for $3/4 < |\lambda(i\omega)| < 3/2$, which means the controller must then be designed to cope with an additive process uncertainty such that $|\Delta/P| < 1/3$, see Figure 13.2. Decoupling or multivariable control should be considered outside this range.

It follows from equation (14.9) that $\lambda(s)$ is product of the $(1,1)$ elements of the matrices P and P^{-1}. This observation can be generatlized and the interaction for systems with n inputs and n outputs is characterized by the matrix

$$\Lambda(s) = P(s) \circ P^{-T}(s) = P(s).*P^{-T}(s). \tag{14.12}$$

where $P^{-T}(s)$ denotes the transpose of $P^{-1}(s)$ and $\circ$ denotes element by element multiplication of matrices (Hadamard product). The matrix $\Lambda(s)$, which was originally derived for the steady state case ($s = 0$) is called *the relative gain array (RGA)* or *Bristol's RGA*, and it was later extended to dynamics.

The relative gain array has the nice property that it is dimension free and that it gives insight into interactions and pairing of variables, by analysing the gain $|\Lambda(i\omega)|$ we also get insight into interactions at different frequencies. The RGA also gives information about the variables that should be grouped for multivariable control, see [McA83].

14.5b Parallel Systems

There are situations when several subsystems are used to control the same variable. Typical examples include: temperature control using several cooling or heating devices and control of an electric car with one motor on each wheel. An extreme example is control of a power grid, which may have hundreds of energy sources, that all contribute to maintaining frequency and voltage of the net. Care must be execised when making loop-by-loop design of such systems. We illustrate by an a simple example.

Example 14.1 Cruise Control for Electric Car
Consider speed control of an electrical car with motors on each wheel, for simplicity we will consider linear motion with only two motors, and we will use the

simple model (4.1) in Section 4.1. Neglecting all disturbance forces F_d except the force due to gravity; F_r the model (4.3) becomes

$$m\frac{dv}{dt} = F_1 + F_2 - mg\theta, \tag{14.13}$$

where v is the speed of the car, θ the slope of the road, F_1 and F_2 the forces generated by the drive motor.

We will first consider the case when both motors have proportional controllers. Let v_r be the desired (reference) speed, the controllers are then

$$F_1 = k_{p1}(v_r - v), \qquad F_2 = k_{p1}(v_r - v,). \tag{14.14}$$

Combining equations (14.13) and (14.14) gives the following equation for the closed loop system

$$m\frac{dv}{dt} = (k_{p1} + k_{p2})(v_r - v) - mg\theta.$$

If the slope θ is constant there will be a steady state error $e_{ss} = v_r - v_{ss}$ and the steady-state controller outputs are $u_{1ss} = k_{p1}e_o$ and $u_{2ss} = k_{p1}e_o$. The proportional gains k_{p1} and k_{p2} thus determine how the *compensation for the disturbance is distributed among the motors.*

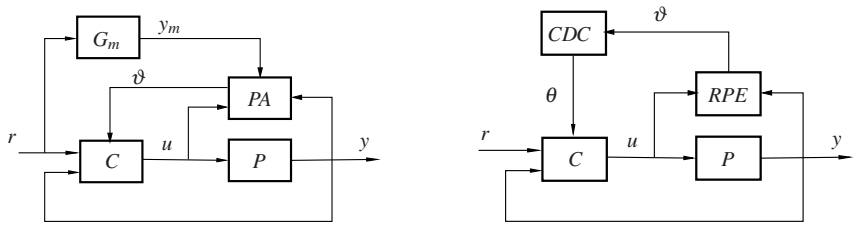
Next we will consider the case when each motor is provided by a PI controller. The closed loop system is then described by the equations

$$\begin{aligned} m\frac{dv}{dt} &= (k_{p1} + k_{p2})(v_r - v) + k_{i1}I_1 + k_{I2}I_2 + mg\theta, \\ \frac{dI_1}{dt} &= v_r - v, \qquad \frac{dI_2}{dt} = v_r - v. \end{aligned} \tag{14.15}$$

This system is not stable, since $\frac{d(I_1 - I_2)}{dt} = 0$ the system has an eigenvalue at the origin. The system (14.15) has the inputs v_r and θ the output y, it is of third order. The system has a the Kalman decomposition Figure 8.9a, with a subsystem $\Sigma_{\bar{r}\bar{o}}$, with integrator dynamics, that is neither reachable nor observable.

If both wheel motors are provided with controllers having controller integral action the integral terms of the controllers will drift in such a way that their sum is constant. As a result one control signal may increase and the other will decrease until saturation occurs. There is a very simple remedy, use one single integrator and distribute the output of that integrator to both motors. ∇

The results of the example can be generalized, if parallel systems are controlled by proportional controllers, then *the controller gains determine how disturbance attenuation is divided among the subsystems*. Moreover, *integral control cannot be used in the individual subsystems*, instead we can select one controller with integral action or we can use *a central integrator* and distribut its output to the controler of the subsystems.



(a) Model reference adaptive control (b) Self-tuning regulator

Figure 14.17: Block diagrams of systems with adaptive control. The left figure shows a model reference adaptive system (MRAC) and the right figure shows a self-tuning regulator (STR). The block P is the process, C is a controller with adjustable parameters θ, ϑ, G_m is a model that gives the ideal response to command signals, *PA* is a parameter adjustment mechanism, *RPE* is a recursive parameter estimator that estimates process parameters recursively and *CDC* is a design calculation that computes the controller parameters ϑ from the process parameters θ.

14.6 Adaptation, Learning and Cognition

In this section we will discuss more sophisticased control laws with abilities to adapt, learn and reason. These functions are key elements for autonomous control. Adapt is to adjust to a specified use or situation, learn is to acquire knowledge or skill by study, instruction or experience, reason is the intellectual process of seeking truth or knowledge by infering from either fact of logic and autonomy is the ability of beeing self-governing. When these words are used in the engineering context it is clear that the abilities are far from what we can be accomplished by humans, but the development of autonomous cars and air vehicles are good indicators of progress.

14.6a Adaptive Control

Adaptive control is a technique that can be used when there are significant variations in the process and its environment and where neither robust control nor gain scheduling is applicable. Model reference control and the self-tuning controller are two common approaches to adaptive control. Their block diagrams are shown in Figure 14.17. Notice that in both cases there are two feedback loops: one conventional feedback loop involving the process P and the controller C and a slower loop to adjust the controller parameters θ.

Model reference adaptive control (MRAC) is primarily used for command signal following. A block diagram of the controller is shown in Figure 14.17a. The controller consists of three blocks G_m, C and *PA*. The desired response to command signals is given by the transfer function G_m. The controller C has adjustable parameters θ. The parameter adjustment mechanism *PA*, receives the process input u and output y and the derired response y_m and it generates the the process parameters. The MIT rule is given by

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}, \tag{14.16}$$

where $e = y_m - y$ and γ is a parameter, is a very simple way to adjust the parameters. There are many other rules, some of them are derived from Lyapunov theory.

The self-tuning regulator (STR) is used both for command signal following and for regulation. The controller is based on the idea of developing a process model automatically and to apply some design method to find a suitable controller. A block diagram of a system is shown in Figure 14.17b. The controller has three blocks, a controller C with adjustable parameters θ, a recursive parameter estimator *RPE* and a controller design calculation *CDC*. The parameter estimator *RPE* estimtes the process parameters ϑ recursively from the process input u and output y. The controller design block *CDC* determines the controller parameters from the process parameters using some design method. In this calculation it is common to treat the estimates as the true parameters, a principle from decision making under uncertainty called the certainty equivalence principle [Sim56]. Uncertainties in the estimates can be taken into account because many estimation schemes provide estimates of parameter uncertainty. The self-tuning regulator is very flexible because many different methods can be used for parameter estimation and control design.

Recursive least squares method for parameter estimation. For the model

$$\begin{aligned} y_{t+1} &= -a_1 y_t - a_2 y_{t-1} + \cdots + b_1 u_t + \cdots + e_{t+1} = \varphi_t \theta^T + e_{t+1}, \\ \varphi_t &= \begin{pmatrix} -y_t & -y_{t-1} & \cdots & u_t & u_{t-1} & \cdots \end{pmatrix}, \\ \theta &= \begin{pmatrix} a_1 & a_2 & \cdots & b_1 & b_2 & \cdots \end{pmatrix}. \end{aligned} \tag{14.17}$$

the parameter estimates are given by

$$\begin{aligned} \hat{\theta}_t &= \hat{\theta}_{t-1} + K_t(y_t - \varphi_t \hat{\theta}_{t-1}^T), \\ K_t &= P_t \varphi_t^T, \qquad P_t = P_{t-1}\varphi_t^T(\lambda + \varphi_t P_{t-1} \varphi_t^T)^{-1}. \end{aligned} \tag{14.18}$$

Notice that the similarity with the Kalman filter (**??**). The parameter λ controls how quickly old data is forgotten There are many other versions of the parameters estimator, of particular interest versions with directional forgetting and square root versions, where the square root of P is updated instead of P itself.

Applications of adaptive control are found in process control, for ship steering and aerospace applications we illustrate by an example.

Example 14.2 An adaptive ship steering auto-pilot
A conventional autopilot for ship steering is based on PID control. The major disturbances are due to wind and waves, which can change significantly during operation of the ship. It is attractive to use an adaptive controller that attempts to model the external disturbances. This can be captures by a model of the form (14.17) with 4 a-parameters, constrained to contain an integrator, and 2 b-parameters parameters. Extensive sea trials have shown that the adaptive autopilot has better performance than the conventional autopilot in normal weather conditions and substantially better performance in bad weather conditions. Figure 14.18 shows results from evaluation of the Steermaster autopilot developed by Kockums and now marketed by Northrop Grumman. In the experiments the conventional and adaptive

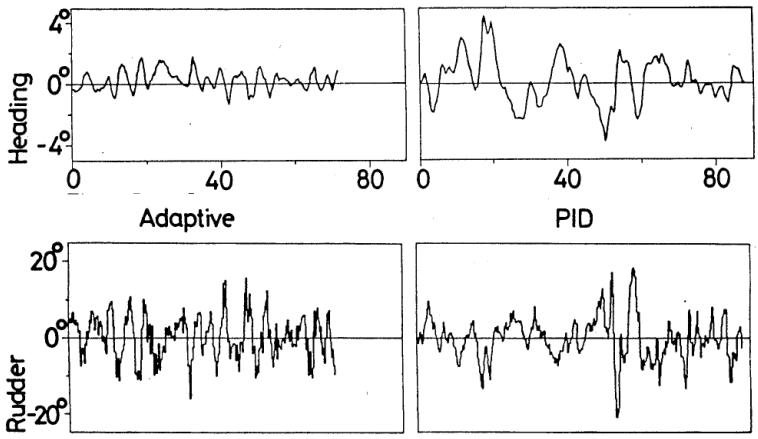


Figure 14.18: Comparison of a conventional autopilot and an adaptive autopilot for ship steering. The experiments were performed on the tanker Seascape, a 225000 ton tanker. The wind velocity was 20 m/s.

autopilots were run repeatedly for about an hour each during normal operation The figure shows that the adaptive autopilot has significantly smaller variations in heading than the conventional autopilot. The difference corresponds to about 3% less fuel consumption. ∇

A difficulty with adaptive control is that parameter estimates are performed when the system is in closed loop. It is then very important where the excitation signals occur. Consider for example the standard feedback loop in Figure 14.17. If the only perturbation on the system is the signal v we have

$$Y(s) = \frac{P(s)}{1+P(s)C(s)}v, \qquad U(s) = -\frac{C(s)P(s)}{1+P(s)C(s)}v,$$

and it thus follows that $Y(s) = -\frac{1}{C(s)}U(s)$ any attempt to find a model relating u and y will thus result in the negative inverse of the controller transfer function. However, if $v = 0$ we have instead

$$Y(s) = \frac{P(s)C(s)}{1+P(s)C(s)}(F(s)R(s) - N(s)), \;\; U(s) = \frac{C(s)}{1+P(s)C(s)}(F(s)R(s) - N(s)v,$$

hence $Y(s) = P(s)U(s)$ and the process model can indeed be estimated.

To obtain reliable estimates of the process parameters it is necessary that there are sufficient variations in the control signal. This can be captured by the notion of persistant excitation, which is defined as follows:. Aasignal $u(t)$ is persistently exciting of order n if

$$U = \lim_{t\to\infty} \frac{1}{T}\int_0^T (A(p)u(k))^2 > 0,$$

for all nonzero polynomials A of of the differential operator $p = d/dt$ with degree

$n-1$ or less. Persistency of excitation roughly tells how many paramters that can be estimated reliably. A constant is persistently exciting of order 1 and permits estimation of one parameter. A sinusoid is persistently exciting of order two.

To have a reliable parameter estimation it is important to be aware of where disturbances enter and to monitor the excitation. Load disturbances of the process are particularly harmful. A scheme for detecting harmful load disturbances is presented in see [THÅ00]. To obtain reliable estimates it is necessary to monitor the excitation of the process and only update parameters when there is sufficient excitation of the process.

An interesting approach to control of uncertain systems was proposed by Feldbaum [], who emphasized that control should be *investigating* as well as *directing* and he coined the term dual control for this property. Feldbaum used optimal stochastic control to obtain a controller that was actively introducing perturbations in the process when the process was not properly excited by natural disturbances. The hyper state of a dual controller is the conditional probability distribution of the regular states of the process and the parameters. The computations of dual controller can only be performed in simple cases because the state of the system is a conditional probability distribution over states and parameters many adhoc schemes to monitor excitation and to introduce perturbations when needed have therefore been produced .

14.6b Learning

A nonlinear function with a learning mechanism is a simple example of a learning system. Learning can be done in two different ways. In supervised learning, the function is created automatically by providing it with a large number of arguments and corresponding values. In unsupervised learning a criterion for good fit is provided and learning is executed by selecting random arguments and changing the values until a good fit is provided. Representation of the learning mechanism are central issues. A simple way to represent a function of several variables is to quantize the variables which we illustrate by an example of unsupervised learning.

Example 14.3 Michie's Boxes
Michie and Chambers developed a simple program for the classical program to stabilize an inverted pendulum. Cart position and velocity and pendulum angle and angular velocity are measured. The control signal is thus a function: $f : \mathbb{R}^4 \to \mathbb{R}$. To implement the system the states were quantized crudely: 5 levels for position and angle and 3 levels for velocities, only two levels were used for the control signal. The control law can thus be represented by a table with 225 entries. The control law is obtained by introducing the notions of life (L) and usage (U). The life of a decision is the number of future decision taken before failure and the usage is the number of decisions taken on entry to this box. There is also a separation of left and right lifes and decisions. Heuristic rules based on the values of the 4 entries LL, RL, LU and RU are used to determine weather to apply a control to the left or to the right. The system is initialized for example by introducing random numbers

in the table. Experiments are run and the table is updated. In a typical experiment the system was able to stabilize the pendulum in 25 minutes after a 60 hour training period, see [MC68]. ∇

Pendulum stabilization is not the best case to demonstrate learning, since it would take less than 60 hours for any reasonably good student to design a controller that stabilizes an inverted pendulum. Control performance in stabilization will also be better because a conventional design can avoid the crude quantization used in Boxes.

Boxes is similar to algoritms game-playing algorithms, where the state is determined and appropriate actions for each state are developed. AlphaGo's algorithm uses a Monte Carlo tree search to find its moves based on knowledge previously "learned" by machine learning, specifically by an artificial neural network (a deep learning method) by extensive training, both from human and computer play.

14.6c Simultaneous Positioning and Mapping

Simultaneous localization and mapping *SLAM* is a learning problem that occurs for moving vehicles. The problem can be approached using algebra and Kalman filtering.

14.6d Neural Networks

A severe drawback of schemes like Boxes is that the nonlinear function is represented by gridding the state variables, which requires very large tables. To have efficient learning schems it is necessary to find more efficient ways to represent nonlinear functions with learning mechanisms. Artificial neural betworks is one representation.

A real neuron has many synapses which permits it to receive inhibitory or exhibitory signals and it will emit a pulse if the received signals are above a certain level. An artificial neuron mimics a real neuron but it operates on continuous variables. A simple model for an artificial neuron is

$$y = f\Big(\sum_{k=1}^{n} w_i u_i\Big), \tag{14.19}$$

where the parameters w_i are weights and the f is a sigmoid shaped function, for example $f(x) = \arctan x$, $f(x) = \frac{1}{1+e^{-x}}$ or lately the ReLU funtion $f(x) = \max(x,0)$. Many other functions have also be used for example: $e^{-x^T Qx}$ and $(1+x^T Qx)^{-1}$.

