Linear Controller Design: Limits of Performance

Stephen Boyd Craig Barratt

Originally published 1991 by Prentice-Hall.

Contents

Preface

This book is motivated by the following technological developments: high quality integrated sensors and actuators, powerful control processors that can implement complex control algorithms, and powerful computer hardware and software that can be used to design and analyze control systems. We believe that these technological developments have the following ramifications for linear controller design:

- When many high quality sensors and actuators are incorporated into the design of a system, sophisticated control algorithms can outperform the simple control algorithms that have sufficed in the past.

- Current methods of computer-aided control system design underutilize available computing power and need to be rethought.

This book is one small step in the directions suggested by these ramifications.

We have several goals in writing this text:

- To give a clear description of how we might formulate the linear controller design problem, without regard for how we may actually solve it, modeling *fundamental specifications* as opposed to specifications that are artifacts of a particular method used to solve the design problem.

- To show that a wide (but incomplete) class of linear controller design problems can be cast as convex optimization problems.

- To argue that solving the controller design problems in this restricted class is in some sense fundamentally tractable: although it involves more computing than the standard methods that have "analytical" solutions, it involves much less computing than a global parameter search. This provides a partial answer to the question of how to use available computing power to design controllers.

- To emphasize an aspect of linear controller design that has not been emphasized in the past: the determination of *limits of performance*, *i.e.*, specifications that cannot be achieved with a given system and control configuration.

It is *not* our goal to survey recently developed techniques of linear controller design, or to (directly) teach the reader how to design linear controllers; several existing texts do a good job of that. On the other hand, a clear formulation of the linear controller design problem, and an understanding that many of the performance limits of a linear control system can be computed, are useful to the practicing control engineer.

Our intended audience includes the sophisticated industrial control engineer, and researchers and research students in control engineering.

We assume the reader has a basic knowledge of linear systems (Kailath [KAI80], Chen [CHE84], Zadeh and Desoer [ZD63]). Although it is not a prerequisite, the reader will benefit from a prior exposure to linear control systems, from both the "classical" and "modern" or state-space points of view. By *classical control* we refer to topics such as root locus, Bode plots, PI and lead-lag controllers (Ogata [OGA90], Franklin, Powell, Emami [FPE86]). By *state-space control* we mean the theory and use of the linear quadratic regulator (LQR), Kalman filter, and linear quadratic Gaussian (LQG) controller (Anderson and Moore [AM90], Kwakernaak and Sivan [KS72], Bryson and Ho [BH75]).

We have tried to maintain an informal, rather than completely rigorous, approach to the mathematics in this book. For example, in chapter 13 we consider linear functionals on infinite-dimensional spaces, but we do not use the term *dual space*, and we avoid any discussion of their continuity properties. We have given proofs and derivations only when they are simple and instructive. The references we cite contain precise statements, careful derivations, more general formulations, and proofs.

We have adopted this approach because we believe that many of the basic ideas are accessible to those without a strong mathematics background, and those with the background can supply the necessary qualifications, guess various generalizations, or recognize terms that we have not used.

A Notes and References section appears at the end of each chapter. We have not attempted to give a complete bibliography; rather, we have cited a few key references for each topic. We apologize to the many researchers and authors whose relevant work (especially, work in languages other than English) we have not cited. The reader who wishes to compile a more complete set of references can start by computing the transitive closure of ours, *i.e.*, our references along with the references in our references, and so on.

Our first acknowledgment is to Professor C. Desoer, who introduced the idea of the Q-parametrization (along with the advice, "this is good for CAD") to Stephen Boyd in EECS 290B at Berkeley in 1981. We thank the reviewers, Professor C. Desoer, Professor P. Kokotovic, Professor L. Ljung, Dr. M. Workman of IBM, and Dr. R. Kosut of Integrated Systems, Inc. for valuable suggestions. We are very grateful to Professor T. Higgins for extensive comments on the history and literature of control engineering, and a thorough reading of our manuscript. We thank S. Norman, a coauthor of the paper [BBN90], from which this book was developed, and V. Balakrishnan, N. Boyd, X. Li, C. Oakley, D. Pettibone, A. Ranieri, and Q. Yang for numerous comments and suggestions.

During the research for and writing of this book, the authors have been supported by the National Science Foundation (ECS-85-52465), the Office of Naval Research (N00014-86-K-0112), and the Air Force Office of Scientific Research (89-0228).

Stephen Boyd
Craig Barratt

Stanford, California
May, 1990

This book was typeset by the authors using LaTeX. The simulations, numerical computations, and figures were developed in MATLAB and C, under the UNIX operating system. We encourage readers to attempt to reproduce our plots and figures, and would appreciate hearing about any errors.

Chapter 1

Control Engineering and Controller Design

Controller design, the topic of this book, is only a part of the broader task of control engineering. In this chapter we first give a brief overview of control engineering, with the goal of describing the context of controller design. We then give a general discussion of the goals of controller design, and finally an outline of this book.

1.1 Overview of Control Engineering

The goal of control engineering is to improve, or in some cases enable, the performance of a system by the addition of *sensors*, *control processors*, and *actuators*. The sensors measure or sense various signals in the system and operator commands; the control processors process the sensed signals and drive the actuators, which affect the behavior of the system. A schematic diagram of a general *control system* is shown in figure 1.1.

This general diagram can represent a wide variety of control systems. The system to be controlled might be an aircraft, a large electric power generation and distribution system, an industrial process, a head positioner for a computer disk drive, a data network, or an economic system. The signals might be transmitted via analog or digitally encoded electrical signals, mechanical linkages, or pneumatic or hydraulic lines. Similarly the control processor or processors could be mechanical, pneumatic, hydraulic, analog electrical, general-purpose or custom digital computers.

Because the sensor signals can affect the system to be controlled (via the control processor and the actuators), the control system shown in figure 1.1 is called

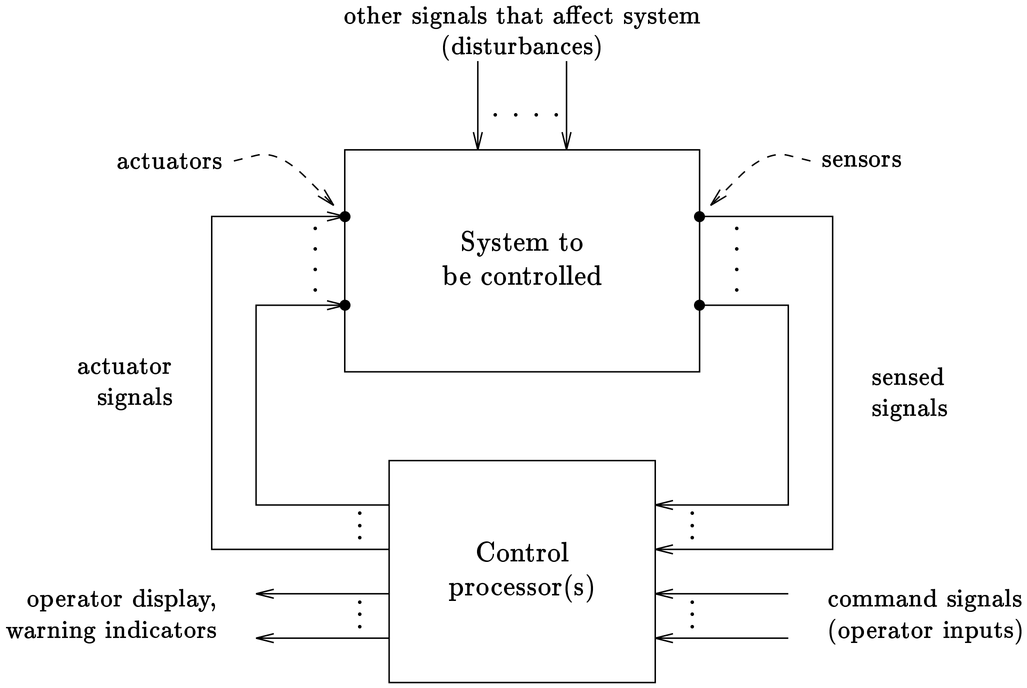


Figure 1.1 A schematic diagram of a general control system.

a *feedback* or *closed-loop* control system, which refers to the signal "loop" that circulates clockwise in this figure. In contrast, a control system that has no sensors, and therefore generates the actuator signals from the command signals alone, is sometimes called an *open-loop* control system. Similarly, a control system that has no actuators, and produces only operator display signals by processing the sensor signals, is sometimes called a *monitoring system.*

In industrial settings, it is often the case that the sensor, actuator, and processor signals are *boolean*, *i.e.* assume only two values. Boolean sensors include mechanical and thermal limit switches, proximity switches, thermostats, and pushbutton switches for operator commands. Actuators that are often configured as boolean devices include heaters, motors, pumps, valves, solenoids, alarms, and indicator lamps. Boolean control processors, referred to as *logic controllers*, include industrial relay systems, general-purpose microprocessors, and commercial *programmable logic controllers.*

In this book, we consider control systems in which the sensor, actuator, and processor signals assume real values, or at least digital representations of real values. Many control systems include both types of signals: the real-valued signals that we will consider, and boolean signals, such as fault or limit alarms and manual override switches, that we will not consider.

In control systems that use digital computers as control processors, the signals are sampled at regular intervals, which may differ for different signals. In some cases these intervals are short enough that the sampled signals are good approximations of the continuous signals, but in many cases the effects of this sampling must be considered in the design of the control system. In this book, we consider control systems in which all signals are continuous functions of time.

In the next few subsections we briefly describe some of the important tasks that make up control engineering.

1.1.1 System Design and Control Configuration

Control configuration is the selection and placement of the actuators and sensors on the system to be controlled, and is an aspect of system design that is very important to the control engineer. Ideally, a control engineer should be involved in the design of the system itself, even before the control configuration. Usually, however, this is not the case: the control engineer is provided with an already designed system and starts with the control configuration. Many aircraft, for example, are designed to operate without a control system; the control system is intended to improve the performance (indeed, such control systems are sometimes called *stability augmentation* systems, emphasizing the secondary role of the control system).

Actuator Selection and Placement

The control engineer must decide the type and placement of the actuators. In an industrial process system, for example, the engineer must decide where to put actuators such as pumps, heaters, and valves. The specific actuator hardware (or at least, its relevant characteristics) must also be chosen. Relevant characteristics include cost, power limit or authority, speed of response, and accuracy of response. One such choice might be between a crude, powerful pump that is slow to respond, and a more accurate but less powerful pump that is faster to respond.

Sensor Selection and Placement

The control engineer must also decide which signals in the system will be measured or sensed, and with what sensor hardware. In an industrial process, for example, the control engineer might decide which temperatures, flow rates, pressures, and concentrations to sense. For a mechanical system, it may be possible to choose *where* a sensor should be placed, *e.g.*, where an accelerometer is to be positioned on an aircraft, or where a strain gauge is placed along a beam. The control engineer may decide the particular type or relevant characteristics of the sensors to be used, including the type of transducer, and the signal conditioning and data acquisition hardware. For example, to measure the angle of a shaft, sensor choices include a potentiometer, a rotary variable differential transformer, or an 8-bit or 12-bit absolute or differential shaft encoder. In many cases, sensors are smaller than actuators, so a change of sensor hardware is a less dramatic revision of the system design than a change of actuator hardware.

There is not yet a well-developed theory of actuator and sensor selection and placement, possibly because it is difficult to precisely formulate the problems, and possibly because the problems are so dependent on available technology. Engineers use experience, simulation, and trial and error to guide actuator and sensor selection and placement.

1.1.2 Modeling

The engineer develops mathematical models of

- the system to be controlled,
- noises or disturbances that may act on the system,
- the commands the operator may issue,
- desirable or required qualities of the final system.

These models might be deterministic (*e.g.*, ordinary differential equations (ODE's), partial differential equations (PDE's), or transfer functions), or stochastic or probabilistic (*e.g.*, power spectral densities).

Models are developed in several ways. *Physical modeling* consists of applying various laws of physics (*e.g.*, Newton's equations, energy conservation, or flow balance) to derive ODE or PDE models. *Empirical modeling* or *identification* consists of developing models from observed or collected data. The a priori assumptions used in empirical modeling can vary from weak to strong: in a "black box" approach, only a few basic assumptions are made, for example, linearity and time-invariance of the system, whereas in a physical model identification approach, a physical model structure is assumed, and the observed or collected data is used to determine good values for these parameters. Mathematical models of a system are often built up from models of subsystems, which may have been developed using different types of modeling.