A neural network is obtained by combining neurons in a network, that is typically layered, as shown in Figure 14.6d. Neural networks can be used to represent functions of several variables. Figure 14.6d represents the function $f : \mathbb{R}^5 \to \mathbb{R}^2$ as

$$f(u_1,u_2,\ldots,u_5) = g\Big(\sum_{i=1}^{5} w_{ij}^{(1)}\, g\big(\sum_{j=1}^{5} w_{ik}^{(2)} u_k\big)\Big), \tag{14.20}$$

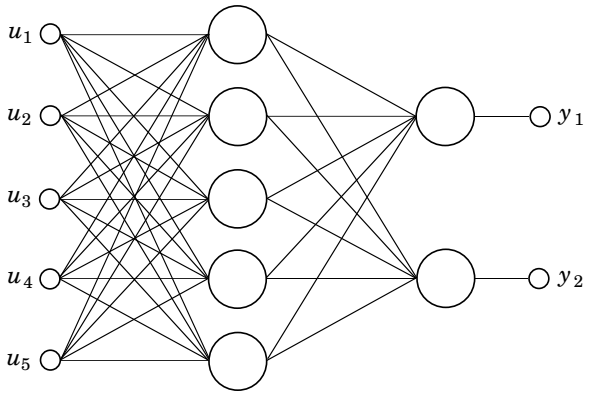


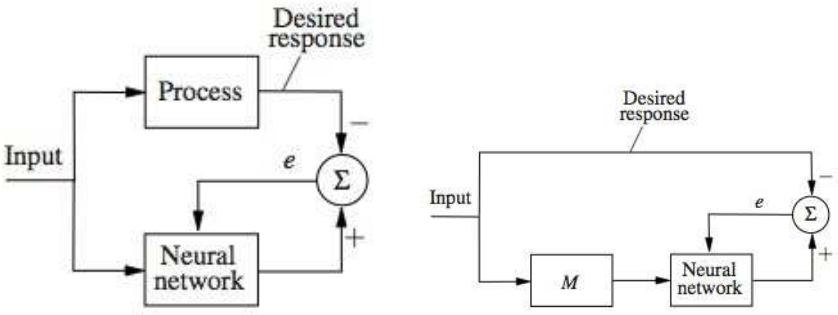
Figure 14.19: Schematic diagram of an neural network with two-layers.

where $w_{jk}^{(1)}$ and $w_{ij}^{(2)}$ are the weights in the first and second layers. An advantage with neural networks is that a function of many variables is represented by linear weights of nondecreasing functions of one variable. The usefulness of such approximations was demonstrated by a by Kolmogorov who showed that any continuous function on a finite cube can be represented by a neural network with one hidden layer

The parameters w_{ij} of a single neuron (14.19) can be trained by the providing a series of values of input x and output y. Learning can be accomplished by feeding the systems known inputs and outputs as shown in Figure 14.6d. A useful feature is that both the function and its invers can be generated from data. Hebb's rule

$$w_i(k+1) = w_i(k) + \gamma u_i^0(k)^0(y^0(k) - y(k)), \tag{14.21}$$

is one algorithm for updating the parameters in a single layer network. The argument k refers to the k:th experiment and the superscript ^{0} refers to the training data. This rule can be interpreted as an approximation of a gradient scheme for



(a) Traning a neural network (b) Training of an inverse model

Figure 14.20: Supervised training neural networks to obtain a process model (a) and the inverse model (b).

minimizing $\sum(y^0(k) - y(k))$, which is similar to to the MIT (14.16) for model reference adaptive control. Parameters of multi-layer neural networks can be updated by similar approximate schemes based on optimization.

Neural networks with many layers, so-called deep learning, has proven very useful. Deep learning has proven very useful for recognition in computer vision and for game-playing. AlphaGo's algorithm uses a Monte Carlo tree search to find its moves based on knowledge previously "learned" by machine learning, specifically by an artificial neural network (a deep learning method) by extensive training, both from playing wiht humans and computers.

14.6e Autonomous Vehicles

One of the interesting areas of research in higher levels of decision-making is autonomous control of cars. Early experiments with autonomous driving were performed by Ernst Dickmanns, who in the 1980s equipped cars with cameras and other sensors [Dic07]. In 1994 his group demonstrated autonomous driving with human supervision on a highway near Paris and in 1995 one of his cars drove autonomously (with human supervision) from Munich to Copenhagen at speeds of up to 175 km/hour. The car was able to overtake other vehicles and change lanes automatically.

This application was explored anew through the DARPA Grand Challenge, a series of competitions sponsored by the U.S. government to build vehicles that can autonomously drive themselves in desert and urban environments. Over the course of approximately 4 years, hundreds of teams design and implemented autonomous vehicles, eventually demonstrating the ability to drive in both desert and urban environments. The capabilities demonstrated in the DARPA Grand Challenge are now being incorporated into advanced vehicle designs by the autonomotive industry, as well as other companies.

Caltech competed in the 2005 and 2007 Grand Challenges using a modified Ford E-350 offroad van nicknamed "Alice." It was fully automated, including electronically controlled steering, throttle, brakes, transmission and ignition. Its sensing systems included multiple video cameras scanning at 10–30 Hz, several laser ranging units scanning at 10 Hz and an inertial navigation package capable of providing position and orientation estimates at 5 ms temporal resolution. Computational resources included 12 high-speed servers connected together through a 1-Gb/s Ethernet switch. The vehicle is shown in Figure 14.21, along with a block diagram of its control architecture.

The software and hardware infrastructure that was developed enabled the vehicle to traverse long distances at substantial speeds. In testing, Alice drove itself more than 500 km in the Mojave Desert of California, with the ability to follow dirt roads and trails (if present) and avoid obstacles along the path. Speeds of more than 50 km/h were obtained in the fully autonomous mode. Substantial tuning of the algorithms was done during desert testing, in part because of the lack of systems-level design tools for systems of this level of complexity. Other competitors in the race (including Stanford, which won the 2005 competition) used algo-

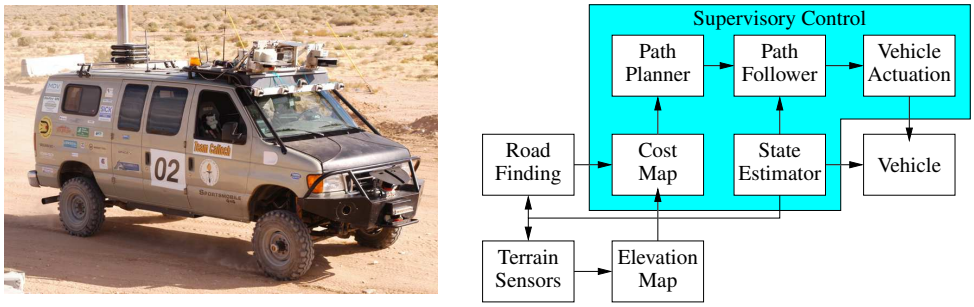


Figure 14.21: DARPA Grand Challenge. "Alice," Team Caltech's entry in the 2005 and 2007 competitions and its networked control architecture [CFG+06]. The feedback system fuses data from terrain sensors (cameras and laser range finders) to determine a digital elevation map. This map is used to compute the vehicle's potential speed over the terrain, and an optimization-based path planner then commands a trajectory for the vehicle to follow. A supervisory control module performs higher-level tasks such as handling sensor and actuator failures.

rithms for adaptive control and learning, increasing the capabilities of their systems in unknown environments. Together, the competitors in the Grand Challenge demonstrated some of the capabilities of the next generation of control systems and highlighted many research directions in control at higher levels of decision making.

A key element of our system is the use of a networked control systems (NCS) architecture that we developed in the first two grand challenge competitions. Building on the open source *Spread* group communications protocol, we have developed a modular software architecture that provides inter-computer communications between sets of linked processes [CFG+06]. This approach allows the use of significant amounts of distributed computing for sensor processing and optimization-based planning, as well as providing a very flexible backbone for building autonomous systems and fault tolerant computing systems. This architecture also allows us to include new components in a flexible way, including modules that make use of planning and sensing modules from the Jet Propulsion Laboratory (JPL).

A schematic of the high-level system architecture that we developed for the Urban Challenge is shown in Figure 14.22. This architecture shares the same underlying approach as the software used for the 2005 Grand Challenge, but with three new elements:

Canonical Software Architecture for mission and contingency management. The complexity and dynamic nature of the urban driving problem make centralized goal and contingency management impractical. For the navigation functions of our system, we have developed a decentralized approach where each module only communicates with the modules directly above and below it in the hierarchy. Each module is capable of handling the faults in its own domain, and anything the module is unable to handle is propagated "up the chain" until the correct level has been reached to resolve the fault or conflict. This architecture is described in more detail

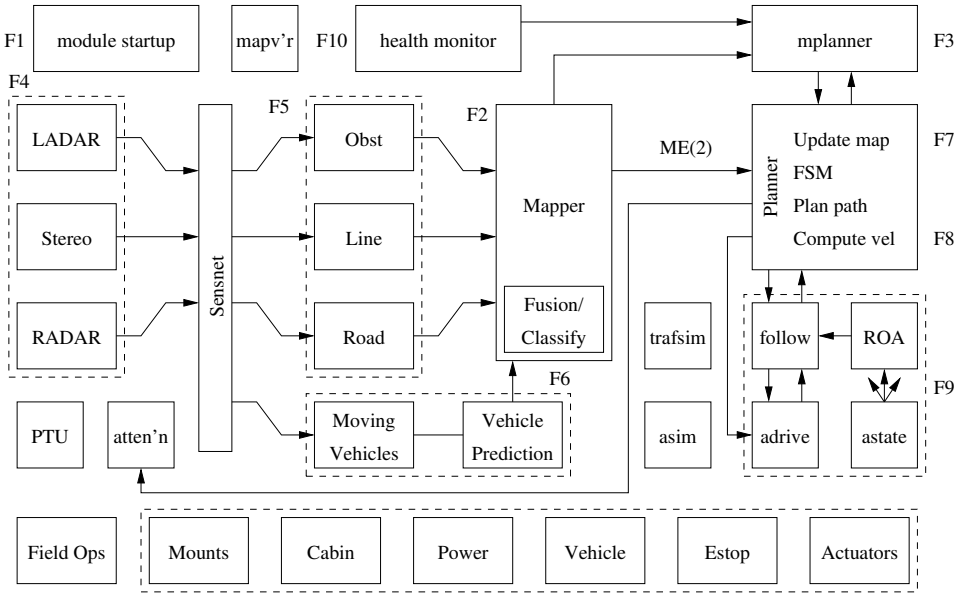


Figure 14.22: Systems architecture for operation of Alice in the 2007 Challenge. The sensing subsystem is responsible for building a representation of the local environment and passing this to the navigation subsystems, which computes and commands the motion of the vehicle. Additional functionality is provided for process and health management, along with data logging and simulation.

in Section **??** and builds on previous work at JPL [DRRS00, IRBM05, Ras01].

The sensing subsystem is responsible for maintaining both a detailed geometric model of the vehicle's environment, as well as a higher level representation of the environment around the vehicle, including knowledge of moving obstacles and road features. It associates sensed data with prior information and broadcasts a structured representation of the environment to the navigation subsystem. The mapping module maintains a vectorized representation of static and dynamic sensed obstacles, as well as detected lane lines, stop lines and waypoints. The map uses a 2.5 dimensional representation where the world is projected into a flat 2D plane, but individual elements may have some non-zero height. Each sensed element is tracked over time and when multiple sensors overlap in field of view, the elements are fused to improve robustness to false positives as well as overall accuracy. These methods are described in more detail in Section **??**.

The planning subsystem determines desired motion of the system, taking into account the current route network and mission goals, traffic patterns and driving rules and terrain features (including static obstacles). This subsystem is also responsible for predicting motion of moving obstacles, based on sensed data and road information, and for implementing defensive driving techniques. The planning problem is divided into three subproblems (route, traffic, and path planning) and implemented in separate modules. This decomposition was well-suited to implementation by a large development team since modules could be developed and

tested using earlier revisions of the code base as well as using simulation environments. Additional details are provided in Section **??**.

14.7 Control Design in Common Application Fields

Control is sometimes called the *hidden technology* because it is successfully used practically everywhere without being noticed []. In this section we will present the role of control in five different applications fields, which provides a flavor of the commercial landscape of controls.

14.7a Aerospace - High Performance Systems and Highly Skilled Users

Control was used early in the aerospace, the Wright brothers flew in 1905 because they had a good insight into dynamics and control, the first autopilot was designed by Sperry in 1914. Today aerospace is a flourishing application area for control. It is dominated by large companies for civil and military markets. The industry produces airplanes, helicopters, satellites, rockets, missiles and quadrocopters and infrastructure for flight control, which includes automatic landing systems. An indication of the size of the business is that about 2500 aircrafts were produced in 2015, generating revenues over 20 billon dollars (2×10^{10} \$), and that is only part of the industry. The companies typically have large central groups for systems and control.

The aerospace systems are operated by highly selected, skilled and well-trained pilots and astronauts, they are trained in simulators and interact with the system by changing operating modes and set-points of controllers. A consequence is that the system are designed so that the user can influence the system directly in many different ways. Sometimes they have taken over operation of the system and save the mission as was done in Apollo 13 [Min08].

The consortium *Future Airborn Capability Environment (FACE)* is a vendor-neutral professional group of industry suppliers, customers, academia, and users formed to define an open avionics environment for all military airborne platforms, see http://www.opengroup.org/face.

The aerospace industry has been a technology driver, hardware and control techniques emerged from the industry. There are extreme requirements on safety, which has lead to the practice of redundant systems and components. The industry pioneered use of simulation and model-based systems engineering, development of high precision accelerometers and gyroscopes. Extremal control was first applied to control of aircraft engines. Wide variations of operating conditions stimulated the development of gain scheduling and adaptive control. Optimal control and Kalman filtering were used extensively in the early space efforts. Nonlinear control was used extensively in control of satellites. Unmanned air vehicles were used operationally as early as 1970.

14.7b Automotive - Complex Systems used by Common Man

The automotive industry is a multi-trillon dollar business (9×10^{12} $ in 2014). It is dominated by six large companies and a many subcontractors. Control is used extensively both in the cars themselves and in the manufacturing of cars. Automobiles are used by ordinary people who interact with the system by changing modes and set points and by direct actuation.

Servo assisted power braking was used in racing cars in 1914 and became commonly used in the 1920s. Computer control was introduced in the late 1970's to cope with the stringent emission requirements. Once computers were introduced they were applied to more functions: suspension control, anti-lock braking systems (ABS), electronic braking systems (EBS) and electronic stability control (ESC). These systems used accelerometers and gyroscopes to control the brakes individually to improve stability and steering. Adaptive cruise control, based on radar sensors, maintains a constant distance to the car in front. The excellent experience with these systems inspired car manufacturers to introduce more sophisticated systems such as collision avoidance and parking assist. Autonomous driving is well on its way. Control is a key element of all these efforts.

Model-based systems engineering is used extensively in the automotive industry to improve the efficiency of engineering. The design of the Toyota Prius is an example where modeling and simulation replaced much prototype testing. Another example is design of climate control systems. The major European car manufacturers and their component suppliers have created an infrastructure for model based design where the suppliers to deliver components with validated dynamic models enabling the car manufacturers to simulated complete systems.

A worldwide development partnership of vehicle manufacturers, suppliers and software companies AUTOSAR (AUTomotive Open System Architecture) was formed in 2003, see http://www.autosar.org.Standards that enable modularity, scalability, transferability and reusability of functions have been created, providing a standardized platform for automotive software systems. The standard enables system-wide configuration and optimization to meet run-time requirements of automotive devices.

The large size of the automotive industry provides a mass market for a wide range of industries to develop components and subsystems. The industry stimulated the development of inexpensive emission sensors, accelerometers and gyroscopes but more importantly the microcontroller and the PLC.