Often, several models are developed, varying in complexity and fidelity. A simple model might capture some of the basic features and characteristics of the system, noises, or commands; a simple model can simplify the design, simulation, or analysis of the control system, at the risk of inaccuracy. A complex model could be very detailed and describe the system accurately, but a complex model can greatly complicate the design, simulation, or analysis of the system.

1.1.3 Controller Design

Controller design is the topic of this book. The *controller* or *control law* describes the algorithm or signal processing used by the control processor to generate the actuator signals from the sensor and command signals it receives.

Controllers vary widely in complexity and effectiveness. Simple controllers include the *proportional* (P), the *proportional plus derivative* (PD), the *proportional plus integral* (PI), and the *proportional plus integral plus derivative* (PID) controllers, which are widely and effectively used in many industries. More sophisticated controllers include the *linear quadratic regulator* (LQR), the estimated-state-feedback controller, and the *linear quadratic Gaussian* (LQG) controller. These sophisticated controllers were first used in state-of-the-art aerospace systems, but are only recently being introduced in significant numbers.

Controllers are designed by many methods. Simple P or PI controllers have only a few parameters to specify, and these parameters might be adjusted empirically, while the control system is operating, using “tuning rules”. A controller design method developed in the 1930’s through the 1950’s, often called *classical controller design*, is based on the 1930’s work on the design of vacuum tube feedback amplifiers. With these heuristic (but very often successful) techniques, the designer attempts to synthesize a compensation network or controller with which the closed-loop system performs well (the terms “synthesize”, “compensation”, and “network” were borrowed from amplifier circuit design).

In the 1960’s through the present time, state-space or “modern” controller design methods have been developed. These methods are based on the fact that the solutions to some optimal control problems can be expressed in the form of a feedback law or controller, and the development of efficient computer methods to solve these optimal control problems.

Over the same time period, researchers and control engineers have developed methods of controller design that are based on extensive computing, for example, numerical optimization. This book is about one such method.

1.1.4 Controller Implementation

The signal processing algorithm specified by the controller is implemented on the control processor. Commercially available control processors are generally restricted to logic control and specific types of control laws such as PID. Custom control processors built from general-purpose microprocessors or analog circuitry can implement a very wide variety of control laws. General-purpose *digital signal processing* (DSP) chips are often used in control processors that implement complex control laws. Special-purpose chips designed specifically for control processors are also now available.

1.1.5 Control System Testing, Validation, and Tuning

Control system testing may involve:

- extensive computer simulations with a complex, detailed mathematical model,
- real-time simulation of the system with the actual control processor operating ("hardware in the loop"),
- real-time simulation of the control processor, connected to the actual system to be controlled,
- field tests of the control system.

Often the controller is modified after installation to optimize the actual performance, a process known as tuning.

1.2 Goals of Controller Design

A well designed control system will have desirable performance. Moreover, a well designed control system will be tolerant of imperfections in the model or changes that occur in the system. This important quality of a control system is called *robustness.*

1.2.1 Performance Specifications

Performance specifications describe how the closed-loop system should perform. Examples of performance specifications are:

- *Good regulation against disturbances.* The disturbances or noises that act on the system should have little effect on some critical variables in the system. For example, an aircraft may be required to maintain a constant bearing despite wind gusts, or the variations in the demand on a power generation and distribution system must not cause excessive variation in the line frequency. The ability of a control system to attenuate the effects of disturbances on some system variables is called *regulation.*
- *Desirable responses to commands.* Some variables in the system should respond in particular ways to command inputs. For example, a change in the commanded bearing in an aircraft control system should result in a change in the aircraft bearing that is sufficiently fast and smooth, yet does not excessively overshoot or oscillate.
- *Critical signals are not too big.* Critical signals always include the actuator signals, and may include other signals in the system. In an industrial process

control system, for example, an actuator signal that goes to a pump must remain within the limits of the pump, and a critical pressure in the system must remain below a safe limit.

Many of these specifications involve the notion that a signal (or its effect) is small; this is the subject of chapters 4 and 5.

1.2.2 Robustness Specifications

Robustness specifications limit the change in performance of the closed-loop system that can be caused by changes in the system to be controlled or differences between the system to be controlled and its model. Such *perturbations* of the system to be controlled include:

- The characteristics of the system to be controlled may change, perhaps due to component drift, aging, or temperature coefficients. For example, the efficiency of a pump used in an industrial process control system may decrease, over its life time, to 70% of its original value.

- The system to be controlled may have been inaccurately modeled or identified, possibly intentionally. For example, certain structural modes or nonlinearities may be ignored in an aircraft dynamics model.

- Gross failures, such as a sensor or actuator failure, may occur.

Robustness specifications can take several forms, for example:

- *Low differential sensitivities.* The derivative of some closed-loop quantity, with respect to some system parameter, is small. For example, the response time of an aircraft bearing to a change in commanded bearing should not be very sensitive to aerodynamic pressure.

- *Guaranteed margins.* The control system must have the ability to meet some performance specification despite some specific set of perturbations. For example, we may require that the industrial process control system mentioned above continue to have good regulation of product flow rate despite any decrease in pump effectiveness down to 70%.

1.2.3 Control Law Specifications

In addition to the goals and specifications described above, there may be constraints on the control law itself. These *control law specifications* are often related to the implementation of the controller. Examples include:

- The controller has a specific form, *e.g.*, PID.

- The controller is linear and time-invariant (LTI).
- In a control system with many sensors and actuators, we may require that each actuator signal depend on only one sensor signal. Such a controller is called *decentralized*, and can be implemented using many noncommunicating control processors.
- The controller must be implemented using a particular control processor. This specification limits the complexity of the controller.

1.2.4 The Controller Design Problem

Once the system to be controlled has been designed and modeled, and the designer has identified a set of design goals (consisting of performance goals, robustness requirements, and control law constraints), we can pose the controller design problem:

> **The controller design problem:** Given a model of the system to be controlled (including its sensors and actuators) and a set of design goals, find a suitable controller, or determine that none exists.

Controller design, like all engineering design, involves tradeoffs; by *suitable*, we mean a satisfactory compromise among the design goals. Some of the tradeoffs in controller design are intuitively obvious: *e.g.*, in mechanical systems, it takes larger actuator signals (forces, torques) to have faster responses to command signals. Many other tradeoffs are not so obvious.

In our description of the controller design problem, we have emphasized the determination of whether or not there is any controller that provides a suitable tradeoff among the goals. This aspect of the controller design problem can be as important in control engineering as finding or synthesizing an appropriate controller when one exists. If it can be determined that no controller can achieve a suitable tradeoff, the designer must:

- relax the design goals, or
- redesign the system to be controlled, for example by adding or relocating sensors or actuators.

In practice, existing controller design methods are often successful at finding a suitable controller, when one exists. These methods depend upon talent, experience, and a bit of luck on the part of the control engineer. If the control engineer is successful and finds a suitable controller, then of course the controller design problem has been solved. However, if the control engineer *fails* to design a suitable controller, then he or she *cannot* be sure that there is no suitable controller, although the control engineer might suspect this. Another design approach or method (or indeed, control engineer) could find a suitable controller.

1.3 Control Engineering and Technology

1.3.1 Some Advances in Technology

Control engineering is driven by available technology, and the pace of the relevant technology advances is now rapid. In this section we mention a few of the advances in technology that currently have, or will have, an impact on control engineering. More specific details can be found in the Notes and References at the end of this chapter.

Integrated and Intelligent Sensors

Over the past decade the technology of *integrated sensors* has been developed. Integrated sensors are built using the techniques of microfabrication originally developed for integrated circuits; they often include the signal conditioning and interface circuitry on the same chip, in which case they are called *intelligent sensors.* This signal conditioning might include, for example, temperature compensation. Integrated sensors promise greater reliability and linearity than many conventional sensors, and because they are typically cheaper and smaller than conventional sensors, it will be possible to incorporate many more sensors in the design of control systems than is currently done.

Another example of a new sensor technology is the *Global Positioning System* (GPS). GPS position and velocity sensors will soon be available for use in control systems.

Actuator Technology

Significant improvements in actuator technology have been made. For example, direct-drive brushless DC motors are more linear and have higher bandwidths than the motors with brushes and gears (and stiction and backlash) that they will replace. As another example, the trend in aircraft design is towards many actuators, such as canards and vectored thrust propulsion systems.

Digital Control Processors

Over the last few decades, the increase in control processor power and simultaneous decrease in cost has been phenomenal, especially for digital processors such as general-purpose microprocessors, digital signal processors, and special-purpose control processors. As a result, the complexity of control laws that can be implemented has increased dramatically. In the future, custom or semicustom chips designed specifically for control processor applications will offer even more processing power.

Computer-Aided Control System Design and Analysis

Over the past decade we have seen the rise of computer-aided control system design (CACSD). Great advances in available computing power (*e.g.*, the engineering workstation), together with powerful software, have automated or eased many of the tasks of control engineering:

- *Modeling.* Sophisticated programs can generate finite element models or determine the kinematics and dynamics of a mechanical system from its physical description. Software that implements complex identification algorithms can process large amounts of experimental data to form models. Interactive and graphics driven software can be used to manipulate models and build models of large systems from models of subsystems.

- *Simulation.* Complex models can be rapidly simulated.

- *Controller design.* Enormous computing power is now available for the design of controllers. This last observation is of fundamental importance for this book.

1.3.2 Challenges for Controller Design

The technology advances described above present a number of challenges for controller design:

- *More sensors and actuators.* For only a modest cost, it is possible to incorporate many more sensors, and possibly more actuators, into the design of a system. Clearly the extra information coming from the sensors and the extra degrees of freedom in manipulating the system make better control system performance possible. The challenge for controller design is to take advantage of this extra information and degrees of freedom.

- *Higher quality systems.* As higher quality sensors and actuators are incorporated into the system, the system behavior becomes more repeatable and can be more accurately modeled. The challenge for controller design is to take advantage of this more detailed knowledge of the system.

- *More powerful control processors.* Very complex control laws can be implemented using digital control processors. Clearly a more complex control law could improve control system performance (it could also degrade system performance, if improperly designed). The challenge for controller design is to fully utilize the control processor power to achieve better control system performance.

  In particular, control law specifications should be examined carefully. Historically relevant measures of control law complexity, such as the order of an LTI controller, are now less relevant. For example, the order of the compensator used in a vacuum tube feedback amplifier is the number of inductors and capacitors needed to synthesize the compensation network, and was therefore related to cost, size, and reliability. On a particular digital control processor, however, the order of the controller is essentially unrelated to cost, size, and reliability.

- *Powerful computers to design controllers.* The challenge for controller design is to productively use the enormous computing power available. Many current methods of computer-aided controller design simply automate procedures developed in the 1930's through the 1950's, for example, plotting root loci or Bode plots. Even the "modern" state-space and frequency-domain methods (which require the solution of algebraic Riccati equations) greatly underutilize available computing power.

1.4 Purpose of this Book

The main purpose of this book is to describe how the controller design problem can be solved for a restricted set of systems and a restricted set of design specifications, by combining recent theoretical results with recently developed numerical convex optimization techniques.

The restriction on the systems is that they must be linear and time-invariant (LTI). The restriction on the design specifications is that they be *closed-loop convex*, a term we shall describe in detail in chapter 6. This restricted set of design specifications includes a wide class of performance specifications, a less complete class of robustness specifications, and essentially none of the control law specifications.

The basic approach involves directly designing a good closed-loop response, as opposed to designing an open-loop controller that yields a good closed-loop response. We will show that a wide variety of important practical constraints on system performance can be formulated as convex constraints on the response of the closed-loop system. These are the specifications that we call closed-loop convex.

Given a system that is LTI, and a set of closed-loop convex design specifications, the controller design problem can be cast as a convex optimization problem, and consequently, can be effectively solved. This means that if the specifications are achievable, we can find a controller that meets the specifications; if the specifications are not achievable, this fact can be determined, *i.e.*, we will *know* that the specifications are not achievable. In contrast, the designer using a classical controller design scheme is only *likely* to find a controller that meets a set of specifications that is achievable; and, of course, certain not to find a controller that meets a set of specifications that is not achievable. Many controller design techniques do not have any way to determine unambiguously that a set of specifications is not achievable.