Early manufacturing systems were automation systems controlled by relays for logic and sequencing. General Motors challenged the electronics industry with requirements for a standard machine controller that could replace the relays resulting in a device called *Programmable Logic Controller (PLC)*. The system architecture is based on round robin schedulers with different cycle rates. PLCs were originally programmed in a graphical language called ladder diagrams (LD), which emulated the ladder logic used to describe relay circuits. Later several different programming styles were standardized: function block diagrams (FBD), sequential function charts (SFC) and structured text (ST). The PLCs developed rapidly

and became a standard tool for automation in many industries.

14.7c Process Industry - Complex Systems with Many Different Users

Process control provides automation for a variety of industries such as chemicals, oil refining, pulp and paper, pharmaceuticals power plants and many others. A characteristic feature of the industry is that control and automation is typically delivered by special companies. This begun early with instrument companies that developed sensors and later recorders and controllers, examples are Taylor Instruments, founded in 1851, and Foxboro founded in 1908. By the mid 1930s there were many companies who supplied sensors, actuators and controllers to the process industry.

It is normally difficult to find out how much of the turnover of a business is devototed to control. In process control this data is available, since automation is done by special companies. Control and automation is a 100 billion dollar (10^{11} \$) industry, about 20% of the market is the distributed control system DCS, the rest sensors, actuators, software etc. Five dominating suppliers have more than 50% of the market.

Functions of control and logic and sequencing are essential for process operation. Early process control systems had cabinets with analog controllers for regulation and cabinets with relays for logic and sequencing governing startup, shut down and equipment protection. As technology developed the relays were replaced by programmable logic controllers PLCs and the analog controllers were replaced a DCS.

The DCS is now the standard tool to provide control in the process industry, see Figure **??**. It has facilities for connecting sensors, actuators and algorithms and can be viewed as a tool-box for implementing control systems. It is interesting to note that ExxonMobile has recently contracted Lockheed Martin to specify its next generatrion of distributed control system (DCS) for process control. The system will be open, secure and based on standards, experiences from the consortium FACE will be leveraged.

Process control systems typically have thousands of sensors and actuators, the systems are also widely distributed geographically. Sensors and actuators are connected to the DCS system by standardized networks (IEC 61784).

Valves are commonly used for actuation in process control. It is customary to have cascade control with inner loop analog loops with a valve positioner to reduce effects of friction and nonlinearities at the lowest level of the hierarchy, and feedback loops for control of pressure and temperature and quality variables at the higher levels.

There are several different types of users of the DCS: the plant managers who set production schedules and directs equipment maintainance, the process engineers who select, configure and modify the system, the instrument engineers who tune controllers and maintain sensors and actuators and the operators who supervise the operation of the system, see Figure **??**.

Distributed control systems have many control algorithms that easily can be

configured using graphical interfaces. The control algorithms are implemented by process and instrument engineers both by company personel and by consultants. Controllers are tuned during operation and the system is occasionally reconfigured. Algorithms and languages are standardized by international committees. A wide range of standards for control and automation are set by the International Society of Automation (ISA) and the International Electrical Commission IEC. There are also some standards for communication organized by special groups.

Although PID control was used in many fields the major development of the controller and its tuning procedure were developed in process control. Most of the controllers (typically 97% [DM02b]) are PID controllers, only a small fraction use derivative action. A recent investigation [SLL16] of 100 boiler-turbine units in the Guangdong Province in China showed: 94.4% PI, 3.7% PID and 1.9% advanced controllers.

Control paradigms such as cascade, selector and midrange control are common as are gain scheduling, automatic tuning and model predictive control. Model predictive control emerged from efforts at the Shell oil company to develop effective techniques for control of multivariable processes process control. It was originally called dynamic matrix control [CR80, JBRM99].

14.7d Telecommunication - Billions of Systems

Telecommunications is a cradle of control. Black's invention of the negative feedback amplifier was inspired by the needs to make phone calls over long distances. Intellectual giants like Bode, Nyquist and Shannon developed theoretical foundations of control and communications.

Today the global telecommunication system is said to be the world's largest man-made artifact, the total number of mobile subscriptions in the beginning of 2016 was around 7.4 billion. In many countries the number of mobile subscriptions exceeds the population. *Internet of Things (IoT)* is of particular interest because it enables simple ways of using feedback of long and short distances. It is expected that the number of IoT devices will surpass mobile subscriptions by 2018. Telecommunications is a high pace industry where fancies of the consumer quickly change, making it hard to predict what products will be like 2-3 years from now, continuously changing the rules of the game.

The development of camera modules, GPS modules, accelerometers and gyros for the mobile phone industry has decreased cost for such sensors several orders of magnitudes, because of the large volumes. Reduction in size and improvements in power efficiency of these sensors have been required to fit them into hand held devices with reasonable battery life times. The cheap components have then found several uses in other fields, for example virtual reality.

For production in large volumes cost efficiency is vital, making it economical to put large engineering efforts into cost optimization even for minute details. The requirements from the communication field leads the development of smaller and more energy efficient solutions.

There is extensive standardization in the telecommunications markets, forced

by the fact that the technologies share the common radio frequency spectrum and a carefully controlled use of this limited resource has been needed. It is also highly beneficial for the consumer if devices from different manufacturers can cofunction. The development of technology is coordinated in different groups. The 3GPP consortium develops the standards for mobile communication standard GSM (2G), WCDMA (3G), LTE (4G) as well as future communication standards such as 5G and beyond. Similarly the IEEE working groups develop standards, for example 802.11 for wireless routers. Competing operators, vendors and mobile equipment manufacturers, meet in standardization meetings to discuss and decide on new functionality and performance requirements for future devices. New inventions and intellectual property rights are very important for competitive reasons which leads to both portfolio agreements and patent battles.

Control enters on many levels, from analog electronics where it is used to improve performance such as linearity and power efficiency of power amplifiers, to higher functional levels where control is used to continuously choose suitable system parameters such as transmission power and coding schemes depending on existing communication conditions. PID control is often used, with special care being giving to nonlinear effects such as saturation; (use antiwindup) and gain scheduling and simple adaptive schemes are commonly used.

14.8 Further Reading

A comprehensive treatment of architecture and design of complex systems is given in [**?**]. Much of the development of PID controllers occured in the power and process industries, where also may of the associated bottom-up control architectures like cascade, selector and midranging control appeared. A detailed treatment is given in [ÅH05]. Complementary filtering is well described in []. The internal model control architecture is described in [BT78]. The internal model principle which says that a good controller should contain a model of the process is formulated in [FW76]. The Smith predictor [Smi57a, **?**] is closely related to internal model control because it also uses a parallel model. Posicast control [Smi57b] was also invented by Smith, a professor from University of Berkeley. Decoupling has long been used in many areas of control, the architecture with inverted decoupling is introduced in [Wad97], a state space version if given in [**?**]. The relative gain array, a simple way to quickly estimate the interaction, was introduced in [Bri66]. A detailed treatment is found in [McA83]. Gainscheduling is widely used in practice, overviews are presented in[LWE00, SA92, Rug91]. Adaptive control emerged from the desire to avoid gain scheduling in flight control and the dream of having a controller that can automatically adjust itself to good performance in the process industry. Early pioneering work is found in [Gre59]. Current knowlege is found in [ÅW08, CS08, LW13]. System identification is a key element of adaptive control the book [Lju87] is an excellent reference. In spite of advances in adaptive control gain scheduling is still the dominant control scheme for flight control, [Ste80], one reason is a significant development of air-data sensors. The process engineers

dream of having a simple universal controller that self-adjust to good robust performance is still a dream, the relay auto-tuner for PID control is a partial answer. Rosenblatt's perceptron [Ros62] was the first neural network, which was used to separate hyperplanes in pictures. An analog version the Addaline [**?**] invented by Widrow was used in simple adaptive systems and for noise cancellation. Back propagation, which is closely related to dynamics programming [Bel57], made it possible to train networks with many hidden layers [RHW86]. An early application to picture classification is given in [LBD+89], it used deep structures and convolution networks. A nice introduction to the effects of many layers is given in [SZ14]. The power of deep learning was demonstrated in [KSH12], a million pictures were classified in 1000 classes.

Research on control of multi-vehicle systems performing cooperative tasks dates back to the late 1980s, initially beginning in the field of mobile robotics (see [**?**] for a more detailed history). Aided by the development of inexpensive and reliable wireless communications systems, research in this area increased substantially in the 1990s. California's Partners for Advanced Transit and Highways (PATH) project [**?**] demonstrated multiple automobiles driving together in "platoons" and this was quickly followed by other highway automation projects [**?**, **?**]. In the late 1990s and early 2000s, cooperative control of multiple aircraft, especially unmanned aerial vehicles (UAVs), became a highly active research area in the United States [**?**], spurring further advances. Over the last decade this research area has blossomed, with many new systems being proposed in application areas ranging from military battle systems to mobile sensors networks to commercial highway and air transportation systems.

Bibliography

[Abk69] M. A. Abkowitz. *Stability and Motion Control of Ocean Vehicles*. MIT Press, Cambridge, MA, 1969.

[Ack72] J. Ackermann. Der Entwurf linearer Regelungssysteme im Zustandsraum. *Regelungstechnik und Prozessdatenverarbeitung*, 7:297–300, 1972.

[Ack85] J. Ackermann. *Sampled-Data Control Systems*. Springer, Berlin, 1985.

[Agn76] C. E. Agnew. Dynamic modeling and control of congestion-prone systems. *Operations Research*, 24(3):400–419, 1976.

[ÅH05] K. J. Åström and T. Hägglund. *Advanced PID Control*. ISA—The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC, 2005.

[Ahl66] L. V. Ahlfors. *Complex Analysis*. McGraw-Hill, New York, 1966.

[ÅHL94] K. J. Åström, C. C. Hang, and B. C. Lim. A new smith predictor for controlling a process with an integrator and long dead-time. *IEEE Transactions on Automatic Control*, 32:343–345, 1994.

[ÅKL05] K. J. Åström, R. E. Klein, and A. Lennartsson. Bicycle dynamics and control. *IEEE Control Systems Magazine*, 25(4):26–47, 2005.

[Alu15] R. Alur. *Principles of Cyber-Physical Systems*. MIT Press, 2015.

[AM90] B. D. O. Anderson and J. B. Moore. *Optimal Control Linear Quadratic Methods*. Prentice Hall, Englewood Cliffs, NJ, 1990. Republished by Dover Publications, 2007.

[Apo67] T. M. Apostol. *Calculus,* Vol. II: *Multi-Variable Calculus and Linear Algebra with Applications*. Wiley, New York, 1967.

[Apo69] T. M. Apostol. *Calculus,* Vol. I: *One-Variable Calculus with an Introduction to Linear Algebra*. Wiley, New York, 1969.

[Ari94] R. Aris. *Mathematical Modeling Techniques*. Dover, New York, 1994. Originally published by Pitman, 1978.

[Arn78] V. I. Arnold. *Mathematical Methods in Classical Mechanics*. Springer, New York, 1978.

[Arn87] V. I. Arnold. *Ordinary Differential Equations*. MIT Press, Cambridge, MA, 1987. 10th printing 1998.

[AS82] R. H. Abraham and C. D. Shaw. *Dynamics—The Geometry of Behavior,* Part 1: *Periodic Behavior*. Aerial Press, Santa Cruz, CA, 1982.

[ASMN03] M. Atkinson, M. Savageau, J. Myers, and A. Ninfa. Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. *Cell*, 113(5):597–607, 2003.

[Åst00] K. J. Åström. Limitations on control system performance. *European Journal on Control*, 6(1):2–20, 2000.

[Åst06] K. J. Åström. *Introduction to Stochastic Control Theory*. Dover, New York, 2006. Originally published by Academic Press, New York, 1970.

[Ath75] D. P. Atherton. *Nonlinear Control Engineering*. Van Nostrand, New York, 1975.

[AVK87] A. A. Andronov, A. A. Vitt, and S. E. Khaikin. *Theory of Oscillators*. Dover, New York, 1987.

[ÅW97] K. J. Åström and B. Wittenmark. *Computer-Control Systems: Theory and Design*. Prentice Hall, Englewood Cliffs, NJ, 3rd edition, 1997.

[ÅW08] K. J. Åström and B. Wittenmark. *Adaptive Control*. Dover, New York, 2nd edition, 2008. Originally published by Addison Wesley, 1995.

[BÅ70] R. Bellman and K. J. Åström. On structural identifiability. *Mathematical Biosciences*, 7:329–339, 1970.

[Bas01] T. Basar, editor. *Control Theory: Twenty-five Seminal Papers (1932–1981)*. IEEE Press, New York, 2001.

[BB91] T. Basar and P. Bernhard. *H^∞-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhauser, Boston, 1991.

[BC48] G. S. Brown and D. P. Campbell. *Principles of Servomechanims*. Wiley, New York, 1948.

[BD04] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations*. Wiley, New York, 2004.

[BdH+07] J. W. Burdick, N. duToit, A. Howard, C. Looman, J. Ma, R. M. Murray, and T. Wongpiromsarn. Sensing, navigation and reasoning technologies for the DARPA Urban Challenge. Available online at http://gc.caltech.edu/media/papers/dgc07-final.pdf, DARPA Urban Challenge Final Report, 2007.

[Bec05] J. Bechhoefer. Feedback for physicists: A tutorial essay on control. *Reviews of Modern Physics*, 77:783–836, 2005.

[Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton NJ, 1957.

[Ben79] S. Bennett. *A History of Control Engineering: 1800–1930*. Peter Peregrinus, Stevenage, 1979.

[Ben93] S. Bennett. *A History of Control Engineering: 1930–1955*. Peter Peregrinus, Stevenage, 1993.

[Ber54] L. L. Beranek. *Acoustics*. McGraw-Hill, New York, 1954.

[Ber89] R. N. Bergman. Toward physiological understanding of glucose tolerance: Minimal model approach. *Diabetes*, 38:1512–1527, 1989.

[BG68] B. Brawn and F. Gustavson. Program behavior in a paging environment. *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 1019–1032, 1968.

[BG87] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, 1987.

[BH75] A. E. Bryson, Jr. and Y.-C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Wiley, New York, 1975.

[Bia95] B. Bialkowski. Process control sample problems. In N. J. Sell, editor, *Process Control Fundamentals for the Pulp & Paper Industry*. Tappi Press, Norcross, GA, 1995.

[BK64] R. E. Bellman and R. Kalaba. *Selected Papers on Mathematical Trends in Control Theory*. Dover, New York, 1964.

[Bla34] H. S. Black. Stabilized feedback amplifiers. *Bell System Technical Journal*, 13:1–2, 1934.

[Bla77] H. S. Black. Inventing the negative feedback amplifier. *IEEE Spectrum*, pages 55–60, 1977.

[Bla91] J. H. Blakelock. *Automatic Control of Aircraft and Missiles*. Addison-Wesley, Cambridge, MA, 2nd edition, 1991.

[Bli90] G. Blickley. Modern control started with Ziegler-Nichols tuning. *Control Engineering*, 37:72–75, 1990.

[BM99] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407 – 427, 1999.

[Boa16] BKCASE Editorial Board, editor. *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*. Stevens Institute of Technology, v. 1.6 edition, 2016. Accessed 27 August 2016, http://sebokwiki.org.

[Bod45] H. W. Bode. *Network Analaysis and Feedback Amplifier Design*. Van Nostrand, New York, 1945.

[Bod60] H. W. Bode. Feedback—The history of an idea. In *Symposium on Active Networks and Feedback Systems*. Polytechnic Institute of Brooklyn, New York, 1960. Reprinted in [BK64].

[BP96] M. B. Barron and W. F. Powers. The role of electronic controls for future automotive mechatronic systems. *IEEE Transactions on Mechatronics*, 1(1):80–89, 1996.

[Bri66] E. H. Bristol. On a new measure of interactions for multivariable process control. *IEEE Transactions on Automatic Control*, 11(1):133–134, 1966.

[Bro70] R. W. Brockett. *Finite Dimensional Linear Systems*. Wiley, New York, 1970.

[Bro00] R. W. Brockett. New issues in the mathematics of control. In B. Engquist and W. Schmid, editors, *Mathematics Unlimited—2001 and Beyond*, pages 189–220. Springer-Verlag, Berlin, 2000.

[BRS60] J. F. Blackburn, G. Reethof, and J. L. Shearer. *Fluid Power Control*. MIT Press, Cambridge, MA, 1960.

[BT78] C. Brosilov and M. Tong. Inferential control systems. *AIChE Journal*, 24(3):457–465, 1978.

[Can03] R. H. Cannon. *Dynamics of Physical Systems*. Dover, New York, 2003. Originally published by McGraw-Hill, 1967.

[CCG+02] A. Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *Proceedings of the 14th International Conference on Computer Aided Verification*, pages 359–364. Springer-Verlag, 2002.

[CD75] R. F. Coughlin and F. F. Driscoll. *Operational Amplifiers and Linear Integrated Circuits*. Prentice Hall, Englewood Cliffs, NJ, 6th edition, 1975.

[CD91] F. M. Callier and C. A. Desoer. *Linear System Theory*. Springer-Verlag, London, 1991.

[CEHM10] M. Campbell, M. Egerstedt, J. P. How, and R. M Murray. Autonomous driving in urban environments: Approaches, lessons and challenges. *Philosophical Transactions of the Royal Society – A*, 368(1928), 2010.

[CFG+06] L. B. Cremean, T. B. Foote, J. H. Gillula, G. H. Hines, D. Kogan, K. L. Kriechbaum, J. C. Lamb, J. Leibs, L. Lindzey, C. E. Rasmussen, A. D. Stewart, J. W. Burdick, and R. M. Murray. Alice: An information-rich autonomous vehicle for high-speed desert navigation. *Journal of Field Robotics*, 23(9):777–810, 2006.

[CJ59] H. S. Carslaw and J. C. Jaeger. *Conduction of Heat in Solids*. Clarendon Press, Oxford, UK, 2nd edition, 1959.

[CM51] H. Chestnut and R. W. Mayer. *Servomechanisms and Regulating System Design,* Vol. 1. Wiley, New York, 1951.

[CR80] C. R. Cutler and B. C. Ramaker. Dynamic matrix control—A computer control algorithm. In *Proceedings Joint Automatic Control Conference*, San Francisco, CA, 1980.

[Cro75] Crocus. *Systemes d'Exploitation des Ordinateurs*. Dunod, Paris, 1975.

[CS08] G. C. and K. S. Sin. *Adaptive Filtering Prediction and Control*. Dover, New York, 2008. Originally published by Prentice Hall, 1984.

[CT84] C. Cobelli and G. Toffolo. Model of glucose kinetics and their control by insulin, compartmental and non-compartmental approaches. *Mathematical Biosciences*, 72(2):291–316, 1984.

[DB04] R. C. Dorf and R. H. Bishop. *Modern Control Systems*. Prentice Hall, Upper Saddle River, NJ, 10th edition, 2004.

[DFT92] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback Control Theory*. Macmillan, New York, 1992.

[DGH+02] Y. Diao, N. Gandhi, J. L. Hellerstein, S. Parekh, and D. M. Tilbury. Using MIMO feedback control to enforce policies for interrelated metrics with application to the Apache web server. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, pages 219–234, 2002.

[DGKF89] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[DH85] J. P. Den Hartog. *Mechanical Vibrations*. Dover, New York, 1985. Reprint of 4th ed. from 1956; 1st ed. published in 1934.

[Dic07] E. D. Dickmanns. *Dynamic Vision for Perception and Control of Motion*. Springer, Berlin, 2007.

[dJ02] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9:67–103, 2002.

[DM02a] L. Desborough and R. Miller. Increasing customer value of industrial control performance monitoring—Honeywell's experience. In *Sixth International Conference on Chemical Process Control*. AIChE Symposium Series Number 326 (Vol. 98), 2002.

[DM02b] L. Desbourough and R. Miller. Increasing customer value of industrial control performance monitoring - Honeywell's experience. In *Sixth International Conference on Chemical Process Control*. AIChE Symposium Series Number 326 (Volume 98), 2002.

[DM14] D. Del Vecchio and R. M. Murray. *Biomolecular Feedback Systems*. Princeton University Press, 2014.

[Dos68] F. H. Dost. *Grundlagen der Pharmakokinetik*. Thieme Verlag, Stuttgart, 1968.

[Doy78] J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.

[Dra55] C. S. Draper. Flight control. *Jorunal Royal Aeronautical Society*, 59(July):451–477, 1955. 45th Wilber Wright Memorial Lecture.

[DRRS00] D. Dvorak, R. D. Rasmussen, G. Reeves, and A. Sacks. Software architecture themes in jpl's mission data system. In *Proceedings of 2000 IEEE Aerospace Conference*, 2000.

[Dub57] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[Dys04] F. Dyson. A meeting with Enrico Fermi. *Nature*, 247(6972):297, 2004.

[EG05] S. P. Ellner and J. Guckenheimer. *Dynamic Models in Biology*. Princeton University Press, Princeton, NJ, 2005.

[EHBM08] B. Eisenhower, G. Hagen, A. Banaszuk, and I. Mezič. Passive control of limit cycle oscillations in a thermoacoustic system using asymmetry. *Journal of Applied Mechanics*, 75:011021–1–011021–9, 2008.

[EKR03] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In *Power-Aware Computer Systems*, pages 179–197. Springer, 2003.

[EL00] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, 2000.

[Ell94] J. R. Ellis. *Vehicle Handling Dynamics*. Mechanical Engineering Publications, London, 1994.

[ESGK02] H. El-Samad, J. P. Goff, and M. Khammash. Calcium homeostasis and parturient hypocalcemia: An integral feedback perspective. *Journal of Theoretical Biology*, 214:17–29, 2002.

[FCF+06] P. G. Fabietti, V. Canonico, M. O. Federici, M. Benedetti, and E. Sarti. Control oriented model of insulin and glucose dynamics in type 1 diabetes. *Medical and Biological Engineering and Computing*, 44:66–78, 2006.

[FLMR92] M. Fliess, J. Levine, P. Martin, and P. Rouchon. On differentially flat nonlinear systems. *Comptes Rendus des Séances de l'Académie des Sciences,* Serie I, 315:619–624, 1992.

[For61] J. W. Forrester. *Industrial Dynamics*. MIT Press, Cambridge, MA, 1961.

[Fou07] J. B. J. Fourier. On the propagation of heat in solid bodies. Memoir, read before the Class of the Instut de France, 1807.

[FPEN05] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, Upper Saddle River, NJ, 5th edition, 2005.

[Fra87] B. A. Francis. *A Course in $\mathcal{H}_\infty$ Control*. Springer-Verlag, Berlin, 1987.

[Fra07] A. Fradkov. *Cybernetical Physics: From Control of Chaos to Quantum Control*. Springer, Berlin, 2007.

[Fri04] B. Friedland. *Control System Design: An Introduction to State Space Methods*. Dover, New York, 2004.

[FW76] B. A. Francis and M. W. Wonham. The internal model principle of control theory. *Automatica*, 12(5):492–500, 1976.

[Gan60] F. R. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, 1960.

[GB42] M. A. Gardner and J. L. Barnes. *Transients in Linear Systems*. Wiley, New York, 1942.

[GF71] L. Gunkel and G. F. Franklin. A general solution for linear sampled data systems. *IEEE Transactions on Automatic Control*, AC-16:767–775, 1971.

[GGS01] G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Prentice Hall, Upper Saddle River, NJ, 2001.

[GH83] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer-Verlag, Berlin, 1983.

[Gil63] E. Gilbert. Controllability and observability in multivariable control systems. *SIAM Journal of Control*, 1(1):128–151, 1963.

[GL95] M. Green and D. J. N. Limebeer. *Linear Robust Control*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[GM61] D. Graham and D. McRuer. *Analysis of Nonlinear Control Systems*. Wiley, New York, 1961.

[God83] K. Godfrey. *Compartment Models and Their Application*. Academic Press, New York, 1983.

[Gol53] H. Goldstein. *Classical Mechanics*. Addison-Wesley, Cambridge, MA, 1953.

[Gol70] S. W. Golomb. Mathematical models—Uses and limitations. *Simulation*, 4(14):197–198, 1970.

[GP82] M. Giobaldi and D. Perrier. *Pharmacokinetics*. Marcel Dekker, New York, 2nd edition, 1982.

[GPD59] J. C. Gille, M. J. Pelegrin, and P. Decaulne. *Feedback Control Systems; Analysis, Synthesis, and Design*. McGraw-Hill, New York, 1959.

[Gre59] P. C. Gregory, editor. *Proc. Self Adaptive Flight Control Symposium*. Wright Air Development Center, Wright-Patterson Air Force Base, Ohio, 1959.

[Gui63] E. A. Guillemin. *Theory of Linear Physical Systems*. MIT Press, Cambridge, MA, 1963.

[Hah67] W. Hahn. *Stability of Motion*. Springer, Berlin, 1967.

[HB90] J. K. Hedrick and T. Batsuen. Invariant properties of automobile suspensions. In *Proceedigns of the Institution of Mechanical Engineers*, volume 204, pages 21–27, London, 1990.

[HD95] M. B. Hoagland and B. Dodson. *The Way Life Works*. Times Books, New York, 1995.

[HDPT04] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. *Feedback Control of Computing Systems*. Wiley, New York, 2004.

[Her04] D. V. Herlihy. *Bicycle—The History*. Yale University Press, New Haven, CT, 2004.

[HH52] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117(500–544), 1952.

[HMTG00] C. V. Hollot, V. Misra, D. Towsley, and W-B. Gong. A control theoretic analysis of RED. In *Proceedings of IEEE Infocom*, pages 1510–1519, 2000.

[Hol03] G. J. Holzmann. *The SPIN Model Checker*. Addison-Wesley, 2003.

[Hor63] I. M. Horowitz. *Synthesis of Feedback Systems*. Academic Press, New York, 1963.

[Hor75] I. M. Horowitz. Superiority of transfer function over state-variable methods in linear, time-invariant feedback system design. *IEEE Transactions on Automatic Control*, AC-20(1):84–97, 1975.

[Hor91] I. M. Horowitz. Survey of quantitative feedback theory. *International Journal of Control*, 53:255291, 1991.

[HSH15] R. J. Hawkins, J. K. Speakes, and D. E. Hamilton. Monetary policy and PID control. *Journal of Economic Interaction and Coordination*, 10(1):183–197, 2015.

[Hug93] T. P. Hughes. *Elmer Sperry: Inventor and Engineer*. John Hopkins University Press, Baltimore, MD, 1993.

[HW00] D. Hanahan and R. A. Weinberg. The hallmarks of cancer. *Cell*, 100:57–70, 2000.

[IRBM05] M. Ingham, R. Rasmussen, M. Bennett, and A. Moncada. Engineering complex embedded systems with state analysis and the mission data system. *J. Aerospace Computing, Information and Communication*, 2, 2005.

[Isi95] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, Berlin, 3rd edition, 1995.

[Ito70] M. Ito. Neurophysiological aspects of the cerebellar motor system. *International Journal of Neurology*, 7:162–178, 1970.

[Jac72] J. A. Jacquez. *Compartment Analysis in Biology and Medicine*. Elsevier, Amsterdam, 1972.

[Jac95] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 25:157–173, 1995.

[Jan14] P. K. Janert. *Feedback Control for Computer Scientists*. O'Reilly Media, 2014.

[JBRM99] J. B. J. B Rawlings and D. Q. Mayne. *Model Predictive Control Theory and Design*. Nob Hill Publishing, Madison, WI, 1999.

[JNP47] H. James, N. Nichols, and R. Phillips. *Theory of Servomechanisms*. McGraw-Hill, New York, 1947.

[JT61] P. D. Joseph and J. T. Tou. On linear control theory. *Transactions of the AIEE*, 80(18), 1961.

[Jun02] W. G. Jung, editor. *Op Amp Applications*. Analog Devices, Norwood, MA, 2002.

[Kal60] R. E. Kalman. Contributions to the theory of optimal control. *Boletin de la Sociedad Matématica Mexicana*, 5:102–119, 1960.

[Kal61a] R. E. Kalman. New methods and results in linear prediction and filtering theory. Technical Report 61-1, Research Institute for Advanced Studies (RIAS), Baltimore, MD, February 1961.

[Kal61b] R. E. Kalman. On the general theory of control systems. In *Proceedings of the First IFAC Congress on Automatic Control, Moscow, 1960*, volume 1, pages 481–492. Butterworths, London, 1961.

[KB61] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Transactions of the ASME (Journal of Basic Engineering)*, 83 D:95–108, 1961.

[Kel85] F. P. Kelly. Stochastic models of computer communication. *Journal of the Royal Statistical Society*, B47(3):379–395, 1985.

[Kel94] K. Kelly. *Out of Control*. Addison-Wesley, Reading, MA, 1994. Available at http://www.kk.org/outofcontrol.

[KFA69] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in Mathematical System Theory*. McGraw-Hill, New York, 1969.

[KG02] B. C. Kuo and F. Golnaraghi. *Automatic Control Systems*. Wiley, New York, 8th edition, 2002.

[Kha01] H. K. Khalil. *Nonlinear Systems*. Macmillan, New York, 3rd edition, 2001.

[KHN63] R. E. Kalman, Y. Ho, and K. S. Narendra. *Controllability of Linear Dynamical Systems*, volume 1 of *Contributions to Differential Equations*. Wiley, New York, 1963.

[Kit95] C. Kittel. *Introduction to Solid State Physics*. Wiley, New York, 1995.

[Kit04] Hiroaki Kitano. Biological robustness. *Nature Reviews Genetics*, 5(11):826–837, 2004.

[KKK95] M. Krstić, I. Kanellakopoulos, and P. Kokotović. *Nonlinear and Adaptive Control Design*. Wiley, 1995.

[Kle75] L. Kleinrock. *Queuing Systems,* Vols. I and II. Wiley-Interscience, New York, 2nd edition, 1975.

[KN00] U. Kiencke and L. Nielsen. *Automotive Control Systems: For Engine, Driveline, and Vehicle*. Springer, Berlin, 2000.

[KNP11] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, pages 585–591. Springer, 2011.

[Kra63] N. N. Krasovski. *Stability of Motion*. Stanford University Press, Stanford, CA, 1963.

[Kru09] Paul Krugman. *The Return of Depression Economics and the Crisis of 2008*. W. W. Norton & Company, New York, 2009.

[KS01] J. Keener and J. Sneyd. *Mathematical Physiology*. Springer, New York, 2001.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Kum01] P. R. Kumar. New technological vistas for systems and control: The example of wireless networks. *Control Systems Magazine*, 21(1):24–37, 2001.

[Kun93] P. Kundur. *Power System Stability and Control*. McGraw-Hill, New York, 1993.

[KV86] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice Hall, Englewood Cliffs, NJ, 1986.

[LaS60] J. P. LaSalle. Some extensions of Lyapunov's second method. *IRE Transactions on Circuit Theory*, CT-7(4):520–527, 1960.

[LBD+89] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R. E. Howard, W. Hubbrd, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[Lew03] A. D. Lewis. A mathematical approach to classical control. Technical report, Queens University, Kingston, Ontario, 2003.

[Lju87] L. Ljung. *System Identification—Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 1987. Second Edition 1999.

[Lju99] L. Ljung. *System Indentification – Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[LPD02] S. H. Low, F. Paganini, and J. C. Doyle. Internet congestion control. *IEEE Control Systems Magazine*, pages 28–43, February 2002.

[LPW+02] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle. Dynamics of TCP/RED and a scalable control. In *Proceedings of IEEE Infocom*, pages 239–248, 2002.

[LS15] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*. http://LeeSeshia.org, 2015. ISBN 978-1-312-42740-2.

[Lun05] K. H. Lundberg. History of analog computing. *IEEE Control Systems Magazine*, pages 22–28, March 2005.

[LW13] E. Lavretsky and K. A. Wise. *Robust and Adaptive Control with Aerospace Applications*. Springer, London, 2013.

[LWE00] D. J. Leith and Leithead W. E. Survey of gain-scheduling analysis and design. *International Journal on Control*, 73(11):1001–1025, 2000.

[MÅB+03] R. M. Murray, K. J. Åström, S. P. Boyd, R. W. Brockett, and G. Stein. Future directions in control in an information-rich world. *Control Systems Magazine*, April 2003.

[Mac37] D. A. MacLulich. *Fluctuations in the Numbers of the Varying Hare (Lepus americanus)*. University of Toronto Press, 1937.

[Mac45] L.A. MacColl. *Fundamental Theory of Servomechanims*. Van Nostrand, Princeton, NJ, 1945. Dover reprint 1968.