For controller design problems of the restricted form, we shall show how to determine which specifications can be achieved and which cannot, and therefore how the *limits of performance* can be determined for a given system and control configuration.

No matter which controller design method is used by the engineer, knowledge of the achievable performance is extremely valuable practical information, since it provides an absolute yardstick against which any designed controller can be compared. To know that a certain candidate controller that is easily implemented, or has some other advantage, achieves regulation only 10% worse than the best regulation achievable by *any* LTI controller, is a strong point in favor of the design. In this sense, this book is not about a particular controller design method or synthesis procedure; rather it is about a method of determining what specifications (of a large but restricted class) can be met using any controller design method, for a given system and control configuration.

We have in addition several subsidiary goals, some of which we have already mentioned. The first is to develop a framework in which we can precisely formulate the controller design problem which we vaguely described above. Our experience suggests that carefully formulating a real controller design problem in the framework we develop will help identify the critical issues and design tradeoffs. This clarification is useful in practical controller design.

We also hope to initiate a discussion of how we can apply the enormous computing power that will soon be available to the controller design problem, beyond, for example, solving the algebraic Riccati equations of "modern" controller design methods. In this book we start this discussion with a specific suggestion: solving convex nondifferentiable optimization problems.

1.4.1 An Example

We can demonstrate some of the main points of this book with an example. We will consider a specific system that has one actuator and one output that is supposed to track a command input, and is affected by some noises; the system is described in section 2.4 (and many other places throughout the book), but the details are not relevant for this example.

Goals for the design of a controller for this system might be:

- *Good RMS regulation*, *i.e.*, the root-mean-square (RMS) value of the output, due to the noises, should be small.
- *Low RMS actuator effort*, *i.e.*, the RMS value of the actuator signal should be small.

It is intuitively clear that by using a larger actuator signal, we may improve the regulation, since we can expend more effort counteracting the effect of the noises. We will see in chapter 12 that the exact nature of this tradeoff between RMS regulation and RMS actuator effort can be determined; it is shown in figure 1.2.

The shaded region shows every pair of RMS regulation and RMS actuator effort specifications that can be achieved by a controller; the designer must, of course, pick one of these.

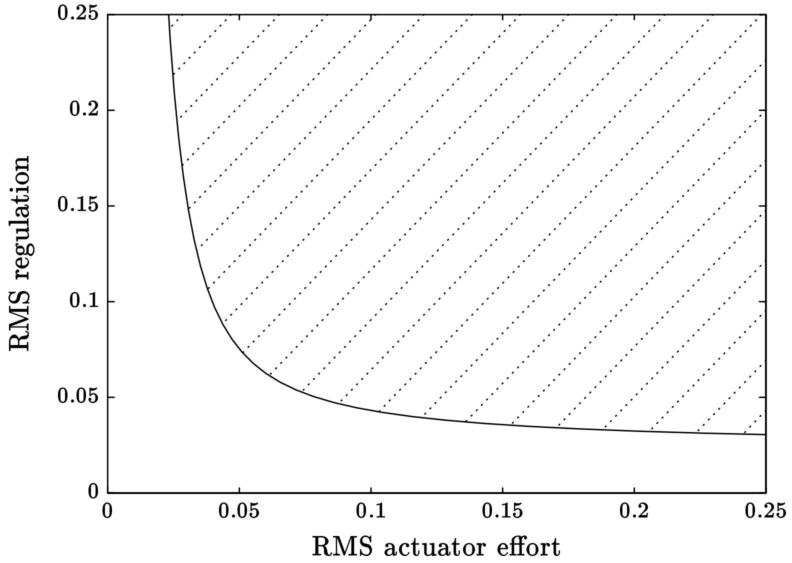


Figure 1.2 The shaded region shows specifications on RMS actuator effort and RMS regulation that are achievable. The unshaded region, at the lower left, shows specifications that no controller can achieve: this region shows a fundamental limit of performance for this system.

The *unshaded* region at the lower left is very important: it consists of RMS regulation and RMS actuator effort specifications that cannot be achieved by any controller, no matter which design method is used. This unshaded region therefore describes a *fundamental limit of performance* for this system. It tells us, for example, that if we require an RMS regulation of 0.05, then we *cannot* simultaneously achieve an RMS actuator effort of 0.05.

Each shaded point in figure 1.2 represents a possible design; we can view many controller design methods as "rummaging around in the shaded region". If the designer knows that a point is shaded, then the designer can find a controller that achieves the corresponding specifications, if the designer is clever enough. On the other hand, each unshaded point represents a limit of performance for our system. Knowing that a point is unshaded is perhaps disappointing, but still very useful information for the designer.

The reader may know that this tradeoff of RMS regulation against RMS actuator effort can be determined using LQG theory. The main point of this book is that for a much wider class of specifications, a similar tradeoff curve can be computed. Suppose, for example, that we add the following specification to our goals above:

- *Command to output overshoot limit*, *i.e.*, the step response overshoot of the closed-loop system, from the command to the output, does not exceed 10%.

Of course, intuition tells us that by adding this specification, we make the design problem “harder”: certain RMS regulation and RMS actuator effort specifications that could be achieved without this new specification will no longer be achievable once we impose it.

In this case there is no analytical theory, such as LQG, that shows us the exact tradeoff. The methods of this book, however, can be used to determine the exact tradeoff of RMS regulation versus RMS actuator effort with the overshoot limit imposed. This tradeoff is shown in figure 1.3. The dashed line, below the shaded region of achievable specifications, is the tradeoff boundary when the overshoot limit is not imposed. The “lost ground” represents the cost of imposing the overshoot limit. We can compute this new region because limits on RMS actuator effort, RMS regulation, and step response overshoot are all closed-loop convex specifications.

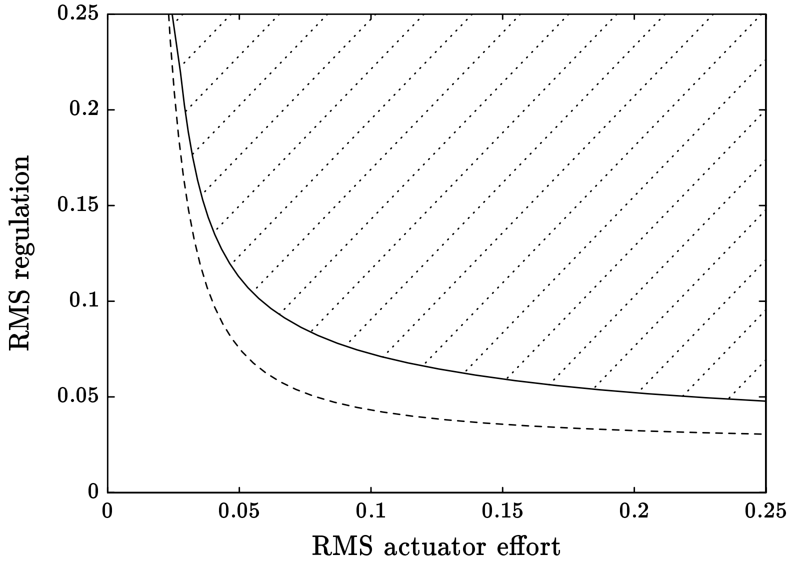


Figure 1.3 The shaded region shows specifications on RMS actuator effort and RMS regulation that are achievable when an additional limit of 10% step response overshoot is imposed; it can be computed using the methods described in this book. The dashed line shows the tradeoff boundary without the overshoot limit; the gap between this line and the shaded region shows the cost of imposing the overshoot limit.

In contrast, suppose that instead of the overshoot limit, we impose the following control law constraint:

- *The controller is proportional plus derivative (PD)*, *i.e.*, the control law has a specific form.

This constraint might be needed to implement the controller using a specific commercially available control processor. This specification is *not* closed-loop convex, so the methods described in this book *cannot* be used to determine the exact tradeoff between RMS actuator effort and RMS regulation. This tradeoff can be computed, however, using a brute force approach described in the Notes and References, and is shown in figure 1.4. The dashed line is the tradeoff boundary when the PD controller constraint is not imposed. Specifications on RMS actuator effort and RMS regulation that lie in the region between the dashed line and the shaded region can be achieved by some controller, but no PD controller.

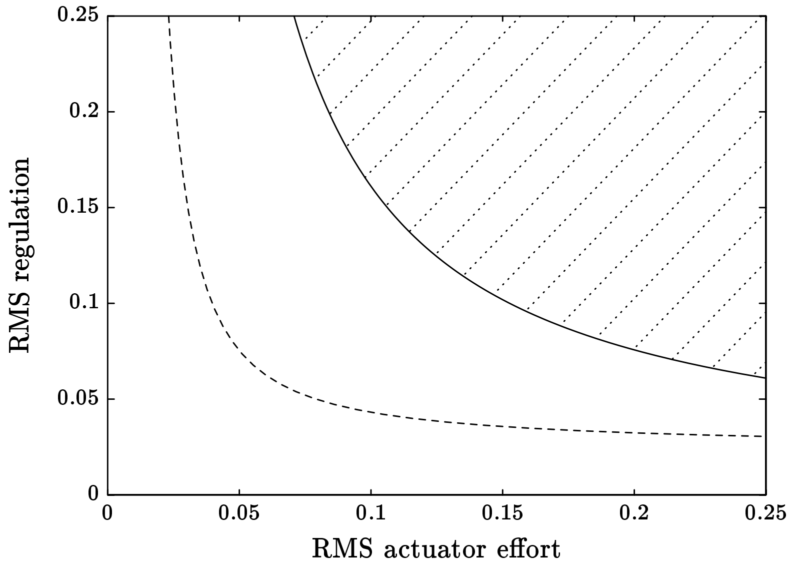


Figure 1.4 The shaded region shows specifications on RMS actuator effort and RMS regulation that can be achieved using a PD controller; it cannot be computed using the methods described in this book. The dashed line shows the tradeoff boundary when no constraint on the control law is imposed.

An important point of this book is that we can compute tradeoffs among closed-loop convex specifications, such as shown in figure 1.3, although it requires more computation than determining the tradeoff for a problem that has an analytical solution, such as shown in figure 1.2; in return, however, a much larger class of problems can be considered. While the computation needed to determine a tradeoff such as shown in figure 1.3 is more than that required to compute the tradeoff shown in figure 1.2, it is *much less* than the computation required to compute tradeoffs such as the one shown in figure 1.4.

The fact that a tradeoff like the one shown in figure 1.4 is much harder to compute than a tradeoff like the one shown in figure 1.3 presents a paradox. To produce figure 1.3 we search over the set of all possible LTI controllers, which has infinite dimension. To produce figure 1.4, however, we search over the set of all PD controllers, which has dimension two. We shall see that convexity makes figure 1.3 “easier” to produce than figure 1.4, even though we must search over a far “larger” set of potential controllers.

1.5 Book Outline

In part I, *A Framework for Controller Design*, we develop a formal framework for many of the concepts described above: the system to be controlled, the control configuration, the controller, and the design goals and objectives for controller design.

In part II, *Analytical Tools*, we first describe *norms* of signals and systems, which can be used to make precise such design goals as “error signals should be made small, while the actuator signals should not be too large”. We then study some important geometric properties that many controller design specifications have, and introduce the important notion of a closed-loop convex design specification.

In part III, *Design Specifications*, we catalog many closed-loop convex design specifications. These design specifications include specifications on the response of the closed-loop system to the various commands and disturbances that may act on it, as well as robustness specifications that limit the sensitivity of the closed-loop system to changes in the system to be controlled.

In part IV, *Numerical Methods*, we describe numerical methods for solving the controller design problem. We start with some controller design problems that have analytic solutions, *i.e.*, can be solved rapidly and exactly using standard methods. We then turn to the numerical solution of controller design problems that can be expressed in terms of closed-loop convex design specifications, but do not have analytic solutions.

In the final chapter we give some discussion of the methods described in this book, as well as some history of the main ideas.

1.5.1 Book Structure

The structure of this book is shown in detail in figure 1.5. From this figure the reader can see that the structure of this book is more vertical than that of most books on linear controller design, which often have parallel discussions of different design techniques. In contrast, this book tells essentially one story, with a few chapters covering related subplots.

A minimal path through the book, which conveys only the essentials of the story, consists of chapters 2, 3, 6, 8–10, and 15. This path results from following every dashed line labeled “experts can skip” in figure 1.5. We note, however, that the term “expert” depends on the context: for example, the reader may be an expert on norms (and thus can safely skip or skim chapters 4 and 5), but not on convex optimization (and thus should read chapters 13 and 14).

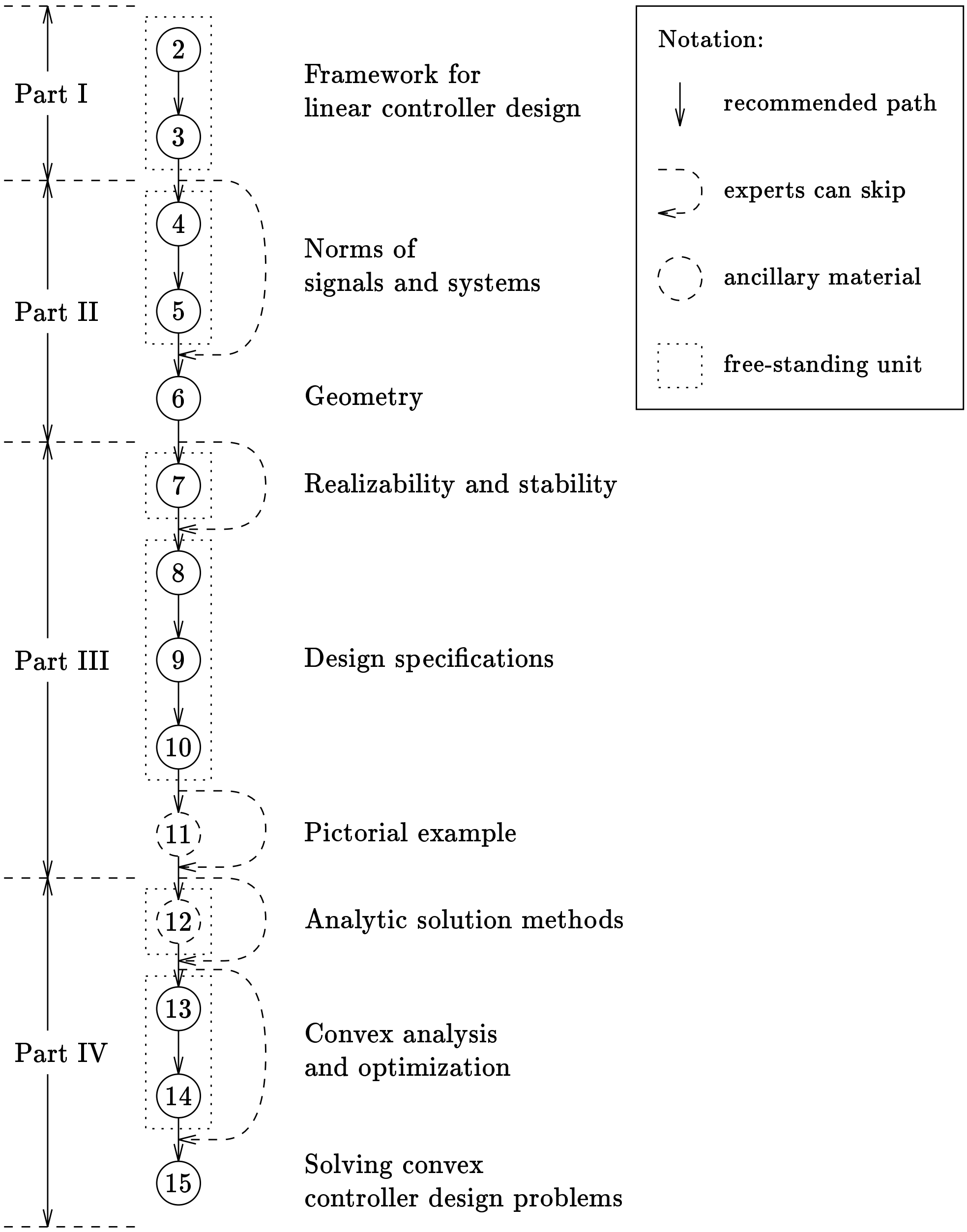

Part I
Part II
Part III
Part IV
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Framework for linear controller design
Norms of signals and systems
Geometry
Realizability and stability
Design specifications
Pictorial example
Analytic solution methods
Convex analysis and optimization
Solving convex controller design problems
Notation:
recommended path
experts can skip
ancillary material
free-standing unit

Figure 1.5 Book layout.

Notes and References

A history of feedback control is given in Mayr [MAY70] and the book [BEN79] and article [BEN76] by Bennett.

Sensors and Actuators

Commercially available sensors and actuators for control systems are surveyed in the books by Hordeski [HOR87] and DeSilva [DES89]; the reader can also consult commercial catalogs and manuals such as [ECC80] and [TRA89B].

The technology behind integrated sensors and actuators is discussed in the survey article by Petersen [PET82]. Commercial implications of integrated sensor technology are discussed in, *e.g.*, [ALL80] (many of the predictions in this article have come to pass over the last decade). Research developments in integrated sensors and actuators can be found in the conference proceedings [TRA89A] (this conference occurs every other year), and the journal *Sensors and Actuators*, published by Elsevier Sequoia. The journal *IEEE Trans. on Electron Devices* occasionally has special issues on integrated sensors and actuators (*e.g.*, Dec. 1979, Jan. 1982).

Overviews of GPS can be found in the book compiled by Wells [WEL87] and the two volume set of reprints published by the Institute of Navigation [GPS84].

Modeling and Identification

Formulation of dynamics equations for physical modeling of mechanical systems is covered in Kane and Levinson [KL85], Crandall et. al. [CKK68], and Cannon [CAN67]. Texts treating identification include those by Box and Jenkins [BJ70], Norton [NOR86], and Ljung [LJU87], which has a complete bibliography.

Linear Controller Design

P and PI controllers have been in use for a long time; for example, the advantage of a PI controller over a P controller is discussed in Maxwell's 1868 article [MAX68], which is one of the first articles on controller design and analysis. PID tuning rules that have been widely used originally appeared in the 1942 article by Ziegler and Nichols [ZN42].

The book *Theory of Servomechanisms* [JNP47], edited by James, Nichols, and Philips, gives a survey of controller design right after World War II. The 1957 book by Newton, Gould, and Kaiser [NGK57] is among the first to adopt an "analytical" approach to controller design (see below). Texts covering classical linear controller design include Bode [BOD45], Ogata [OGA90], Horowitz [HOR63], and Dorf [DOR88]. The root locus method was first described in [EVA50].

Recent books covering classical and state-space methods of linear controller design include Franklin, Powell, and Emami [FPE86] and Chen [CHE87]. Linear quadratic methods for LTI controller design are covered in Athans and Falb [AF66, CH.9], Kwakernaak and Sivan [KS72], Anderson and Moore [AM90], and Bryson and Ho [BH75].

Three recent books on LTI controller design deserve special mention: Lunze's *Robust Multivariable Feedback Design* [LUN89], Maciejowski's *Multivariable Feedback Design* [MAC89], and Vidyasagar's *Control System Synthesis: A Factorization Approach* [VID85]. The first two, [LUN89] and [MAC89], cover a broad range of current topics and linear controller design techniques, although neither covers our central topic, convex closed-loop design. Compared to this book, these two books address more directly the question of how to design linear controllers. Vidyasagar's book [VID85] contains the "recent results" that we referred to at the beginning of section 1.4. Our book can be thought of as an extension or application of the ideas in [VID85].

Digital Control

Digital control systems are covered in the books by Ogata [OGA87], Ackermann [ACK85], and Åström and Wittenmark [AW90]. A recent comprehensive text covering all aspects of digital control systems is by Franklin, Powell, and Workman [FPW90].

Control Processors and Controller Implementation

Programmable logic controllers and other industrial control processors are covered in Warnock [WAR88]. An example of a commercially available special-purpose chip for control systems is National Semiconductor's LM628 precision motion controller [PRE89], which implements a PID control law.

The use of DSP chips as control processors is discussed in several articles and manufacturers' applications manuals. For example, the implementation of a simple controller on a Texas Instruments TMS32010 is described in [SB87], and the implementation of a PID controller on a Motorola DSP56001 is described in [SS89]. Chapter 12 of [FPW90] describes the implementation of a complex disk drive head positioning controller using the Analog Devices ADSP2101. The article [CHE82] describes the implementation of simple controllers on an Intel 2920.

The design of custom integrated circuits for control processors is discussed in [JTP85] and [TL80].

General issues in controller implementation are discussed in the survey paper by Hanselmann [HAN87].

The book [AT90] discusses real-time software used to program general-purpose computers as control processors. Topics covered include implementing the control law, interface to actuators and sensors, communication, data logging, and operator display.

Computers and Control Engineering

Examples of computer-based equipment for control systems engineering include Hewlett-Packard's HP3563A Control Systems Analyzer [HEP89], which automates frequency response measurements and some simple identification procedures, and Integrated Systems' AC-100 control processor [AC100], which allows rapid implementation of a controller for prototyping.

A review of various software packages for structural analysis is given in [NIK86], in particular the chapter [MAC86]. A widely used finite element code is NASTRAN [CIF89]. Computer software packages (based on Kane's method) that symbolically form the system dynamics include SD/EXACT, described in [RS86, SR88] and AUTOLEV, described in [SL88]. Examples of software for system identification are the SYSTEM-ID toolbox [LJU86] for use with MATLAB and the SYSTEM-ID package [MAT88] for use with MATRIX-X (see below).

Examples of controller design software are MATLAB [MLB87] (and [LL87]), MATRIX-X [SFL85, WSG84], DELIGHT-MIMO [PSW85], and CONSOLE [FWK89]. Some of these programs were originally based on the linear algebra software packages LINPACK [DMB79] and EISPACK [SBD76]. A new generation of reliable linear algebra routines is now being developed in the LAPACK project [DEM89]; LAPACK will take advantage of some of the advances in computer hardware, *e.g.*, vector processing.

General discussion of CACSD can be found, for example, in the article [AST83], and the special issue of the *Proceedings of the IEEE* [PIE84]. See also [JH85] and [DEN84].

Determining Limits of Performance

The value of being able to determine that a set of specifications cannot be achieved, and the failing of many controller design methods in this regard, has been noted before. In the 1957 book *Analytical Design of Linear Feedback Controls*, by Newton, Gould, and Kaiser [NGK57, §1.6], we find:

> Unfortunately, the trial and error design method is beset with certain fundamental difficulties, which must be clearly understood and appreciated in order to employ it properly. From both a practical and theoretical viewpoint its principal disadvantage is that it cannot recognize an inconsistent set of specifications.
>
> ...The analytical design procedure has several advantages over the trial and error method, the most important of which is the facility to detect immediately and surely an inconsistent set of specifications. The designer obtains a "yes" or "no" answer to the question of whether it is possible to fulfill any given set of specifications; he is not left with the haunting thought that if he had tried this or that form of compensation he might have been able to meet the specifications.
>
> ...Even if the reader never employs the analytical procedure directly, the insight that it gives him into linear system design materially assists him in employing the trial and error design procedure.

This book is about an *analytical design procedure*, in the sense in which the phrase is used in this quote.

There are a few results in classical controller design that can be used to determine some specifications that cannot be achieved. The most famous is Bode's integral theorem [BOD45]; a more recent result is due to Zames [ZAM81, ZF83]. These results were extended to unstable plants by Freudenberg and Looze [FL85, FL88], and plants with multiple sensors and actuators by Boyd and Desoer [BD85].

The article by Barratt and Boyd [BB89] gives some specific examples of using convex optimization to numerically determine the limits of performance of a simple control system. The article by Boyd, Barratt, and Norman [BBN90] gives an overview of the closed-loop convex design method.

About the Example in Section 1.4.1

The plant used is described in section 2.4; the process and sensor noises are described in chapter 11, and the precise definitions of RMS actuator effort, RMS regulation, and step response overshoot are given in chapters 3, 5, and 8.

The method used to determine the shaded region in figure 1.2 is explained in section 12.2.1; a similar figure appears in Kwakernaak and Sivan [KS72, P205]. The method used to determine the shaded region in figure 1.3 is explained in detail in chapter 15.

The exact form of the PD controller was

$$K_{\mathrm{pd}}(s) = \frac{k_{\mathrm{p}} + s k_{\mathrm{d}}}{(1 + s/20)^2}, \tag{1.1}$$

where $k_{\rm p}$ and $k_{\rm d}$ are constants, the proportional and derivative gains, respectively.