[Mac89] J. M. Maciejowski. *Multivariable Feedback Design*. Addison Wesley, Reading, MA, 1989.

[Mal59] J. G. Malkin. *Theorie der Stabilität einer Bewegung*. Oldenbourg, München, 1959.

[Man02] R. Mancini. *Op Amps for Everyone*. Texas Instruments, Houston. TX, 2002.

[May70] O. Mayr. *The Origins of Feedback Control*. MIT Press, Cambridge, MA, 1970.

[MC68] D. Michie and R. A. Chambers. Boxes: An experiment in adaptive control. In E. Dale and D. Michie (Eds)., editors, *Machine Intelligence 2*, pages 137–152. Oliver and Boyd, Edinburgh, 1968.

[McA83] Thomas J. McAvoy. *Interaction Analysis - Principles and Applications*. Instrument Society of America, Research Triangle Park, NC, 1983.

[McA83] T. McAvoy. *Interaction Analysis*. ISA, Research Triangle Park, NC, 2083.

[McF53] M. W. McFarland, editor. *The Papers of Wilbur and Orville Wright*. McGraw-Hill, New York, 1953.

[MG90] D. C. McFarlane and K. Glover. *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*. Springer, New York, 1990.

[MH98] J. E. Marsden and M. J. Hoffmann. *Basic Complex Analysis*. W. H. Freeman, New York, 1998.

[Mil66] H. T. Milhorn. *The Application of Control Theory to Physiological Systems*. Saunders, Philadelphia, 1966.

[Min02] D. A. Mindel. *Between Human and Machine: Feedback, Control, and Computing Before Cybernetics*. Johns Hopkins University Press, Baltimore, MD, 2002.

[Min08] D. A. Mindel. *Digital Apollo: Human and Machine in Spaceflight*. The MIT Press, Cambridge, MA, 2008.

[MLK06] A. Makroglou, J. Li, and Y. Kuang. Mathematical models and software tools for the glucose-insulin regulatory system and diabetes: An overview. *Applied Numerical Mathematics*, 56:559–573, 2006.

[MLS94] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[MR94] J. E. Marsden and T. S. Ratiu. *Introduction to Mechanics and Symmetry*. Springer-Verlag, New York, 1994.

[MRRS00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[Mur03] R. M. Murray, editor. *Control in an Information Rich World: Report of the Panel on Future Directions in Control, Dynamics and Systems*. SIAM, Philadelphia, 2003.

[Mur04] J. D. Murray. *Mathematical Biology,* Vols. I and II. Springer-Verlag, New York, 3rd edition, 2004.

[Nah88] P. J. Nahin. *Oliver Heaviside: Sage in Solitude: The Life, Work and Times of an Electrical Genius of the Victorian Age*. IEEE Press, New York, 1988.

[NS99] H. Nijmeijer and J. M. Schumacher. Four decades of mathematical system theory. In J. W. Polderman and H. L. Trentelman, editors, *The Mathematics of Systems and Control: From Intelligent Control to Behavioral Systems*, pages 73–83. University of Groningen, 1999.

[Nyq32] H. Nyquist. Regeneration theory. *Bell System Technical Journal*, 11:126–147, 1932.

[Nyq56] H. Nyquist. The regeneration theory. In R. Oldenburger, editor, *Frequency Response*, page 3. MacMillan, New York, 1956.

[Oga01] K. Ogata. *Modern Control Engineering*. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2001.

[Old56] R. Oldenburger, editor. *Frequency Response*. MacMillan, New York, 1956.

[OSFM07] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[PB86] G. Pacini and R. N. Bergman. A computer program to calculate insulin sensitivity and pancreatic responsivity from the frequently sampled intraveneous glucose tolerance test. *Computer Methods and Programs in Biomedicine*, 23:113–122, 1986.

[Phi48] G. A. Philbrick. Designing industrial controllers by analog. *Electronics*, 21(6):108–111, 1948.

[PN00] W. F. Powers and P. R. Nicastri. Automotive vehicle control challenges in the 21st century. *Control Engineering Practice*, 8:605–618, 2000.

[PPP02] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. SOSTOOLS: Sum of squares optimization toolbox for MATLAB, 2002. Available from http://www.cds.caltech.edu/sostools.

[Pto14] Claudius Ptolemaeus, editor. *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, 2014.

[Pyt] Python control systems library. Available from http://python-control.org.

[QB97] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In J.C. Kantor, C.E. Garcia, and B. Carnahan, editors, *Fifth International Conference on Chemical Process Control*, pages 232–256, 1997.

[Ras01] R. D. Rasmussen. Goal-based fault tolerance for space systems using the Mission Data System. In *IEEE Aerospace Conference*, 2001.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating error. *Nature*, 323(6088):533–536, 1986.

[Rig63] D. S. Riggs. *The Mathematical Approach to Physiological Problems*. MIT Press, Cambridge, MA, 1963.

[RM71] H. H. Rosenbrock and P. D. Moran. Good, bad or optimal? *IEEE Transactions on Automatic Control*, AC-16(6):552–554, 1971.

[Ros62] F. Rosenblatt. *Principles of Neurodynamics*. Spartan Books, New York, 1962.

[RST12] G. Rafal, R. G. Sanfelice, and A. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, 2012.

[Rug91] W. J. Rugh. Analytic framework for gain scheduling. *Control Systems Magazine*, 11(1):79–84, 1991.

[Rug95] W. J. Rugh. *Linear System Theory*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1995.

[SA92] J. S. Shamma and M. Athans. Gain scheduling: potential hazards and possible remedies. *Control Systems Magazine*, 12(3):101–107, 1992.

[SA14] T. Samad and A. M. Annaswamy, editors. *The Impact of Control Technology*. IEEE Control Systems Society, 2nd edition edition, 2014. Available at www.ieeecss.og.

[SÅD+07] G. Schitter, K. J. Åström, B. DeMartini, P. J. Thurner, K. L. Turner, and P. K. Hansma. Design and modeling of a high-speed AFM-scanner. *IEEE Transactions on Control System Technology*, 15(5):906–915, 2007.

[Sar91] D. Sarid. *Atomic Force Microscopy*. Oxford University Press, Oxford, UK, 1991.

[Sas99] S. Sastry. *Nonlinear Systems*. Springer, New York, 1999.

[SC92] R. Shishko and R. G. Chamberlain. NASA systems engineering handbook. Technical report, National Aeronautics and Space Administration, 1992.

[Sch87] M. Schwartz. *Telecommunication Networks*. Addison Wesley, Reading, MA, 1987.

[Sch01] G. Schitter. High performance feedback for fast scanning atomic force microscopes. *Review of Scientific Instruments*, 72(8):3320–3327, 2001.

[SEM04] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp. *Process Dynamics and Control*. Wiley, Hoboken, NJ, 2nd edition, 2004.

[Sen01] S. D. Senturia. *Microsystem Design*. Kluwer, Boston, MA, 2001.

[Sha01] L. Sha. Using simplicity to control complexity. *IEEE Software*, 18:20–28, 2001.

[Shi96] F. G. Shinskey. *Process-Control Systems. Application, Design, and Tuning*. McGraw-Hill, New York, 4th edition, 1996.

[Sim56] H. A. Simon. Dynamic programming under uncertainty with a quadratic criterion function. *Econometrica*, 24:74–81, 1956.

[SLL16] Li Sun, Donghai Li, and Kwang Y. Lee. Optimal disturbance rejection for pi controller with constraints on relative delay margin. *ISA Transactions*, 2016.

[Smi52a] J. M. Smith. The importance of the nervous system in the evolution of animal flight. *Evolution*, 6(1):127–129, 1952.

[Smi52b] J. M. Smith. The importance of the nervous sytem in the evolution of aimal flight. *Evolution*, 6:127–129, 1952.

[Smi57a] O. J. M. Smith. Closed control of loops with dead time. *Chemical Engineering Progress*, 53:217–219, 1957.

[Smi57b] O. J. M. Smith. Posicast control of damped oscillatory systems. *Proc. IRE*, 45:1249–1255, 1957.

[Son98] E. P. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Springer, New York, 2nd edition, 1998.

[SP05] S. Skogestad and I Postlethwaite. *Multivariable Feedback Control*. Wiley, Hoboken, NJ, 2nd edition, 2005.

[SS02] E. B. Saff and A. D. Snider. *Fundamentals of Complex Analysis with Applications to Engineering, Science and Mathematics*. Prentice Hall, Englewood Cliffs, NJ, 2002.

[SSS+04] Jörg Stelling, Uwe Sauer, Zoltan Szallasi, Francis J Doyle III, and John Doyle. Robustness of cellular functions. *Cell*, 118(6):675–685, 2004.

[Sta68] L. Stark. *Neurological Control Systems—Studies in Bioengineering*. Plenum Press, New York, 1968.

[Ste80] G. Stein. Adaptive flight control: A pragmatic view. In K. S. Narendra and R. V. Monopoli, editors, *Applications of Adaptive Control*. Academic Press, 1980.

[Ste02] J. Stewart. *Calculus: Early Transcendentals*. Brooks Cole, Pacific Grove, CA, 2002.

[Ste03] G. Stein. Respect the unstable. *Control Systems Magazine*, 23(4):12–25, 2003.

[Str88] G. Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, San Diego, 3rd edition, 1988.

[Str94] S. H. Strogatz. *Nonlinear Dynamics and Chaos, with Applications to Physics, Biology, Chemistry, and Engineering*. Addison-Wesley, Reading, MA, 1994.

[SV89] M. W. Spong and M. Vidyasagar. *Dynamics and Control of Robot Manipulators*. John Wiley, 1989.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Tan96] A. S. Tannenbaum. *Computer Networks*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 1996.

[Teo37] T. Teorell. Kinetics of distribution of substances administered to the body, I and II. *Archives Internationales de Pharmacodynamie et de Therapie*, 57:205–240, 1937.

[Tha89] G. T. Thaler. *Automatic Control Systems*. West Publishing, St. Paul, MN, 1989.

[THÅ00] Tore T. Hägglund and K. J. Åström. Supervision of adaptive control algorithms. *Automatica*, 36(8):1171–1180, 2000.

[Til01] M. Tiller. *Introduction to Physical Modeling with Modelica*. Springer, Berlin, 2001.

[Tru55] J. G. Truxal. *Automatic Feedback Control System Synthesis*. McGraw-Hill, New York, 1955.

[TS90] D. Tipper and M. K. Sundareshan. Numerical methods for modeling computer networks under nonstationary conditions. *IEEE Journal of Selected Areas in Communications*, 8(9):1682–1695, 1990.

[Tsi54] H. S. Tsien. *Engineering Cybernetics*. McGraw-Hill, New York, 1954.

[Tus52] A. Tustin. Feedback. *Scientific American*, 48–54, 1952.

[Vin01] G. Vinnicombe. *Uncertainty and Feedback: $\mathscr{H}_\infty$ Loop-Shaping and the ν-Gap Metric*. Imperial College Press, London, 2001.

[Wad97] H.L. Wade. Inverted decoupling: a neglected technique. *ISA Transactions*, 36(1):3–10, 1997.

[Whi99] F. J. W. Whipple. The stability of the motion of a bicycle. *Quarterly Journal of Pure and Applied Mathematics*, 30:312–348, 1899.

[Wid41] D. V. Widder. *Laplace Transforms*. Princeton University Press, Princeton, NJ, 1941.

[Wie48] N. Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. Wiley, 1948.

[Wig90] S. Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer-Verlag, Berlin, 1990.

[Wil99] H. R. Wilson. *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience*. Oxford University Press, Oxford, UK, 1999.

[Wil04] D. G. Wilson. *Bicycling Science*. MIT Press, Cambridge, MA, 3rd edition, 2004. With contributions by Jim Papadopoulos.

[Wis07] K. A. Wise. Guidance and control for military systems: Future challenges. In *AIAA Conference on Guidance, Navigation, and Control*, 2007. AIAA Paper 2007-6867.

[WT24] E. P. M. Widmark and J. Tandberg. Über die Bedingungen für die Akkumulation indifferenter Narkotika. *Biochemische Zeitung*, 148:358–389, 1924.

[YH91] S. Yamamoto and I. Hashimoto. Present status and future needs: The view from Japanese industry. In Y. Arkun and W. H. Ray, editors, *Chemical Process Control—CPC IV*, 1991.

[YHSD00a] T M Yi, Y Huang, M I Simon, and J Doyle. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proceedings of the National Academy of Sciences*, 97(9):4649–4653, 2000.

[YHSD00b] T.-M. Yi, Y. Huang, M. I. Simon, and J. Doyle. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *PNAS*, 97:4649–4653, 2000.

[Zam81] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximative inverse. *IEEE Transactions on Automatic Control*, AC-26(2):301–320, 1981.

[ZD63] L. A. Zadeh and C. A. Desoer. *Linear System Theory: the State Space Approach*. McGraw-Hill, New York, 1963.

[ZDG96] J. C. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, Englewood Cliffs, NJ, 1996.

[ZN42] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Transactions of the ASME*, 64:759–768, 1942.

Index

$\mathscr{E}$, 9-5
+, 1-6, 1-13, 1-25, 1-26, 4-13, 4-18, 4-25, 4-29, 14-13, 14-31, 14-38

acausal modeling, 3-7
access control, *see* admission control
acknowledgment (ack) packet, 4-13–4-15
activator, 1-12, 3-39, 5-36
active filter, *see also* operational amplifier, 6-24
actuators, 1-5, 3-5, 3-30, 4-1, 4-17, 7-12, 8-25, 9-37, 10-17, 11-22, 12-15, 12-27–12-29, 12-32
  effect on zeros, 10-17, 12-29
  in computing systems, 4-11
  saturation, 3-9, 8-27, 11-9, 11-15–11-17, 11-23, 12-15
A/D converters, *see* analog-to-digital converters
Adapt, 14-24
adaptation, 11-5, 14-15
adaptive control, 13-28, 14-31
additive uncertainty, 13-3, 13-7, 13-10, 13-31
adjacency matrix, 3-36
admission control, 3-33, 3-43, 4-14, 4-15, 10-8
aerospace systems, *see also* vectored thrust aircraft; X-29 aircraft, 1-7–1-8, 1-14, 12-33
AFM, *see* atomic force microscope
air-fuel ratio control, 1-22
aircraft, *see* flight control
alcohol, metabolism of, 4-30
algebraic loops, 3-10–3-11, 8-11
aliasing, 8-27
all-pass transfer function, 12-25
alternating current (AC), 6-25
amplifier, *see* operational amplifier
amplitude ratio, *see* gain
analog computing, 3-10, 4-7, 11-21
analog implementation, controllers, 4-10, 9-35, 11-21–11-22
analog-to-digital converters, 1-5, 4-18, 8-25, 8-26, 11-22
angle, 2-4
anti-windup compensation, 1-19
anticipation, in controllers, *see also* derivative action, 1-19, 11-5
antiresonance, 6-26
anti-windup compensation, 11-15–11-17, 11-22, 11-24, 11-26
Apache web server, *see also* web server control, 4-12
apparent volume of distribution, 4-22, 4-30
Arbib, M. A., 7-1
architecture, 14-1, 14-34
argument, of a complex number, 9-24
arrival rate (queuing systems), 3-33
assume-guarantee, 14-11
asymptotes, in Bode plot, 9-26, 9-27
asymptotic stability, 3-22, 5-8, 5-10, 5-12, 5-13, 5-18, 5-20, 5-23–5-25, 5-27, 6-10
  discrete-time systems, 6-35
atmospheric dynamics, *see* environmental science
atomic force microscopes, 1-3, 3-30, 4-17–4-20
  contact mode, 4-17, 6-26, 7-35
  horizontal positioning, 10-15, 13-20
  system identification, 9-31
  tapping mode, 4-17, 10-24, 11-7, 11-13, 12-18
  with preloading, 4-29
attractor (equilibrium point), 5-10
automatic reset, in PID control, 11-4
automatic tuning, 11-15, 13-28
automation, 3-5
automotive control systems, *see also* cruise control; vehicle steering, 1-17, 3-30, 4-5
autonomous differential equation, *see also* time-invariant systems, 3-3
autonomous vehicles, 14-30–14-33
autonomy, 14-24
autopilot, 1-15, 1-16
AUTOSAR, 14-34