Determining the shaded region shown in figure 1.4 required the solution of many *global* optimization problems in the variables $k_{\rm p}$ and $k_{\rm d}$. We first used a numerical local optimization method designed especially for parametrized controller design problems with RMS specifications; see, *e.g.*, the survey by Mäkilä and Toivonen [MT87]. This produced a region that was likely, but not certain, to be the whole region of achievable specifications. To verify that we had found the whole region, we used the Routh conditions to determine analytically the region in the $k_{\rm p}$, $k_{\rm d}$ plane that corresponds to stable closed-loop systems; this region was very finely gridded and the RMS actuator effort and regulation checked over this grid. This exhaustive search revealed that for this example, the local optimization method had indeed found the global minima; it simply took an enormous amount of computation to verify that the solutions were global. Of course, in general, local methods can miss the global minimum. (See the discussion in section 14.6.4.)

A more sophisticated global optimization algorithm, such as branch-and-bound, could have been used (see, *e.g.*, Pardalos and Rosen [PR87]). But all known global optimization algorithms involve computation that in the worst case grows exponentially with the number of variables. A similar five parameter global optimization problem would probably be computationally intractable.

Part I

A FRAMEWORK FOR CONTROLLER DESIGN

Chapter 2

A Framework for Control System Architecture

In this chapter we describe a formal framework for what we described in chapter 1 as the system to be controlled, the control configuration, and the control law or controller.

2.1 Terminology and Definitions

We start with a mathematical model of the system to be controlled that includes the sensors and actuators. We refer to the independent variables in this model as the *input signals* or *inputs*, and the dependent variables as the *output signals* or *outputs.* In this section we describe an important further division of these signals into those the controller can access and those it cannot.

The inputs to the model include the actuator signals, which come from the controller, and other signals that represent noises and disturbances acting on the system. We will see in chapter 10 that it may be advantageous to include among these input signals some *fictitious inputs.* These fictitious inputs are not used to model any specific noise or disturbance; they allow us to ask the question, "*what if* a signal were injected here?".

Definition 2.1: *The inputs to the model are divided into two vector signals:*

- *The* actuator signal vector*, denoted u, consists of the inputs to the model that can be* manipulated *by the controller. The actuator signal vector u is exactly the signal vector generated by the control processor.*

- *The* exogenous input vector*, denoted w, consists of all other input signals to the model.*

The number of actuator and exogenous input signals, *i.e.*, the sizes of u and w, will be denoted n_u and n_w, respectively.

Our model of the system must provide as output every signal that we care about, *i.e.*, every signal needed to determine whether a proposed controller is an acceptable design. These signals include the signals we are trying to regulate or control, all actuator signals (u), all sensor signals, and perhaps important internal variables, for example stresses on various parts of a mechanical system.

Definition 2.2: *The outputs of the model consist of two vector signals:*

- *The* sensor signal vector*, denoted y, consists of output signals that are* accessible *to the controller. The sensor signal y is exactly the input signal vector to the control processor.*
- *The* regulated outputs signal vector*, denoted z, consists of every output signal from the model.*

The number of sensor and regulated output signals, *i.e.*, the sizes of y and z, will be denoted n_y and n_z, respectively.

We refer to the model of the system, with the two vector input signals w and u and the two vector output signals z and y, as the *plant*, shown in figure 2.1. Even though z includes the sensor signal y, we draw them as two separate signal vectors.

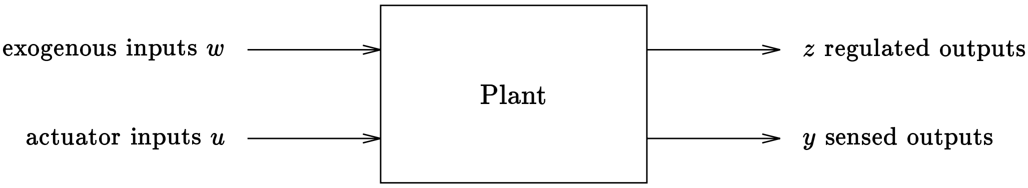


Figure 2.1 The plant inputs are partitioned into signals manipulable by the controller (u) and signals not manipulable by the controller (w). Among all of the plant outputs (z) are the outputs that the controller has access to (y).

When the control system is operating, the controller processes the sensor signal y to produce the actuator signal u, as shown in figure 2.2. We refer to this connection of the plant and controller as the *closed-loop system.* The closed-loop system has input w and output z.

2.1.1 Comparison to the Classical Plant

Our notion of the plant differs from that used in classical control texts in several ways. First, our plant includes information about which signals are accessible to the controller, whereas in classical control, this is often *side information* given along with the classical plant in the controller design problem. For example, what would

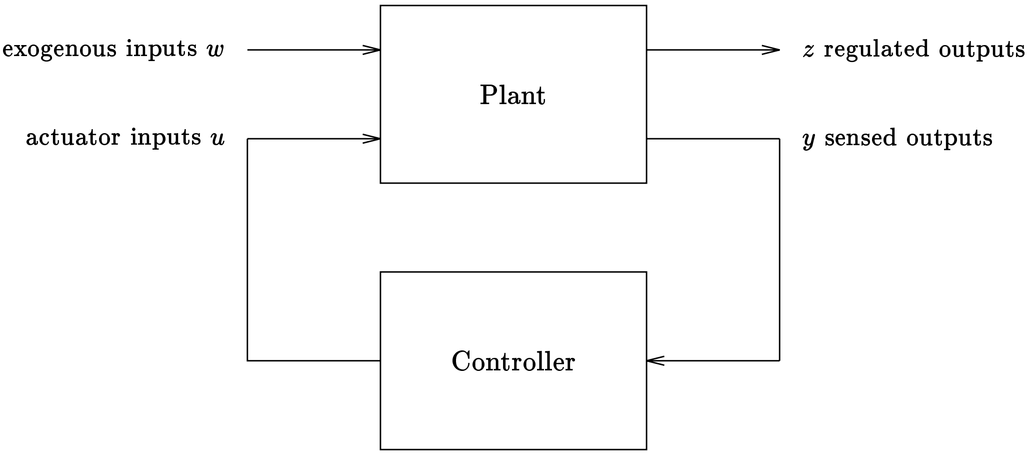


Figure 2.2 The closed-loop system. The controller processes the sensed signals to produce the actuator signals.

be called a "state-feedback" control system and an "output feedback" control system for a given classical plant, we describe as closed-loop systems for *two different plants*, since the sensed signals (our y) differ in the two systems. Similarly, the distinction made in classical control between one degree-of-freedom and two degree-of-freedom control systems is expressed in our framework as a difference in *plants*.

Second, our plant includes information about where exogenous commands such as disturbances and noises enter the system. This information is also given as side information in classical control problems, if it is given at all. In classical control, the disturbances *might* be indicated in a block diagram showing where they enter the system; some important exogenous inputs and regulated variables are commonly left out, since it is expected that the designer will intuitively know that an acceptable design cannot excessively amplify, say, sensor noise.

Similarly, our plant makes the signal z explicit—the idea is that z contains every signal that is important to us. In classical control texts we find extensive discussion of critical signals such as actuator signals and tracking errors, but no attempt is made to list *all* of the critical signals for a given problem.

If w and z contain every signal about which we will express a constraint or specification, then candidate closed-loop systems can be evaluated by simulations or tests involving only the signals w and z. Thus, specifications (in the sense of a contract) for the control system could be written in terms of w and z only.

We believe that the task of defining the signals u, w, y, and z for a system to be controlled is itself useful. It will help in forming sensible specifications for a controller to be designed and it helps in identifying the simulations that should be done to evaluate a candidate controller.

2.1.2 Command Inputs and Diagnostic Outputs

The reader may have noticed that command signals entering the controller and diagnostic signals produced by the controller do not appear explicitly in figure 2.2, even though they do in figures 1.1 and 2.3. In this section we show how these operator interface signals are treated in our framework. Our treatment of command signals simply follows the definitions above: if the command signal is directly accessible to the controller, then it is included in the signal y. There remains the question of how the command signals enter the plant. Again, we follow the definitions above: the command signals are plant inputs, not manipulable by the controller (they are presumably manipulable by some external agent issuing the commands), and so they must be *exogenous inputs*, and therefore included in w. Often, exogenous inputs that are commands pass directly through the plant to some of the components of y, as shown in figure 2.4. Diagnostic outputs are treated in a similar way.

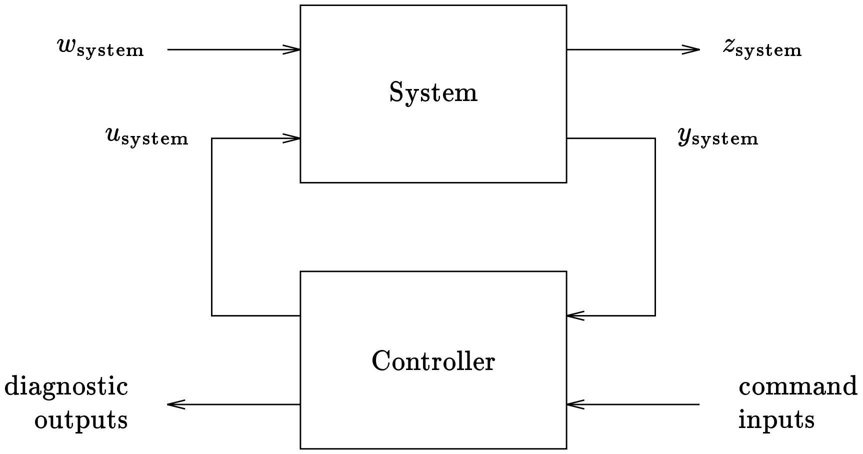


Figure 2.3 The controller may accept command signals as inputs and produce diagnostic and warning signals as outputs (see figure 1.1). This is described in our framework by including the command signals in w and y, and the diagnostic and warning signals in u and z, as shown in figure 2.4.

2.2 Assumptions

In this book we make the following assumptions:

Assumption 2.1: *The signals w, u, z, and y are real vector-valued continuous-time signals, i.e., functions from a real, nonnegative time variable into the appropriately dimensioned vector space:*

$$w : \mathbf{R}_+ \to \mathbf{R}^{n_w}, \qquad u : \mathbf{R}_+ \to \mathbf{R}^{n_u}, \qquad z : \mathbf{R}_+ \to \mathbf{R}^{n_z}, \qquad y : \mathbf{R}_+ \to \mathbf{R}^{n_y}.$$

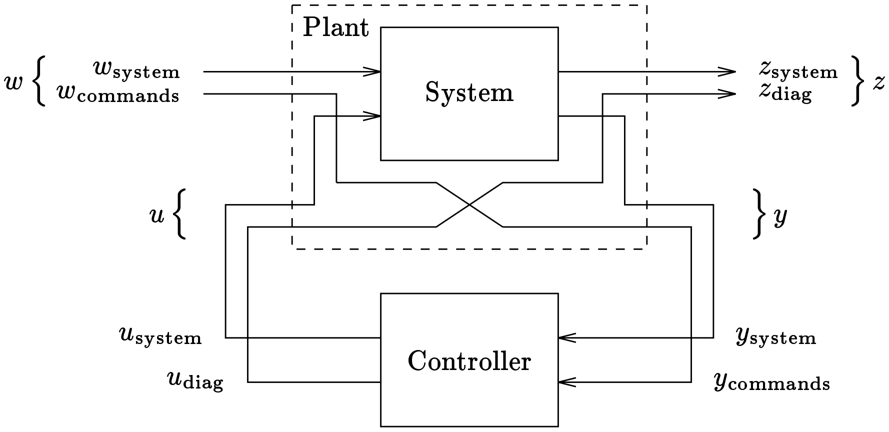


Figure 2.4 In our framework the command signals are included in w and pass through the plant to y, so that the controller has access to them. Similarly, diagnostic signals produced by the controller are included in u, and pass through the plant to z.

Assumption 2.2: *The plant is linear and time-invariant (LTI) and lumped, i.e., described by a set of constant coefficient linear differential equations with zero initial conditions.*

Assumption 2.3: *The controller is also LTI and lumped.*

The differential equations referred to will be described in detail in section 2.5.

2.2.1 Some Comments

About assumption 2.2:

- Many important plants are *highly nonlinear*, *e.g.*, mechanical systems that undergo large motions.
- Assumption 2.2 is always an approximation, only good for certain ranges of values of system signals, over certain time intervals or frequency ranges, and so on.

About assumption 2.3:

- Even if the plant is LTI, it is still a restriction for the controller to be LTI.
- We have already noted that control systems that use digital control processors process sampled signals. These controllers are linear, but time-varying.

We wish to emphasize that our restriction to LTI plants and controllers is hardly a minor restriction, even if it is a commonly made one. Nevertheless, we believe the material of this book is still of great value, for several reasons:

- Many nonlinear plants *are* well modeled as LTI systems, especially in regulator applications, where the goal is to keep the system state near some operating point.

- A controller that is designed on the basis of an approximate linear model of a nonlinear plant often works well with the nonlinear plant, even if the linear model of the plant is not particularly accurate. (See the Notes and References at the end of this chapter.)

- Some of the effects of plant nonlinearities can be accounted for in the framework of an LTI plant and controller. (See chapter 10.)

- Linear control systems often form the core or basis of control systems designed for nonlinear systems, for example in gain-scheduled or adaptive control systems. (See the Notes and References at the end of this chapter.)

- A new approach to the control of nonlinear plants, called *feedback linearization*, has recently been developed. If feedback linearization is successful, it reduces the controller design problem for a nonlinear plant to one for which assumption 2.2 holds. (See the Notes and References at the end of this chapter.)

- Even when the final controller is time-varying (*e.g.*, when implemented on a digital control processor), a preliminary design or analysis of achievable performance is usually carried out under the assumption 2.3. This design or analysis then helps the designer select appropriate sample rates.

- The results of this book can be extended to cover linear time-varying plants and controllers, in particular, the design of single-rate or multi-rate digital controllers. (See chapter 16.)

2.2.2 Transfer Matrix Notation

We will now briefly review standard notation for LTI systems. Consider an LTI system with a single (scalar) input a and a single (scalar) output b. Such a system is completely described by its *transfer function*, say G, so that

$$B(s) = G(s)A(s), \tag{2.1}$$

where A and B are the Laplace transforms of the signals a and b respectively:

$$A(s) = \int_0^\infty a(t)e^{-st}\,dt,$$
$$B(s) = \int_0^\infty b(t)e^{-st}\,dt.$$

Equivalently, the signals a and b are related by convolution,

$$b(t) = \int_0^t g(\tau)a(t-\tau)\,d\tau, \tag{2.2}$$

where g is the *impulse response* of the linear system,

$$G(s) = \int_0^\infty g(t)e^{-st}\,dt.$$

We will write (2.1) and (2.2) as

$$b = Ga. \tag{2.3}$$

We will also use the symbol G to denote both the transfer function of the LTI system, and the LTI system itself. An interpretation of (2.3) is that the LTI system G acts on the input signal a to produce the output signal b. We say that G is the transfer function from a to b.

Identical notation is used for systems with multiple inputs and outputs. For example, suppose that an LTI system has n inputs and m outputs. We collect the scalar input signals $a_1, \ldots, a_n$ to form a vector input signal a,

$$a(t) = \begin{bmatrix} a_1(t) \\ \vdots \\ a_n(t) \end{bmatrix},$$

and similarly, we collect the output signals to form the vector signal b. The system is completely characterized by its *transfer matrix*, say G, which is an $m \times n$ *matrix* of transfer functions. All of the previous notation is used to represent the multiple-input, multiple-output (MIMO) system G and its vector input signal a and vector output signal b.

2.2.3 Transfer Matrix Representations

A consequence of assumption 2.2 is that the plant can be represented by a transfer matrix P, with a vector input consisting of the vector signals w and u, and a vector output consisting of the vector signals z and y. Similarly, a consequence of assumption 2.3 is that the controller can be represented by a transfer matrix K, with a vector input y and vector output u. We partition the plant transfer matrix P as

$$P = \begin{bmatrix} P_{zw} & P_{zu} \\ P_{yw} & P_{yu} \end{bmatrix},$$

so that

$$z = P_{zw}w + P_{zu}u, \tag{2.4}$$

$$y = P_{yw}w + P_{yu}u. \tag{2.5}$$

Thus, P_{zw} is the transfer matrix from w to z, P_{zu} is the transfer matrix from u to z, P_{yw} is the transfer matrix from w to y, and P_{yu} is the transfer matrix from u to y. This decomposition is shown in figure 2.5. We will emphasize this partitioning of the plant with dark lines:

$$P = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array}\right].$$

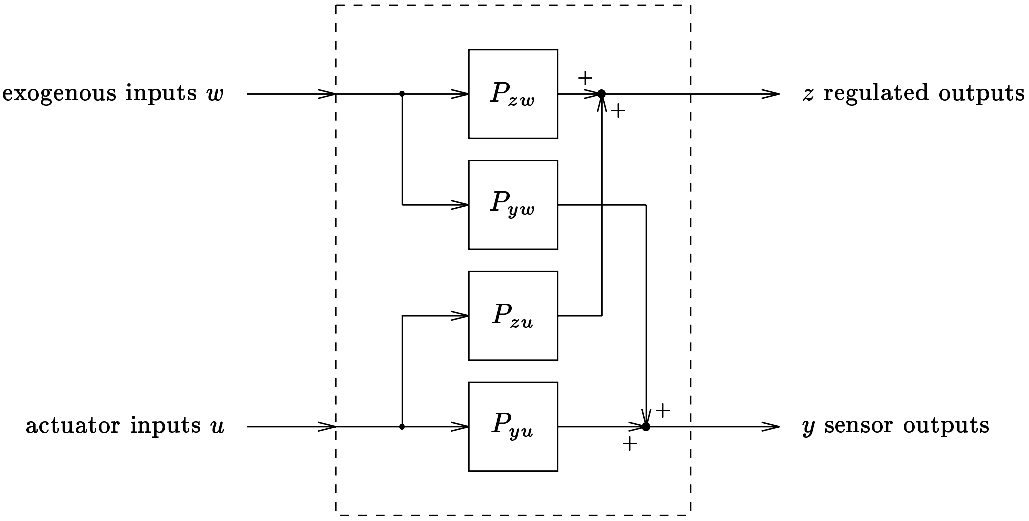


Figure 2.5 The decomposed LTI plant.

Now suppose the controller is operating, so that in addition to (2.4–2.5) we have

$$u = Ky. \tag{2.6}$$

We can solve for z in terms of w to get

$$z = \left(P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}\right) w,$$

provided $\det(I - P_{yu}K)$ is not identically zero, a *well-posedness* condition that we will always assume.

Definition 2.3: *The* closed-loop transfer matrix H *is the transfer matrix from* w *to* z*, with the controller* K *connected to the plant* P*:*

$$H \triangleq P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}. \tag{2.7}$$

Thus

$$z = Hw. \tag{2.8}$$

The entries of the transfer matrix H are the closed-loop transfer functions from each exogenous input to each regulated variable. These entries might represent, for example, closed-loop transfer functions from some disturbance to some actuator, some sensor noise to some internal variable, or some command signal to some actuator signal. The formula (2.7) above shows exactly how each of these closed-loop transfer functions depends on the controller K.

A central theme of this book is that H *should contain every closed-loop transfer function of interest to us.* Indeed, we can arrange for any particular closed-loop transfer function in our system to appear in H, as follows. Consider the closed-loop system in figure 2.6 with two signals A and B which are internal to the plant. If our interest is the transfer function from a signal injected at point A to the signal at point B, we need only make sure that one of the exogenous signals injects at A, and that the signal at point B is one of our regulated variables, as shown in figure 2.7.

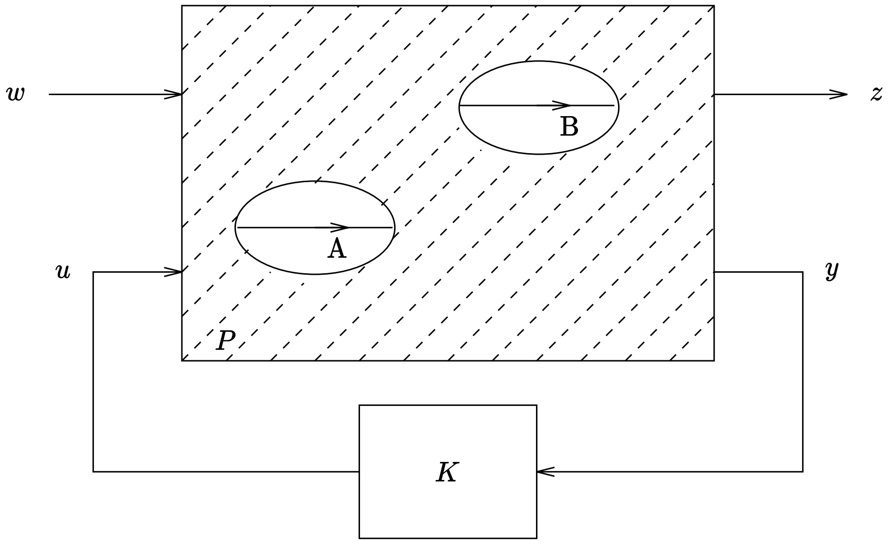


Figure 2.6 Two signals A and B inside the plant P.

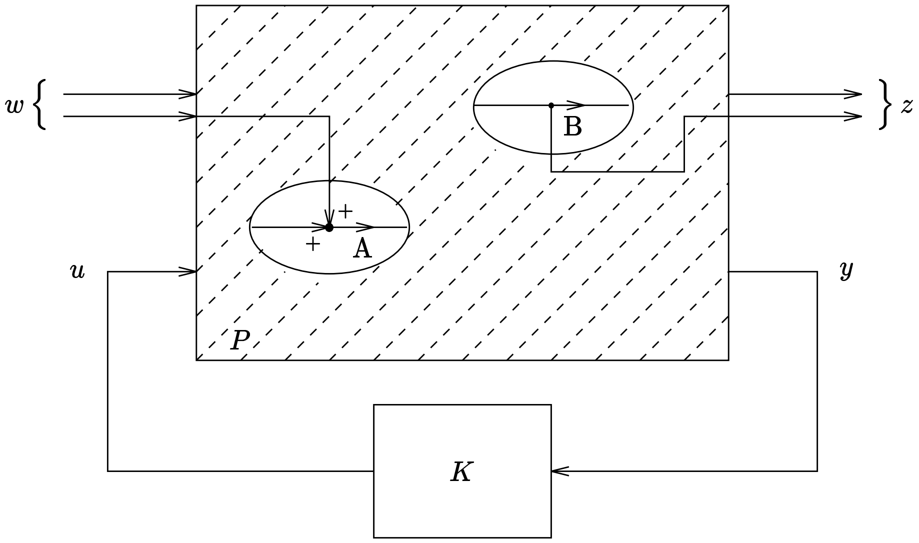


Figure 2.7 Accessing internal signals A and B from w and z.

2.3 Some Standard Examples from Classical Control

In this section we present various examples to illustrate the concepts introduced in this chapter. We will also define some classical control structures that we will periodically refer to.

2.3.1 The Classical Regulator

In this section we consider the classical single-actuator, single-sensor (SASS) regulator system. A conventional block diagram is shown in figure 2.8. We use the symbol P_0 to denote the transfer function of the classical plant. The negative sign at the summing junction reflects the classical convention that feedback should be "negative".

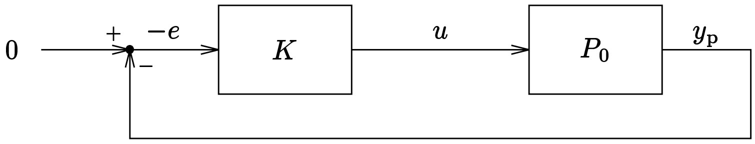


Figure 2.8 Conventional block diagram of a classical single-actuator, single-sensor regulator system.

The signal e is called the *error signal;* it is the difference between the system output y_p and the *reference* or desired output, which is zero for the regulator. The goal of the regulator design is to keep y_{p} small, and u not too large, despite the disturbances that act on the system.

The conventional block diagram in figure 2.8 does not show the disturbances that act on the system, nor does it explicitly show which signals are of interest to us; this is side information in a conventional description of the regulator problem. To cast the classical regulator in our framework, we first add to the block diagram in figure 2.8 inputs corresponding to disturbances and outputs indicating the signals of importance to us. One way to do this is shown in figure 2.9.