balance systems, *see also* cart-pendulum system; inverted pendulum, 3-13–3-14, 3-29, 7-4, 7-23, 9-14, 12-29
band-pass filter, 6-24, 6-25, 9-29

bandwidth, 2-14, 6-25, 7-20, 12-12, 12-27
behavioral modeling, 3-7
Bell Labs, 1-14, 10-23
Bennett, S., 1-27, 10-24, 11-24
bicycle dynamics, 4-5–4-7, 4-27, 5-29, 8-27
  Whipple model, 4-7
bicycle model, 3-31
bicycle model, for vehicle steering, 3-30–3-32
bicycledynamics
  Whipple model, 7-35
bifurcations, *see also* root locus plots, 5-28–5-30, 5-36
biological circuits, 1-12, 3-24, 3-38–3-39, 5-36, 6-36, 9-29
  genetic switch, 3-43, 5-21
  repressilator, 3-39
biological systems, *see also* biological circuits; drug administration; neural systems; population dynamics, 1-1–1-3, 1-8, 1-12, 1-27, 3-38–3-40, 5-33, 11-1, 11-5
bistability, 1-3, 5-23
Black, 1-7
Black, H. S., 1-14, 4-7, 4-9, 6-1, 10-1, 10-24, 13-1
block diagonal form, 5-12, 5-36, 6-19
block diagonal systems, 5-12, 5-36, 6-9, 6-15, 6-19, 8-12
block diagram algebra, 9-18, 9-21, 13-10
block diagrams, 1-1, 3-10, 3-25–3-26, 9-8, 9-18–9-23
  control system, 1-5, 9-1, 9-20, 12-1
  Kalman decomposition, 8-16
  observable canonical form, 8-5
  observer, 8-2, 8-10
  observer-based control system, 8-14
  PID controllers, 11-1, 11-4, 11-22
  reachable canonical form, 7-7
  two degree-of-freedom controller, 8-22, 12-2, 13-12
  Youla parameterization, 13-11
Bode, 1-7
Bode, H., 9-1, 10-23, 10-24, 12-38, 13-28
Bode plots, 9-24–9-30, 10-16
  asymptotic approximation, 9-26, 9-27, 9-36
  low-, band-, high-pass filters, 9-29
  nonminimum phase systems, 10-17
  of rational function, 9-24
  sketching, 9-27
Bode's ideal loop transfer function, 13-10, 13-29
Bode's integral formula, 12-29–12-31, 12-33–12-34
Bode's relations, 10-16, 12-17
Brahe, T., 3-2
breakpoint, 9-26, 10-6
Bristol's RGA, 14-22
Brockett, R. W., xii, 6-33
Bryson, A. E., 7-35
bump test, 11-11
bumpless transfer, 13-28
bumptest, 11-12
Bush, V., 11-24
business systems, 1-4

calibration, versus feedback, 1-8, 7-14, 7-30, 7-32
Cannon, R. H., 3-40, 6-1
capacitor, transfer function for, 9-7
car, *see* automotive control systems; cruise control; vehicle steering
carrying capacity, in population models, 4-26
cart-pendulum system, *see also* balance systems, 2-29, 3-13, 7-6
cascade control, 14-15–14-16, 14-21
  applications, 14-16
  windup, 14-16
causal reasoning, 1-1, 4-6
Cayley-Hamilton theorem, 7-4, 7-34, 8-3
center (equilibrium point), 5-10
centrifugal governor, 1-2, 1-3, 1-13
certainty equivalence principle, 14-25
chain of integrators (normal form), 3-41, 7-7
characteristic equation, 2-2, 2-5
characteristic polynomial, 2-1, 5-11, 7-34, 9-14
  for closed loop transfer function, 10-2
  observable canonical form, 8-5
  output feedback controller, 8-12, 8-13
  reachable canonical form, 7-7, 7-9, 7-13, 7-33
chemical systems, *see also* process control; compartment models, 1-8, 11-1
chordal distance, 13-5
circuits, *see* biological circuits; electrical circuits
class $\mathscr{E}$, 9-5
classical control, xi, 13-28
closed loop, 1-1, 1-2, 1-5, 6-32, 7-10, 7-17, 10-1, 10-2, 10-20, 12-1
  versus open loop, 1-2, 10-3, 10-21, 12-1
closed loop behavior, 10-1
Command signal following, 2-12
command signals, *see also* reference signal; setpoint, 1-4, 1-6, 8-22, 11-1
compartment models, 4-21–4-25, 5-13, 6-21, 7-20, 8-3, 8-8, 8-28
  exercises, 6-34
compensator, *see* control law
complementary filtering, 14-18
complementary sensitivity function, 12-3, 12-15, 12-31, 13-5, 13-8, 13-11,

13-14, 13-19, 13-23, 13-29
complexity, of control systems, 1-8, 1-17, 11-6
computed torque, 6-33
computer implementation, controllers, 8-25–8-27, 11-22–11-24
computer science, relationship to control, 1-6
computer systems, control of, *see also* queuing systems, 1-9–1-10, 1-27, 3-17, 3-35, 3-36, 4-11–4-17, 6-27
conditional integration, 11-26
conditional stability, 10-9
congestion control, *see also* queuing systems, 1-9, 4-13–4-16, 5-10, 10-7, 10-25, 11-25
  router dynamics, 4-29
consensus, 3-36
control
  definition of, 1-4–1-6
  early examples, 1-2, 1-6–1-8, 1-14, 1-17, 1-27, 11-4
  fundamental limitations, 10-17, 12-25–12-34, 12-38, 13-18, 13-20, 13-27–13-28
  history of, 1-27, 11-24
  modeling for, 1-6, 3-5–3-6, 3-40, 13-1
  successes of, 1-8, 1-26
  system, 1-5, 7-9, 8-14, 8-21, 8-25, 9-1, 12-2, 12-4, 13-12
  using estimated state, 8-11–8-14, 13-24
Control architecture
  two degrees of freedom, 2-14
control architecture
  two degrees of freedom, 2-14
control error, 1-18, 9-20, 11-2
control law, 1-5, 1-18, 6-32, 7-10, 7-13, 9-20
control Lyapunov function, 5-31
control matrix, 3-11, 3-15
control protcol, 14-9
control signal, 3-6, 6-27, 11-1
controllability, *see also* reachability, 7-32
controlled differential equation, 3-3, 3-12
controller architecture, 8-21–8-22
convolution equation, 6-15–6-17, 6-19, 6-20, 7-4, 9-12
  discrete-time, 6-35
coordinate transformations, 5-12, 6-17–6-19, 7-7, 8-27, 9-12–9-13
  to Jordan form, 6-9
  to observable canonical form, 8-6
  to reachable canonical form, 7-8, 7-9
Coriolis forces, 3-13, 6-33
corner frequency, 9-26
correct-by-construction, 14-13
correlation matrix, 8-17, 8-18
cost function, 7-25
coupled spring-mass system, 6-12, 6-14, 6-18
covariance matrix, 8-17
critical gain, 11-11, 11-12, 11-14
critical period, 11-12, 11-14
critical point, 10-2, 10-5, 10-7, 10-13, 10-22, 10-23, 11-11, 13-6, 13-7, 13-26
critically damped oscillator, 7-18
crossover frequency, *see* gain crossover frequency; phase crossover frequency
crossover frequency inequality, *see* gain crossover frequency inequality
cruise control, 1-13–1-14, 1-21, 4-1–4-5
  control design, 7-31, 11-9, 11-20
  electric car, 14-22
  feedback linearization, 6-32
  integrator windup, 11-15, 11-16
  linearization, 6-28
  pole/zero cancellation, 9-17
  robustness, 1-14, 13-1, 13-2, 13-8
Curtiss seaplane, 1-15, 1-16
cybernetics, *see also* robotics, 1-8
cyberphysical system, 3-8

D/A converters, *see* digital-to-analog converters
damped frequency, 7-19
damping, 3-2, 3-13, 3-21, 5-2, 9-37, 9-38
damping ratio, 2-11, 7-18, 7-19, 7-22, 11-8
DARPA Grand Challenge, 14-30, 14-31
DC gain, *see also* zero frequency gain, 6-25
DCS distributed control system, 14-35
dead zone, 1-18
decision making, higher levels of, 14-30
decoupling
  drawbacks, 11-16
deep learning, 14-30
delay, *see* time delay
delay compensation, 10-25, 13-30
delay margin, 10-15
delta function, *see* impulse function
derivative action, 1-19, 1-20, 11-1, 11-4–11-6, 11-22, 12-21
  filtering, 11-5, 11-20, 11-23, 11-24
  setpoint weighting, 11-20, 11-24
  time constant, 11-2
  versus lead compensator, 12-21
derivative gain, 11-2
derivative time constant, 11-5
describing functions, 10-21–10-23
design of dynamics, 1-14–1-15, 5-15, 5-30–5-32, 6-1, 7-1, 7-11, 7-17

diabetes, *see* insulin-glucose dynamics
diagonal systems, 5-12, 6-8
  Kalman decomposition for, 8-14
  transforming to, 5-12, 5-36, 6-8
Dickmanns, E., 14-30
difference equations, 3-11, 3-15–3-17, 3-21, 3-41, 6-27, 8-26, 11-23
differential algebraic equations, *see also* algebraic loops, 3-7
differential equations, 3-2, 3-11–3-15, 5-1–5-4
  controlled, 3-3, 6-3
  equilibrium points, 5-6–5-7
  existence and uniqueness of solutions, 5-2–5-4
  first-order, 3-6, 11-7
  isolated solution, 5-7
  periodic solutions, 5-7–5-8, 5-16
  qualitative analysis, 5-4–5-8
  second-order, 5-5, 7-18, 11-7
  solutions, 5-1, 5-2, 6-3, 6-6, 6-15, 9-34
  stability, *see* stability
  transfer functions for, 9-10
differential flatness, 8-25
differentially flat, 8-25, 14-8
digital control systems, *see* computer implementation, controllers
digital-to-analog converters, 1-5, 4-18, 8-25, 8-26, 11-22
dimension-free variables, 3-28, 3-40
direct connection, 3-10
direct term, 3-10, 3-11, 3-15, 6-17, 8-11
discrete control, 3-35
discrete transition system, 14-9
discrete-time systems, 3-15, 3-41, 5-35, 6-27, 6-35, 11-22
  Kalman filter for, 8-17
  linear quadratic regulator for, 7-27
disk drives, 3-44
distributed control system DCS, 14-35
disturbance attenuation, 1-5, 7-10, 12-13–12-15, 13-12–13-13
  design of controllers for, 12-6, 12-17, 12-31, 12-40, 13-24
  fundamental limits, 12-30
  in biological systems, 9-30, 11-5
  integral gain as a measure of, 11-4, 12-15, 13-13
  relationship to sensitivity function, 12-14, 12-29, 12-39, 13-13
disturbance observer, 14-19
disturbance weighting, 13-27
disturbances, 1-5, 3-3, 3-6, 9-17, 9-20, 12-1, 12-4
  generalized, 13-25
  random, 8-17
Dodson, B., 1-1
dominant eigenvalues (poles), 7-22, 11-8, 11-10
dominant pairs, 7-34
double integrator, 2-5, 6-7, 7-2, 9-10, 10-25
Doyle, J. C., xii, 12-38, 13-28
drawbacks of feedforward, 2-24
drug administration, *see also* compartment models, 4-21–4-25, 4-30, 6-21, 7-20
dual control, 14-27
duality, 8-7, 8-11
Dubins car, 3-31
dynamic compensator, 7-31
dynamic inversion, 6-33
dynamical systems, 1-1, 3-1, 5-1, 5-4, 5-32
  linear, 5-11, 6-1
  observer as a, 8-1
  state of, 7-9
  stochastic, 8-17
  uncertainty in, 13-1–13-3
  zz, *see also* differential equations
dynamics matrix, 3-11, 3-15, 5-11, 6-12
Dyson, F., 3-1

e-commerce, 1-10
e-mail server, control of, 3-17, 6-27
economic systems, 1-4, 1-10–1-11, 3-41
ecosystems, *see also* predator-prey system, 1-12–1-13, 4-25, 7-15
eigenvalue assignment, 7-11, 7-13–7-17, 7-23, 8-12, 11-8, 11-24
  by output feedback, 8-13
  for observer design, 8-8
eigenvalues, 5-11, 5-20, 5-29, 6-12, 9-5
  and Jordan form, 6-9–6-11, 6-35
  distinct, 5-35, 5-36, 6-8, 6-14, 8-14
  dominant, 7-22
  effect on dynamic behavior, 7-17–7-19, 7-21, 7-22, 9-5
  for discrete-time systems, 6-35
  invariance under coordinate transformation, 5-12
  relationship to modes, 6-12–6-15
  relationship to poles, 9-13
  relationship to stability, 5-23, 6-10, 6-11
eigenvectors, 5-12, 5-36, 6-12
  relationship to mode shape, 6-13
electric car, 14-22
electric power, *see* power systems (electric)
electrical circuits, *see also* operational amplifier, 3-7, 3-24, 4-10, 6-1, 9-6
electrical engineering, 1-6–1-7, 3-4–3-5, 6-25, 10-9
elephant, modeling of an, 3-1
Elowitz, M. B., 3-39
encirclement, *see also* Nyquist criterion, 10-5
environmental science, 1-3, 1-8
equation-based modeling, 3-7

equilibrium points, 4-26, 5-6, 5-11, 6-2, 6-29, 7-2
bifurcations of, 5-28
discrete time, 3-41
for closed loop system, 7-11, 7-30
for planar systems, 5-10
region of attraction, 5-26–5-28, 5-34
stability, 5-8
equipment protection, 14-17
error feedback, 2-14, 11-1, 11-2, 11-20, 12-3
estimators, *see* oserversI-1
Euler integration, 3-21, 3-22
exponential functions
simplified notation, 9-5
exponential growth, 2-19
boom-bust, 2-19
stampede, 2-19
exponential input, 2-2, 9-3
exponential signal, 9-4
exponential signals, 9-2–9-9, 9-13, 9-24
exponential signals $\mathscr{E}$, 9-5
extended Kalman filter, 8-23
extremal control, 14-15
extremum seeking, 14-18

FACE, 14-35
FACE Future Airborn Capability, 14-33
Falb, P. L., 7-1
Feedback, 2-1
feedback, 1-1–1-4
as technology enabler, 1-3
business, 1-4
combining with feedforward, 2-24
drawbacks of, 1-3, 1-17, 11-20, 13-6, 13-13
economy, 1-4
generation of discrete behavior, 2-23
in biological systems, *see also* biological circuits, 1-1–1-3, 1-12, 1-27, 11-5
in engineered systems, *see* control
in financial systems, 1-4
in nature, 1-3, 1-11–1-13, 4-25
positive, *see* positive feedback
properties, 1-3, 1-6, 1-13–1-17, 12-1, 12-8, 13-1
robustness through, 1-13
versus feedforward, 1-4, 11-4, 12-6
feedback amplifier, 1-7
feedback and feedforward, 2-23
feedback connection, 2-7, 9-19, 10-21
feedback controller, 9-20, 12-1
feedback linearization, 6-32–6-33
feedback loop, 1-5, 10-1, 12-1, 13-12
feedback uncertainty, 13-3, 13-10
feedback:positive, 2-18
feedforward, 1-4, 8-22, 8-23, 9-20, 12-1, 12-5
antiwindup, 11-16
attenuation of measured disturbances, 2-26
business, 1-4
combining with feedback, 2-24
design, 12-5
drawbacks, 2-24
economy, 1-4
pid, 11-16
properties, 12-8
sensitivity to process variations, 12-7
system inversion, 2-23
Fermi, E., 3-1
filters
active, 6-24
for disturbance weighting, 13-27
for measurement signals, 1-17, 8-26, 13-13
zz, *see also* band-pass filters; high-filters; low-pass filters
financial systems, *see* economic systems
finite escape time, 5-3
finite state machine, 1-21, 3-8, 4-5, 4-12
first-order systems, 6-4, 6-35, 9-10, 9-25, 9-27
fisheries management, 4-30
flatness, *see* differential flatness
flight control, 1-8, 1-15, 3-31, 6-33, 11-16
X-29 aircraft, 12-31
zz, *see also* vectored thrust aircraft
flow, of a vector field, 3-3, 5-5
flow in a tank, 5-33
flow model (queuing systems), 3-33, 10-25, 11-25
flyball governor, *see* centrifugal governor
flying home mode, 14-14
force feedback, 1-8
forced response, 6-3, 9-3
forced solution, 6-3
Forrester, J. W., 1-11
FOTD model, 11-12
Fourier, J. B. J., 3-40, 9-33
frequency domain, 9-1–9-3, 10-1, 10-19, 12-1
frequency response, 2-4, 3-5, 3-23, 3-24, 6-22–6-27, 9-2, 10-24, 11-12, 12-12
relationship to Bode plot, 9-24
relationship to Nyquist plot, 10-4, 10-5
second-order systems, 7-20, 9-29
system identification using, 9-31
fully actuated systems, 9-14
fundamental limits, *see* control: fundamental limitations
Furuta pendulum, 5-36
Future Airborn Capability Environment (FACE), 14-33

gain, 1-19, 2-4, 3-23, 4-9, 6-23, 6-24, 7-21, 9-3, 9-6, 9-13, 9-24, 10-12, 10-19–10-22, 13-1
H_∞, 10-20, 13-26
observer, *see* observer gain
of a system, 10-19

reference, 7-30
state feedback, 7-11, 7-15, 7-30, 7-32
steady-state, 9-13
zero frequency, *see* zero frequency gain
zz, *see also* integral gain
gain crossover frequency, 10-13, 12-12, 12-26, 13-19
gain crossover frequency inequality, 12-26, 12-28
gain curve (Bode plot), 9-24–9-28, 10-16, 12-17
gain margin, 10-12–10-14
from Bode plot, 10-13
reasonable values, 10-14
gain scheduling, 8-22, 8-23, 13-28, 14-15
gain-bandwidth product, 4-10, 9-7, 13-15
Gang of Four, 12-3, 12-39, 13-12
Gang of Seven, 12-3, 12-12
gene regulation, 1-12, 3-38, 6-36, 9-29
genetic switch, 3-43, 5-21
global behavior, 5-10, 5-27–5-30
Glover, K., 12-38, 13-28
glucose regulation, *see* insulin-glucose dynamics
Golomb, S., 4-1
governor, *see* centrifugal governor