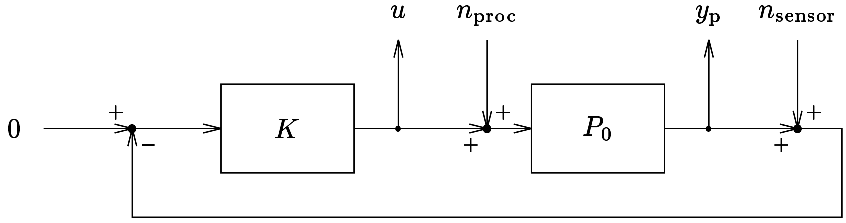


Figure 2.9 Classical regulator system with exogenous inputs and regulated outputs.

The disturbance n_{proc} is an *actuator-referred process noise*; it is the signal that recreates the effect of system disturbances on the system output when it is added to the actuator signal. The disturbance n_{sensor} is a *sensor noise.* Even if this sensor noise is small, its existence in figure 2.9 emphasizes the important fact that the sensor signal $(y_{\mathrm{p}} + n_{\mathrm{sensor}})$ is not exactly the same as the system output that we wish to regulate (y_{p}).

The output signals we have indicated in figure 2.9 are the actuator signal u and the system output y_{p}, since these signals will be important in any regulator design.

The reader can think of the four explicit inputs and output signals we have added to the classical regulator system as *required for a realistic simulation* or *required to evaluate a candidate controller.* For a particular problem, it may be appropriate to include other input or output signals, *e.g.*, the error signal e.

We can now define the signals used in our framework. For the exogenous input vector we take the process noise n_{proc} and the sensor noise n_{sensor},

$$w = \begin{bmatrix} n_{\mathrm{proc}} \\ n_{\mathrm{sensor}} \end{bmatrix}.$$

We take the vector of regulated outputs to consist of the system output y_{p} and the actuator signal u,

$$z = \begin{bmatrix} y_{\mathrm{p}} \\ u \end{bmatrix}.$$

The control input of the plant is just the actuator signal u, and the sensed output is the negative of the system output signal corrupted by the sensor noise:

$$y = -(y_{\mathrm{p}} + n_{\mathrm{sensor}}).$$

Note that this is the signal that enters the controller in figure 2.9.

The plant has three inputs and three outputs; its transfer matrix is

$$P = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array}\right] = \left[\begin{array}{cc|c} P_0 & 0 & P_0 \\ 0 & 0 & 1 \\ \hline -P_0 & -1 & -P_0 \end{array}\right].$$

A block diagram of the closed-loop system is shown in figure 2.10. Using equation (2.7), we find that the closed-loop transfer matrix H from w to z is

$$H = \begin{bmatrix} \dfrac{P_0}{1+P_0K} & -\dfrac{P_0K}{1+P_0K} \\ -\dfrac{P_0K}{1+P_0K} & -\dfrac{K}{1+P_0K} \end{bmatrix}. \tag{2.9}$$

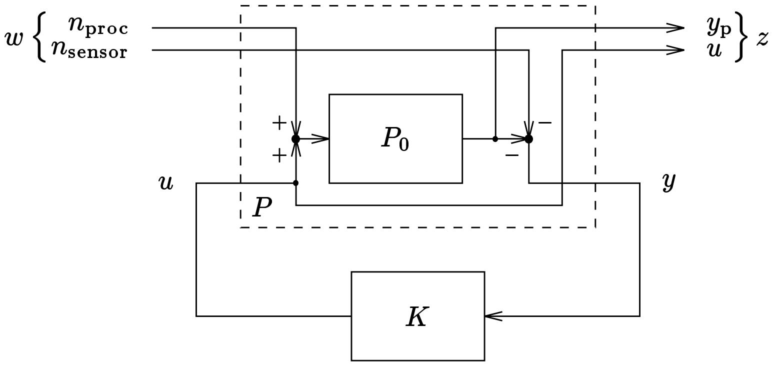


Figure 2.10 The closed-loop regulator.

For future reference, we note the following terms from classical control:

- $L \triangleq P_0K$ is called the *loop gain.*
- $S \triangleq 1/(1+L)$ is called the *sensitivity transfer function.*
- $T \triangleq 1 - S$ is called the *complementary sensitivity transfer function.*

Using these definitions, the classical designer might write the closed-loop transfer matrix H as

$$H = \begin{bmatrix} SP_0 & -T \\ -T & -T/P_0 \end{bmatrix}.$$

Each entry of H, the 2×2 closed-loop transfer matrix from w to z, is significant. The first row consists of the closed-loop transfer functions from the process and sensor noises to the output signal $y_{\rm p}$; our goal is to make these two transfer functions "small" in some appropriate sense. The "size" of these two transfer functions tells us something about the closed-loop *regulation* achieved by our control system. The second row consists of the closed-loop transfer functions from the process and sensor noises to the actuator signal, and thus is related to the actuator effort our control system uses.

The idea is that H contains all the closed-loop transfer functions of interest in our regulator design. Thus, the performance of different candidate controllers could be compared by their associated H's, using (2.7); the specifications for a regulator design could be expressed in terms of the four transfer functions in H.

2.3.2 The Classical 1-DOF Control System

A simple extension of the regulator is the classical *one degree-of-freedom* (1-DOF) *controller*, shown in figure 2.11. In the 1-DOF control system, the reference signal, denoted r, is an external input that can change with time; the regulator is just the 1-DOF controller with the reference input fixed at zero. The basic goal in the design of the 1-DOF control system is to keep the system output $y_{\rm p}$ close to the reference signal r, despite the disturbances $n_{\rm proc}$ and $n_{\rm sensor}$, while ensuring that the actuator signal u is not too large. The difference between the system output signal and the reference signal is called the *tracking error*, and denoted e:

$$e = y_{\rm p} - r.$$

(The tracking error is not shown in figure 2.11 or included in z, although it could be.)

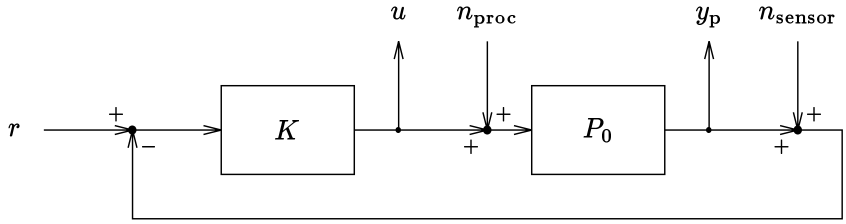


Figure 2.11 Classical 1-DOF control system.

We now describe the 1-DOF control system in our framework. As described in section 2.1.2 the reference input r is an exogenous input, along with the process and sensor noises, so we take the exogenous input vector to be

$$w = \begin{bmatrix} n_{\rm proc} \\ n_{\rm sensor} \\ r \end{bmatrix}.$$

The control input of the plant is again the actuator signal u, and we can take the vector of regulated outputs to be the same as for the regulator:

$$z = \begin{bmatrix} y_\mathrm{p} \\ u \end{bmatrix}.$$

The sensed output y for the 1-DOF control system must be determined carefully. The controller in the 1-DOF control system does not have direct access to the corrupted system output, $y_\mathrm{p} + n_\mathrm{sensor}$. Instead, the controller input is the tracking error signal corrupted by the sensor noise:

$$y = r - y_\mathrm{p} - n_\mathrm{sensor}.$$

The plant has four inputs and three outputs; its transfer matrix is

$$P = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c} P_0 & 0 & 0 & P_0 \\ 0 & 0 & 0 & 1 \\ \hline -P_0 & -1 & 1 & -P_0 \end{array}\right]. \tag{2.10}$$

The 1-DOF control system is shown in our framework in figure 2.12.

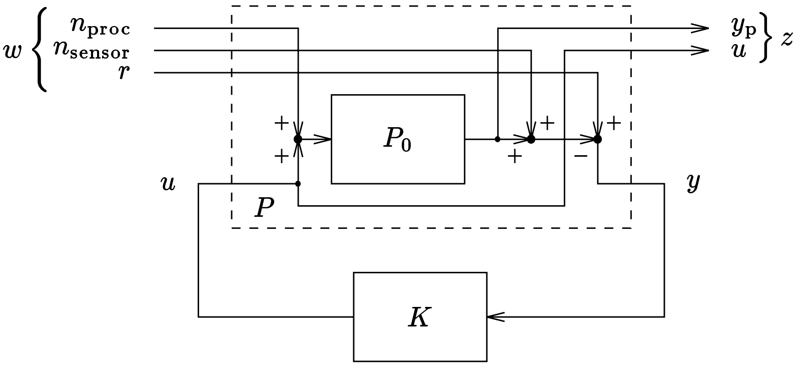


Figure 2.12 The 1-DOF control system in our framework.

The closed-loop transfer matrix H now has three inputs and two outputs:

$$H = \begin{bmatrix} \dfrac{P_0}{1+P_0K} & -\dfrac{P_0K}{1+P_0K} & \dfrac{P_0K}{1+P_0K} \\ -\dfrac{P_0K}{1+P_0K} & -\dfrac{K}{1+P_0K} & \dfrac{K}{1+P_0K} \end{bmatrix}. \tag{2.11}$$

The closed-loop transfer matrix H in this example consists of the closed-loop transfer matrix of the classical regulator, described in section 2.3.1, with a third column appended. The third column consists of the closed-loop transfer functions from the reference signal to the regulated variables. Its first entry, H_{13}, is the transfer function from the reference input to the system output, and is called the *input-output* (I/O) transfer function of the 1-DOF control system. It is the same as the complementary sensitivity transfer function.

2.3.3 The Classical 2-DOF Control System

A generalization of the classical 1-DOF control system is the classical *two degree-of-freedom* (2-DOF) *control system.* A conventional block diagram of the 2-DOF control system is shown in figure 2.13. The key difference between the 1-DOF and 2-DOF control systems is that in the former, the controller processes only the corrupted error signal to produce the actuator signal, whereas in the latter, the controller has access to both the reference and the corrupted system output signals.

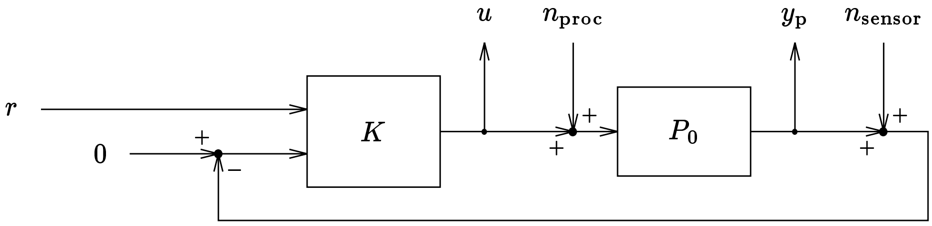


Figure 2.13 Conventional block diagram of a classical 2-DOF control system.

To describe the 2-DOF control system in our framework, we take the same actuator, exogenous input, and regulated variables signals as for the 1-DOF control system:

$$u, \qquad w = \begin{bmatrix} n_{\text{proc}} \\ n_{\text{sensor}} \\ r \end{bmatrix}, \qquad z = \begin{bmatrix} y_{\text{p}} \\ u \end{bmatrix}. \tag{2.12}$$

The sensor signal for the 2-DOF control system is not the same as for the 1-DOF control system; it is

$$y = \begin{bmatrix} -y_{\text{p}} - n_{\text{sensor}} \\ r \end{bmatrix} = \begin{bmatrix} \tilde{y} \\ r \end{bmatrix}. \tag{2.13}$$

The 2-DOF control system is shown in our framework in figure 2.14. The plant has four inputs and four outputs; its transfer matrix is

$$P = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c} P_0 & 0 & 0 & P_0 \\ 0 & 0 & 0 & 1 \\ \hline -P_0 & -1 & 0 & -P_0 \\ 0 & 0 & 1 & 0 \end{array}\right]. \tag{2.14}$$

The controller K in the 2-DOF control system has two inputs and one output. If we write its transfer matrix as

$$K = \begin{bmatrix} K_{\tilde{y}} & K_r \end{bmatrix} \tag{2.15}$$

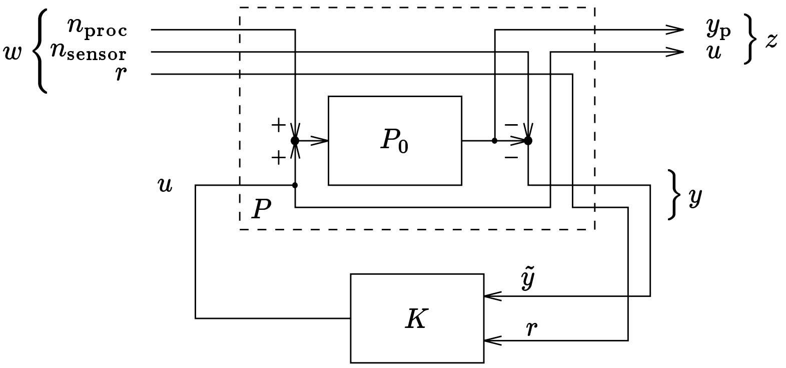


Figure 2.14 The closed-loop 2-DOF system.