H_∞ control, 13-25–13-28, 13-30
haptics, 2-30
Harrier AV-8B aircraft, 3-32
heat propagation, 9-9
Heaviside, O., 6-34
Heaviside step function, 6-20, 6-34
Hellerstein, J. L., 1-27, 4-17
Hewlett Packard, 2-19
Hewlett's oscillator, 2-19
hidden technology, 14-33
high-frequency roll-off, 11-5, 12-17, 13-13, 13-21
high-pass filter, 9-29
Hill function, 3-38
Hoagland, M. B., 1-1
Hodgkin-Huxley equations, 3-39
homeostasis, 1-3, 3-38
homogeneous equation, 2-2
homogeneous solution, 6-3, 6-6
Horowitz, I. M., 8-27, 12-38, 13-23, 13-28
human-machine interface, 1-20, 4-1, 4-5
hybrid system, 3-8, 3-20, 14-9
hysteresis, 1-18, 10-23

identification, *see* system identification
IMC, 14-19
impedance, 9-7, 11-21
impedance control, 2-30
implementation, controllers, *see* analog implementation; computer implementation
impulse function, 6-16, 6-34, 7-4
impulse response, 6-5, 6-16, 6-17, 9-12
inductor, transfer function for, 9-7
inertia matrix, 3-13, 6-33
Inferential control, 14-19
infinity norm, 10-20, 13-26
information systems, *see also* congestion control; web server control, 1-9, 3-33–3-37
initial condition, 5-2, 5-5, 5-8, 6-2, 6-6, 6-7, 6-14, 8-17
initial condition response, 6-3, 6-6–6-9, 6-12, 6-14, 6-17, 9-3
initial value problem, 5-2
inner loop control, 12-35, 12-37
input sensitivity function, *see* load sensitivity function
input/output models, *see also* frequency response; steady-state response; step response, 1-6, 3-4, 3-5, 6-2, 6-15–6-28, 9-1, 10-19
and transfer functions, 9-12
and uncertainty, 3-10, 13-3
from experiments, 9-31
relationship to state space models, 3-6, 5-1, 6-16
steady-state response, 6-19
input/output stable, 10-20
inputs, 3-3, 3-6
insect flight control, 3-25–3-26
instrumentation, 1-8, 4-7
insulin-glucose dynamics, 1-2, 4-24–4-25
integral action, 1-19, 1-20, 1-28, 2-20–2-21, 7-30–7-33, 11-1, 11-3–11-5, 11-7, 12-15
by positive feedback, 2-21
for bias compensation, 8-28
setpoint weighting, 11-20, 11-24
time constant, 11-2
integral gain, 1-19, 11-2, 11-4, 11-7
integrator, *see also* double integrator, 3-25, 6-9, 7-30, 7-31, 8-5, 9-10, 9-25, 10-16, 11-17
integrator windup, 1-19, 8-27, 11-15, 11-17, 11-26
conditional integration, 11-26
intelligent machines, *see* robotics
internal model control, 14-19
internal model principle, 8-13, 8-23
internal stability, 12-3
Internet, *see also* congestion control, 1-10, 4-11, 4-13, 4-16, 4-29
Internet of Things (IoT), 14-36
Internet Protocol (IP), 4-13
invariant set, 5-25, 5-28
inverse, 2-23
inverse model, 6-32, 12-6
inverse response, 10-18, 10-26, 12-9
inverse system approximate, 12-10
inverted pendulum, *see also* balance systems, 3-14–3-15, 4-5, 5-6, 5-14, 5-25, 5-27, 5-35, 5-36,

10-10, 12-32

Jacobian linearization, 6-29–6-31
Janert, P. K., 1-27
Jordan block, 6-9
Jordan form, 6-9–6-12, 6-35, 7-22

Kalman, R. E., 7-1, 7-32, 8-1, 8-15, 8-27
Kalman decomposition, 8-14–8-17, 9-17, 9-34, 9-36
Kalman filter, 8-17–8-21, 8-27, 13-24, 13-25
  extended, 8-23
Kalman-Bucy filter, 8-19
Kelly, F. P., 4-16
Kepler, J., 3-2
Keynesian economic model, 3-41, 6-36
Krasovski-Lasalle principle, 5-24–5-25

LabVIEW, 5-29, 6-34
ladder diagrams, LD, 14-34
lag, *see* phase lag
lag compensation, 12-17–12-19
Laplace transform
  computing transfer function with, 9-11
Laplace transforms, xi, 9-10–9-12
Laplacian matrix, 3-37
Lasalle's invariance principle, *see* Krasovski-Lasalle principle
lead, *see* phase lead
lead compensation, 12-18–12-21, 12-36, 12-41
learn, 14-24
limit cycle, 4-27, 5-7, 5-16, 5-17, 5-29, 10-22
linear quadratic control, 7-25–7-29, 8-18, 8-27, 13-24–13-25
linear system, 2-1
linear systems, 3-4, 3-11, 4-10, 5-11, 6-1–6-34, 8-14, 9-4, 9-33, 10-20
linear temporal logic, 14-10
linear time-invariant systems, 3-4, 3-11, 6-4
linearity, 6-3, 9-24
linearization, 5-15, 5-23, 6-2, 6-28–6-33, 8-23, 13-1
Lipschitz continuity, 5-4
load disturbances, *see also* disturbances, 12-1, 13-13
load sensitivity function, 12-3
local behavior, 5-9, 5-15, 5-24, 5-27, 6-29
locally asymptotically stable, 5-9
logic and sequencing, 14-34
logistic growth model, 4-25, 4-26, 4-30
loop analysis, 10-1, 12-1
loop gain, 10-12
loop shaping, 10-4, 12-16–12-21, 12-38, 13-23
  design rules, 12-18
  fundamental limitations, 12-25–12-34
  zz, *see also* Bode's loop transfer function
loop transfer function, *see also* Bode's loop transfer function, 10-1–10-4, 10-12, 10-20, 12-1, 12-4, 12-16, 12-17, 12-19, 12-31, 12-38
Lotus Notes server, *see* e-mail server
low-order models, 11-7
low-pass filter, 9-29, 11-20
LQ control, *see* linear quadratic control
LTI systems, *see* linear time-invariant systems, 2-2
Lyapunov equation, 5-20, 5-35
Lyapunov functions, 5-17, 5-18, 5-20, 5-21, 5-27, 5-34, 6-35
  design of controllers using, 5-25, 5-31
  existence of, 5-20
Lyapunov stability analysis, 3-23, 5-17–5-26, 5-33
  discrete time, 5-35
magnitude, 2-4
manifold, 5-26
margins, *see* stability margins
materials science, 1-8
Mathematica, 3-21, 5-29, 6-34
MATLAB, 1-27, 3-21, 5-29, 6-34, 7-35
  acker, 7-15, 8-11
  dlqe, 8-18
  dlqr, 7-29
  hinfsyn, 13-26
  jordan, 6-10
  linmod, 6-30
  lqr, 7-26
  place, 7-15, 7-24, 8-11
  trim, 6-30
matrix exponential, 6-6–6-9, 6-13, 6-15, 6-33, 6-34
  coordinate transformations, 6-18
  Jordan form, 6-10
  second-order systems, 6-34
maximum, 14-17
maximum complementary sensitivity, 13-8, 13-19
maximum selector, 1-22
maximum sensitivity, 12-14, 13-6, 13-20
measured signals, 3-6, 3-11, 5-1, 8-1, 8-14, 8-26, 12-2, 12-4, 13-25
measurement noise, 1-5, 1-17, 8-1, 8-3, 8-17, 8-19, 9-20, 11-20, 12-1–12-3, 12-17, 13-13
  response to, 12-15–12-16, 13-13
mechanical systems, 3-6, 3-13, 3-22, 3-30, 3-40, 6-32
mechanics, 3-2–3-3, 3-5, 5-32, 6-1
median selectors, 14-17
mid-range control, 14-15, 14-16
minimal model (insulin-glucose), *see also* insulin-glucose dynamics, 4-24, 4-25
minimum, 14-17
minimum phase, 10-17, 10-24, 12-25
minimum selector, 1-22

mixed integer solvers, 14-11
mixed logical dynamical, 14-11
model checking, 14-13
model following, 14-15
Modelica, 3-7
modeling, 1-6, 3-1–3-10, 3-40, 4-1
  control perspective, 3-5
  discrete control, 3-35
  discrete-time, 3-15, 6-27–6-28
  frequency domain, 9-1–9-3
  from experiments, 3-27–3-28
  model reduction, 1-6
  normalization and scaling, 3-28
  of uncertainty, 3-9–3-10
  simplified models, use of, 3-6, 11-7, 13-2, 13-8, 13-9
  software for, 3-7, 6-30, 6-33
  state space, 3-11–3-23
  uncertainty, *see* uncertainty
modes, 6-12–6-14, 9-13
  relationship to poles, 9-14
monotone step responses, 11-12
motion control systems, 3-30–3-33, 8-27
motors, electric, 3-44, 7-34, 8-29
multi-input, multi-output systems, *see also* input/output models, 10-20, 12-4, 12-18
multiplicative uncertainty, 13-3, 13-10

nanopositioner (AFM), 10-15, 13-20
natural frequency, 7-19, 11-8
negative definite function, 5-18
negative feedback, 1-3, 1-14, 4-9, 7-10, 10-1, 11-6
Nernst's law, 3-40
networking, *see also* congestion control, 1-9, 3-24, 4-16
neural systems, 1-8, 3-26, 3-39, 11-6
neutral stability, 5-8–5-10
Newton, I., 3-2
Nichols, N. B., 6-33, 11-10, 11-11, 12-38
Nichols chart, 13-24
Nobel Prize, 1-8, 3-40, 4-17
noise, *see* disturbances; measurement noise
noise attenuation, 9-30, 12-15–12-16
noise cancellation, 5-31
noise sensitivity function, 12-3
nonlinear systems, 3-6, 5-1, 5-4, 5-7, 5-15, 5-17, 5-21, 5-27–5-32, 8-2, 8-22, 8-23, 10-20, 10-21
  linear approximation, 5-15, 5-23, 6-29, 6-36, 13-1
  system identification, 3-42
nonminimum phase, *see also* inverse response, 10-16, 10-17, 10-26, 12-25–12-27
nonunique solutions (ODEs), 5-3
normalized coordinates, 3-28–3-30, 3-42, 6-31
norms, 10-19–10-20
Nyquist, 1-7
Nyquist, H., 10-1, 10-23
Nyquist criterion, 10-5, 10-7, 10-9, 10-12, 10-20, 10-21, 11-11
  for robust stability, 13-6, 13-31
Nyquist D contour, 10-4, 10-10
Nyquist plot, 10-4–10-5, 10-12, 10-13, 11-11, 12-14, 13-24

observability, 3-6, 8-1–8-2, 8-14, 8-27
  rank condition, 8-3
  tests for, 8-2–8-3
  unobservable systems, 8-4, 8-14–8-16, 9-36
observability matrix, 8-3, 8-5
observable canonical form, 8-4, 8-5, 8-27
observer gain, 8-7, 8-9–8-11, 8-13, 8-18, 8-19
observers, 8-1, 8-6–8-9, 8-19, 8-23
  block diagram, 8-2, 8-10
  zz, *see also* Kalman filter
ODEs, *see* differential equations
Ohm's law, 3-40, 4-9, 9-6
on-off control, 1-18
open loop, 1-1, 4-9, 7-2, 9-22, 10-1, 11-15, 12-1, 12-13, 13-3
open loop gain, 9-7, 12-12
operational amplifiers, 4-7–4-11, 9-7, 11-21, 13-10
  circuits, 4-28, 6-24, 10-2, 13-14
  dynamic model, 4-10, 9-7
  input/output characteristics, 4-8
  oscillator using, 4-28, 5-35
  static model, 4-8, 9-7
optimal control, 7-25, 8-17, 8-19, 13-25
order, of a model, 3-11, 3-12
ordinary differential equations, *see* differential equations
oscillator dynamics, 4-28, 5-2, 5-3, 6-7, 6-8, 7-18, 9-5, 9-10
  normal form, 3-42
  zz, *see also* nanopositioner (AFM); spring-mass system
outer loop control, 12-35–12-37
output feedback, *see also* control: using estimated state; loop shaping; PID control, 8-11, 8-12, 8-27
output sensitivity function, *see* noise sensitivity function
outputs, *see* measured signals
overdamped oscillator, 7-18
overshoot, 6-21, 7-10, 7-20, 12-12

P control, 2-10
Padé approximation, 10-26, 12-27
paging control (computing), 3-35

pairing problem, 14-21
parallel connection, 2-7, 9-19
parallel systems, 14-22–14-23
parametric stability diagram, 5-28–5-30
parametric uncertainty, 3-9, 13-1
particular solution, *see also* forced response, 2-2, 6-3, 6-22, 9-5
  transfer function, 2-2
passive systems, 10-21, 12-30
passivity theorem, 10-21
patch clamp, 1-8
PD control, 2-29, 11-4, 12-18, 12-19
peak frequency, 6-26, 12-12
pendulum dynamics, *see also* inverted pendulum, 5-19
perfect adaptation, 11-6
perfect control, 14-21
performance, 4-12
performance limitations, 12-25, 12-30, 13-20, 13-27
  due to right half-plane poles and zeros, 10-17
  zz, *see also* control: fundamental limitations
performance specifications, *see also* overshoot; maximum sensitivity; resonant peak; rise time; settling time, 6-21, 7-10, 12-1, 12-12–12-16, 12-18, 13-12
periodic solutions, *see* differential equations; limit cycles
persistant excitation, 14-26
persistence, of a web connection, 4-12, 4-13
Petri net, 3-24
pharmacokinetics, *see also* drug administration, 4-21, 4-25
phase, *see also* minimum phase; nonminimum phase, 2-4, 3-23, 6-23, 6-24, 7-21, 9-3, 9-6, 9-24, 10-22
  minimum vs. nonminimum, 10-16
phase crossover frequency, 10-13
phase curve (Bode plot), 9-24–9-26, 9-28
  relationship to gain curve, 10-16, 12-17
phase lag, 6-23, 6-24, 9-29, 10-17, 12-26, 12-28
phase lead, 6-23, 9-29, 12-21, 12-41
phase margin, 10-13, 10-14, 12-17, 12-19, 12-26, 12-41, 13-29
  from Bode plot, 10-13
  reasonable values, 10-14
phase portrait, 3-3, 5-4–5-6, 5-27
Philbrick, G. A., 4-11
photoreceptors, 11-6
physics, relationship to control, 1-6
PI Control, 2-9
PI control, 1-13, 1-19, 2-10, 2-21, 4-1, 4-4, 11-4, 11-10, 12-18, 12-19
  first-order system, 11-8, 13-18
PID control, 1-18–1-19, 11-1–11-24, 12-21
  block diagram, 11-2, 11-4, 11-17
  computer implementation, 11-22
  ideal form, 11-1, 11-24
  implementation, 11-4, 11-19–11-24
  in biological systems, 11-5
  op amp implementation, 11-21–11-22
  tuning, 11-10–11-15
  zz, *see also* derivative action; integral action
pid control
  feedforward signal, 11-16
pitchfork bifurcation, 5-37
planar dynamical systems, *see also* second-order systems, 5-5, 5-10
PLC Programmable Logic Controller, 14-34
PLC programmable logic controllers, 14-35
pole and zeros, 9-13–9-16
pole excess, 12-22
pole placement, *see also* eigenvalue assignment, 7-11, 13-15, 13-19–13-20
  robust, 13-15
pole zero diagram, 9-14
pole/zero cancellations, 9-16–9-18, 9-36, 12-4, 13-20
poles, 2-5, 9-13, 9-14
  dominant, *see also* dominant eigenvalues (poles), 11-10
  fast stable, 13-18, 13-20
  pure imaginary, 10-5, 10-10
  relationship to eigenvalues, 9-13
  right half-plane, 9-14, 10-10, 10-17, 12-25, 12-27–12-30, 12-41, 13-20
poles and zeros, 9-13
population dynamics, *see also* predator-prey system, 4-25–4-27, 4-30
positive definite function, 5-18, 5-20, 5-24
positive definite matrix, 5-20, 7-25
Positive feedback, 1-3
positive feedback, 1-3, 1-17, 2-18, 2-21, 5-36, 11-4
positive real (transfer function), 12-30
power of a matrix, 6-6
power systems (electric), 3-43, 5-7, 5-34
power systems, electric, 1-6–1-7
predator-prey system, 3-15, 4-26–4-27, 5-28, 7-15
prediction, in controllers, *see also* derivative action, 1-19, 8-23, 11-5, 13-29
prediction time, 11-5
principle of the argument, *see* variation of the argument, principle of
process control, 1-8, 3-24
program synthesis, 14-13
Programmable Logic

Controller (PLC), 14-34
programmable logic controllers, PLC, 14-35
proportional control, *see also* PID control, 1-18, 1-19, 2-9, 11-1
proportional, integral, derivative control, *see* PID control
proportional-derivative (PD) controller, 2-29
Proportional-Integral Control, 2-9
protocol, *see* congestion control; consensus
pulse signal, *see also* impulse function, 6-16, 6-17, 7-22
pupil response, 9-32, 11-5
pure exponential solution, 9-5

Q-value, 3-42, 7-20, 9-27
quantitative feedback theory (QFT), 13-23–13-24
quarter car model, 9-37
queuing systems, 3-33–3-35, 3-43

random process, 3-33, 8-17, 8-18, 8-29
reachability, 3-6, 7-1–7-9, 7-32, 8-14
  rank condition, 7-4
  tests for, 7-3
  unreachable systems, 7-5, 7-34, 8-14–8-16, 9-36
reachability matrix, 7-3, 7-8
reachable canonical form, 3-12, 7-6–7-9, 7-13, 7-14, 7-33
reachable set, 7-1
real-time systems, 1-6
reason, 14-24
receding horizon control, 14-7
reference signal, *see also* command signals; setpoint, 1-18, 7-10, 9-1, 9-20, 11-1, 11-20, 12-3
  effect on observer error, 8-12, 8-16, 8-22
  response to, 12-12, 12-13, 12-39
  tracking, 7-10, 8-21, 8-22, 12-17, 13-14
reference weighting, *see* setpoint weighting
region of attraction, *see* equilibrium points: regions of attraction
regulation problem, 2-8
regulator, *see* control law
relative gain array, 14-21–14-22
relay feedback, 10-23, 11-14
Reno (protocol), *see* Internet; congestion control
repetitive control, 14-15
repressilator, 3-39
repressor, 1-12, 3-39, 3-43, 5-21, 6-36, 9-30
requirements, 14-2
reset logic, 3-8
reset, in PID control, 11-3, 11-4
resonant frequency, 7-20, 10-20
resonant peak, 6-26, 7-20, 12-12, 13-9
resource usage, in computing systems, 3-34, 3-36, 4-11, 4-12
response, *see* input/output models
retina, *see also* pupil response, 11-6
Riccati equation, 7-26, 8-19, 13-26, 13-28
Riemann sphere, 13-5
right half plane zero, 12-9
right half-plane poles and zeros, *see* poles: right half-plane; zeros: right half-plane
rise time, 6-21, 7-10, 7-20, 12-12
robotics, 1-8–1-9, 6-33
robustness, 1-12–1-14, 11-5, 12-12, 13-3, 13-28
  performance, 13-12–13-15, 13-23–13-28
  stability, 13-6–13-12
  using gain and phase margin, 10-14, 12-17
  using maximum sensitivity, 12-14, 12-17, 13-7, 13-29, 13-31
  using pole placement, 13-15–13-23
  via gain and phase margin, 10-14
  zz, *see also* uncertainty
roll-off, *see* high-frequency roll-off
root locus, 12-21
  asymptotes, 12-40
  initial direction, 12-40
  real line segment, 12-40
root locus diagram, 5-29, 5-30
root locus method, 12-21
Routh-Hurwitz criterion, 2-6
Routh-Hurwitz stability criterion, 2-5
rush-hour effect, 3-35, 3-43

saddle (equilibrium point), 5-10
safety, 14-17
sampling, 6-27, 8-25, 8-26, 11-23
saturation function, *see also* actuators: saturation, 3-25, 4-8, 11-23
scaling, *see* normalized coordinates
scanning tunneling microscope, 4-17
schematic diagrams, 3-23, 3-24, 4-7
Schitter, G., 4-20
second-order systems, 3-2, 6-34, 7-18–7-21, 7-35, 9-26, 9-27, 11-10
Segway Personal Transporter, 3-13, 7-4
selector, 1-22
selector control, 14-15, 14-17
  of air-fuel, 1-22
selector, maximum, 1-22
selector,minimum, 1-22
self-activation, 5-36
self-optimization, 14-18
self-repression, 6-36, 9-29
semidefinite function, 5-18
sensitivity
  to modeling errors, 2-24, 2-25

sensitivity crossover frequency, 12-14
sensitivity function, 2-26, 12-3, 12-14, 12-15, 12-17, 12-30, 13-6, 13-14, 13-20
  and disturbance attenuation, 12-14, 12-30, 12-39
sensor matrix, 3-11, 3-15
sensor networks, 3-36
sensors, 1-5, 8-2, 8-25, 10-17, 11-22, 12-1, 12-4, 12-27, 12-28, 13-25
  effect on zeros, 10-17, 12-29
  in computing systems, 4-11
  zz, *see also* measured signals
separation principle, 8-13
series connection, 2-7, 9-19
service rate (queuing systems), 3-33
servo problem, 2-12
setpoint, 11-1
setpoint weighting, 11-20–11-21, 11-24
settling time, 6-21, 6-35, 7-10, 7-20, 12-12
similarity of two systems, 13-3–13-6
simplified notation
  exponential functions, 9-5
simulation, 3-10, 3-20–3-21
SIMULINK, 6-30
simultaneous positioning and mapping SLAM, 14-28
single-input, single-output (SISO) systems, 5-1, 6-2, 6-3, 6-29, 8-4, 10-20
singular values, 10-19, 10-20, 13-30
sink (equilibrium point), 5-10
SLAM, 14-28
SLAM simultaneous positioning and mapping, 14-28
small gain theorem, 10-20–10-21, 13-9
Smith predictor, 13-29, 14-20
software tools for control, x
solution (ODE), *see* differential equations: solutions
source (equilibrium point), 5-10
spectrum analyzer, 9-31
Sperry autopilot, 1-15
spring-mass system, 3-2, 3-20, 3-21, 3-23, 4-18, 5-34
  coupled, 6-14, 6-18
  generalized, 3-13, 4-7
  identification, 3-27
  normalization, 3-29, 3-42
  zz, *see also* oscillator dynamics
Stability, 2-5
  Rout-Hurwitz criterion, 2-5
stability, 1-5, 1-6, 1-14, 1-15, 2-5, 3-21, 5-4, 5-8–5-26
  asymptotic stability, 5-8, 5-13
  conditional, 10-9
  in the sense of Lyapunov, 5-8
  local versus global, 5-9, 5-16, 5-27
  Lyapunov analysis, *see* Lyapunov stability analysis
  neutrally stable, 5-8, 5-10
  of a system, 5-11
  of equilibrium points, 3-22, 5-8, 5-10, 5-17, 5-18, 5-23
  of feedback loop, *see* Nyquist criterion
  of limit cycles, 5-16
  of linear systems, 5-11–5-14, 5-20, 6-10
  of solutions, 5-8, 5-9, 5-17
  of transfer functions, 9-14
  robust, *see* robust stability
  unstable solutions, 5-9
  using eigenvalues, 5-23, 6-10, 6-11
  using linear approximation, 5-14, 5-23, 6-30
  using state feedback, 7-9–7-29
  zz, *see also* bifurcations; equilibrium points
stability diagram, *see* parametric stability diagram
stability margin (quantity), 10-13, 10-14, 12-14, 12-41, 13-7, 13-26
  reasonable values, 10-14
stability margins (concept), 10-12–10-16, 10-25, 12-17
stable, 2-5
stable pole, 9-14
stable zero, 9-14
Stark, L., 9-32
state, of a dynamical system, 3-2, 3-6, 3-11
state estimators, *see* observers
state feedback, *see also* eigenvalue assignment; linear quadratic control, 7-1–7-32, 8-7, 8-12, 8-21–8-23, 8-25–8-27, 13-16, 13-25, 14-15
state space, 3-2, 3-11–3-23, 7-9
state vector, 3-2, 3-11
static gain, 2-4
steady state gain, 2-4
steady state solution, 9-5
steady-state gain, *see* zero frequency gain, 9-13
steady-state response, 1-28, 3-21, 6-19–6-27, 6-36, 7-11, 7-20, 9-2, 9-31, 9-33
steam engines, 1-2, 1-13
steering, *see* vehicle steering
Stein, G., xii, 12-1, 12-31
step input, 3-4, 6-5, 6-20, 9-13
step response, 2-2, 3-4, 3-5, 3-27, 3-28, 6-5, 6-17, 6-20, 6-21, 7-10, 7-19, 7-20, 11-11
stochastic systems, 8-17, 8-19
summing junction, 3-25
superposition, 3-4, 6-3, 6-17, 6-34, 9-2
supervised learning, 14-27
supervisory control, *see* decision making: higher levels of
supervisory controller, 1-22
supply chains, 1-11
supremum (sup), 10-20
switching behavior, 1-3, 3-43, 5-23, 5-24, 13-28
system identification, 3-27, 3-28, 3-42, 9-31
system inversion, 2-23, 12-10

approximate, 12-10
system response, 2-4

tapping mode, *see* atomic force microscope
TCP/IP, *see* Internet; congestion control
temporal logic, 14-10
Teorell, T., 4-21, 4-25
the relative gain array, 14-22
three-term controllers, *see also* PID control, 11-2
thrust vectored aircraft, *see* vectored thrust aircraft
time constant, 2-2
time constant, first-order system, 6-35
time delay, 1-10, 9-9, 9-10, 10-15, 10-17, 11-11, 11-12, 11-22, 12-27, 12-28
  compensation for, 13-29, 13-30
  Padé approximation, 10-26, 12-27
time invariant, 2-1
time plot, 3-3
time-invariant systems, 3-4, 3-11, 5-33, 6-4–6-5
tracking, *see* reference signal: tracking
tracking mode, 11-18
trail (bicycle dynamics), 4-6, 4-7
transcription factors, 3-38
transcriptional regulation, *see* gene regulation
transfer function, 2-3
transfer function:looptracing, 9-21
transfer functions, 9-1–9-33
  common systems, 9-10
  derivation using exponential signals, 9-4
  for control systems, 9-20, 9-36
  for electrical circuits, 9-6
  for time delay, 9-9
  frequency response, 9-2, 9-24
  from experiments, 9-31
  irrational, 9-9
  linear input/output systems, 9-4, 9-10, 9-36
  simplified notation, 9-5
transfer functions: , 9-12
transfer functions: Laplace transforms, 9-11
transfer functions:impulse response, 9-12
transfer functions:state space model, 9-11
transient response, 3-21, 6-20, 6-21, 6-23, 7-2, 7-23
Transmission Control Protocol (TCP), 4-13
transmission zero, 2-5
Tsien, H. S., 1-9
tuning rules, *see* Ziegler-Nichols tuning, 11-26
Tustin, A., 2-1
two degree-of-freedom control, 8-22, 11-2, 12-5, 12-38, 12-39
two degrees of freedom, 2-14, 2-25
two degrees of freedom, 2 DOF, 2-25
two-out-of-three selectors, 14-17

uncertainty, 1-5, 1-13–1-14, 3-6, 3-9–3-10, 7-30, 13-1–13-6
  component or parameter variation, 1-5, 3-9, 13-1
  disturbances and noise, 1-5, 3-6, 7-10, 9-20, 12-1
  unmodeled dynamics, 1-5, 3-9, 13-2, 13-8
  zz, *see also* additive uncertainty; feedback uncertainty; multiplicative uncertainty
uncertainty band, 3-9
uncertainty lemon, 3-9, 4-4, 4-10, 4-20
undamped natural frequency, 2-11
underdamped oscillator, 5-3, 7-19, 7-20
unit step, 6-20
unmodeled dynamics, *see* uncertainty: unmodeled dynamics, 2-11, 2-12
  effect for control, 2-12
unstable pole, *see* poles: right half-plane
unstable pole/zero cancellation, 9-17
unstable solution, for a dynamical system, 5-9, 5-10, 5-13, 6-10, 9-14
unstable zero, *see* zeros: right half-plane
unsupervised learning, 14-27

variation of the argument, principle of, 10-11, 10-24
vector field, 3-3, 5-5
vectored thrust aircraft, 3-32–3-33, 6-11, 7-26, 8-20, 9-36, 12-20, 12-35
vehicle steering, 3-30–3-32, 6-30, 7-11, 8-9, 8-13, 8-23, 9-22, 10-18, 10-25, 12-6, 13-16
  ship dynamics, 3-30
vehicle suspension, *see also* coupled spring-mass system, 9-37
vertical contract, 14-11
vertical takeoff and landing, *see* vectored thrust aircraft
vibration absorber, 9-38
Vinnicombe, G., 12-38, 13-5, 13-28
Vinnicombe metric, 13-3–13-6
voltage clamp, 1-8, 1-9, 3-40

waterbed effect, 12-30, 12-31
Watt governor, *see* centrifugal governor
Watt steam engine, 1-3, 1-13
web server control, 4-11–4-13, 7-27
web site, companion, x
Whipple, F. J. W., 4-7
Wiener, N., 1-9
winding number, 10-11
window size (TCP), 4-14, 4-16, 5-10
windup, *see* integrator windup
  cascade control, 14-16
  selector control, 14-17

Wright, W., 1-15
Wright Flyer, 1-7, 1-15

X-29 aircraft, 12-31

Youla parameterization, 13-10–13-12

zero
  blocking property, 2-5
zero frequency gain, 2-4, 6-25, 7-11, 7-14, 7-20, 9-13, 11-12
zeros, 2-5, 9-13
  Bode plot for, 9-36
  effect of sensors and actuators on, 10-17, 10-18, 12-29
  for a state space system, 9-14
  right half-plane, 9-14, 10-17, 12-25–12-29, 12-31, 12-41, 13-19
  signal-blocking property, 9-13
  slow stable, 13-16, 13-17, 13-19
Ziegler, J. G., 11-10, 11-11, 11-24
Ziegler-Nichols tuning, 11-11–11-14, 11-24
  frequency response, 11-11
  improved method, 11-12
  step response, 11-11