(the subscripts remind the reader of the interpretation), then the closed-loop transfer matrix H is

$$H = \begin{bmatrix} \dfrac{P_0}{1+P_0K_{\tilde{y}}} & -\dfrac{P_0K_{\tilde{y}}}{1+P_0K_{\tilde{y}}} & \dfrac{P_0K_r}{1+P_0K_{\tilde{y}}} \\ -\dfrac{P_0K_{\tilde{y}}}{1+P_0K_{\tilde{y}}} & -\dfrac{K_{\tilde{y}}}{1+P_0K_{\tilde{y}}} & \dfrac{K_r}{1+P_0K_{\tilde{y}}} \end{bmatrix}. \qquad (2.16)$$

Just as in the 1-DOF control system, the closed-loop transfer matrix H consists of the closed-loop transfer matrix of the classical regulator described in section 2.3.1 with a third column appended. The interpretations of the elements of H are the same as in the 1-DOF control system, and hence we can compare the two control systems. If $K_{\tilde{y}} = K_r$, then the closed-loop transfer matrix for the 2-DOF control system is the same as the closed-loop transfer matrix for the 1-DOF control system. Thus, the 2-DOF control system is more general than the 1-DOF control system.

2.3.4 2-DOF Control System with Multiple Actuators and Sensors

The standard examples described in the previous sections can be extended to plants with multiple actuators and multiple sensors (MAMS), by interpreting the various signals as vector signals and expressing the transfer functions as the appropriate transfer matrices. As an example, we describe the MAMS 2-DOF control system.

The block diagrams shown in figures 2.13 and 2.14 can describe the MAMS 2-DOF control system, provided we interpret all the (previously scalar) signals as vector signals. The signals w, u, z, and y are given by the expressions (2.12) and (2.13); the plant transfer matrix is essentially the same as (2.14), with the ones and zeros replaced by the appropriate identity and zero matrices:

$$P = \left[\begin{array}{ccc|c} P_0 & 0_{n_{\tilde{y}} \times n_{\tilde{y}}} & 0_{n_{\tilde{y}} \times n_r} & P_0 \\ 0_{n_u \times n_u} & 0_{n_u \times n_{\tilde{y}}} & 0_{n_u \times n_r} & I_{n_u \times n_u} \\ \hline -P_0 & -I_{n_{\tilde{y}} \times n_{\tilde{y}}} & 0_{n_{\tilde{y}} \times n_r} & -P_0 \\ 0_{n_r \times n_u} & 0_{n_r \times n_{\tilde{y}}} & I_{n_r \times n_r} & 0_{n_r \times n_u} \end{array} \right] \tag{2.17}$$

(the subscripts indicate the sizes; n_r is the size of the reference signal r, and need not be the same as $n_{\tilde{y}}$, the size of the y_{p}). In the sequel we will not write out the sizes so explicitly.

We partition the $n_u \times n_y$ transfer matrix K as in (2.15), such that $K_{\tilde{y}}$ is an $n_u \times n_{\tilde{y}}$ transfer matrix, and K_r is an $n_u \times n_r$ transfer matrix (note that $n_y = n_{\tilde{y}} + n_r$). The closed-loop transfer matrix H is a generalization of (2.16):

$$H = \left[\begin{array}{ccc} SP_0 & -P_0 K_{\tilde{y}} S & SP_0 K_r \\ -K_{\tilde{y}} SP_0 & -K_{\tilde{y}} S & \tilde{S} K_r \end{array} \right], \tag{2.18}$$

where

$$S = (I + P_0 K_{\tilde{y}})^{-1},$$
$$\tilde{S} = (I + K_{\tilde{y}} P_0)^{-1}.$$

S is called the sensitivity matrix of the MAMS 2-DOF control system. To distinguish it from $\tilde{S}$, S is sometimes called the *output-referred sensitivity matrix*, for reasons that will become clear in chapter 9. The I/O transfer matrix of the MAMS 2-DOF system is

$$T = (I + P_0 K_{\tilde{y}})^{-1} P_0 K_r = P_0 (I + K_{\tilde{y}} P_0)^{-1} K_r,$$

which is the closed-loop transfer matrix from the reference input r to the system output signal y_{p}.

2.4 A Standard Numerical Example

In this section we describe a particular plant and several controllers that will be used in examples throughout this book. We consider the 1-DOF control system, described in section 2.3.2, with

$$P_0 = P_0^{\mathrm{std}} \stackrel{\Delta}{=} \frac{1}{s^2} \frac{10 - s}{10 + s}.$$

P_0^{std} consists of a double integrator with some excess phase from the allpass term $(10 - s)/(10 + s)$, which approximates a 0.2 second delay at low frequencies.

In various future examples we will encounter the controllers

$$\begin{aligned}
K^{(\mathrm{a})}(s) &\triangleq \frac{44.14s^2+107.3s+39}{s^3+10s^2+55.25s+78.14},\\
K^{(\mathrm{b})}(s) &\triangleq \frac{219.6s^2+1973.95s+724.5}{s^3+19.15s^2+105.83s+965.95},\\
K^{(\mathrm{c})}(s) &\triangleq \frac{95.05s^2+975s+244.95}{s^3+23.91s^2+185.87s+824.84},\\
K^{(\mathrm{d})}(s) &\triangleq \frac{35.28s^2+360.52s+77.46}{s^3+19.38s^2+132.8s+481.0}.
\end{aligned}$$

The corresponding closed-loop transfer matrices, which we denote $H^{(\mathrm{a})}$, $H^{(\mathrm{b})}$, $H^{(\mathrm{c})}$, and $H^{(\mathrm{d})}$ respectively, can be computed from (2.11). The closed-loop systems that result from using the controllers $K^{(\mathrm{a})}$, $K^{(\mathrm{b})}$, $K^{(\mathrm{c})}$, and $K^{(\mathrm{d})}$ can be compared by examining the 2×3 transfer matrices $H^{(\mathrm{a})}$, $H^{(\mathrm{b})}$, $H^{(\mathrm{c})}$, and $H^{(\mathrm{d})}$. For example, figure 2.15 shows $|H_{12}^{(\mathrm{a})}(j\omega)|$, $|H_{12}^{(\mathrm{b})}(j\omega)|$, $|H_{12}^{(\mathrm{c})}(j\omega)|$, and $|H_{12}^{(\mathrm{d})}(j\omega)|$, *i.e.*, the magnitudes of the closed-loop transfer functions from n_{sensor} to y_{p}. From this figure, we can conclude that a high frequency sensor noise will have the greatest effect on y_{p} in the closed-loop system with the controller $K^{(\mathrm{b})}$, and the least effect in the closed-loop system with controller $K^{(\mathrm{d})}$.

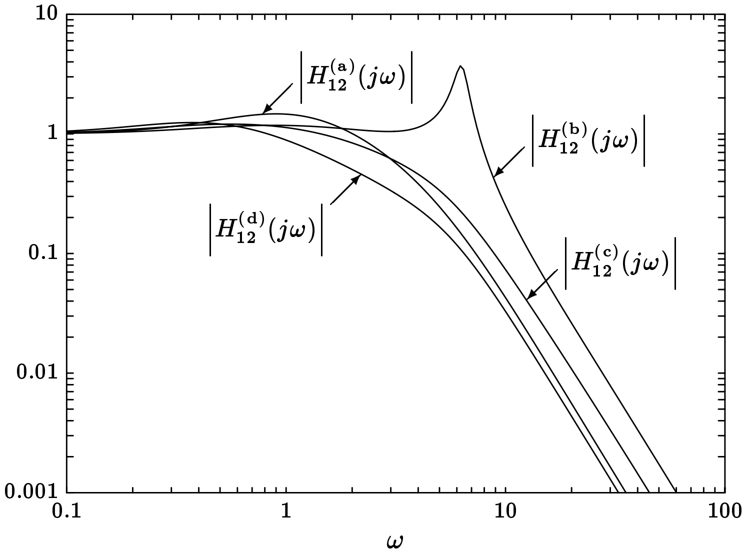


Figure 2.15 Magnitudes of the closed-loop transfer functions from n_{sensor} to y_{p} for the four different closed-loop transfer matrices.

2.5 A State-Space Formulation

We will occasionally refer to state-space realizations of the plant, controller, and closed-loop system. The general *multiple-actuator, multiple-sensor* (MAMS) plant P takes two vectors of input signals (w and u) and produces two vectors of output signals (z and y); a state-space realization of the plant is thus:

$$\dot{x} = A_P x + B_w w + B_u u \tag{2.19}$$

$$z = C_z x + D_{zw} w + D_{zu} u \tag{2.20}$$

$$y = C_y x + D_{yw} w + D_{yu} u, \tag{2.21}$$

(with $x(0) = 0$), so that

$$P(s) = \left[\begin{array}{c|c} P_{zw}(s) & P_{zu}(s) \\ \hline P_{yw}(s) & P_{yu}(s) \end{array}\right] = C_P(sI - A_P)^{-1}B_P + D_P,$$

where

$$B_P = \begin{bmatrix} B_w & B_u \end{bmatrix}$$

$$C_P = \begin{bmatrix} C_z \\ C_y \end{bmatrix}$$

$$D_P = \begin{bmatrix} D_{zw} & D_{zu} \\ D_{yw} & D_{yu} \end{bmatrix}.$$

Many plant models encountered in practice have the property that $D_{yu} = 0$, or equivalently, $P_{yu}(\infty) = 0$; such plants are called *strictly proper.* For strictly proper plants, the state-space formulas that we will encounter are greatly simplified, so we make the following assumption:

Assumption 2.4: *The plant is strictly proper:* $D_{yu} = 0$ *(i.e.,* $P_{yu}(\infty) = 0$*).*

This assumption is not substantial; we make it for convenience and aesthetic reasons. The more complicated state-space formulas for the case $D_{yu} \neq 0$ can be found in the Notes and References that we cite at the end of each chapter.

Suppose that our controller has state-space realization

$$\dot{x}_K = A_K x_K + B_K y \tag{2.22}$$

$$u = C_K x_K + D_K y, \tag{2.23}$$

so that

$$K(s) = C_K(sI - A_K)^{-1}B_K + D_K.$$

A state-space realization of the closed-loop system can be found by eliminating u and y from (2.19–2.21) and (2.22–2.23):

$$\dot{x} = (A_P + B_u D_K C_y)x + B_u C_K x_K + (B_w + B_u D_K D_{yw})w \tag{2.24}$$

$$\dot{x}_K = B_K C_y x + A_K x_K + B_K D_{yw} w \tag{2.25}$$

$$z = (C_z + D_{zu} D_K C_y)x + D_{zu} C_K x_K + (D_{zw} + D_{zu} D_K D_{yw})w \tag{2.26}$$

so that

$$H(s) = C_H(sI - A_H)^{-1}B_H + D_H, \tag{2.27}$$

where

$$\begin{aligned}
A_H &= \begin{bmatrix} A_P + B_u D_K C_y & B_u C_K \\ B_K C_y & A_K \end{bmatrix} \\
B_H &= \begin{bmatrix} B_w + B_u D_K D_{yw} \\ B_K D_{yw} \end{bmatrix} \\
C_H &= \begin{bmatrix} C_z + D_{zu} D_K C_y & D_{zu} C_K \end{bmatrix} \\
D_H &= D_{zw} + D_{zu} D_K D_{yw}.
\end{aligned}$$

The state-space realization of the closed-loop system is shown in figure 2.16.

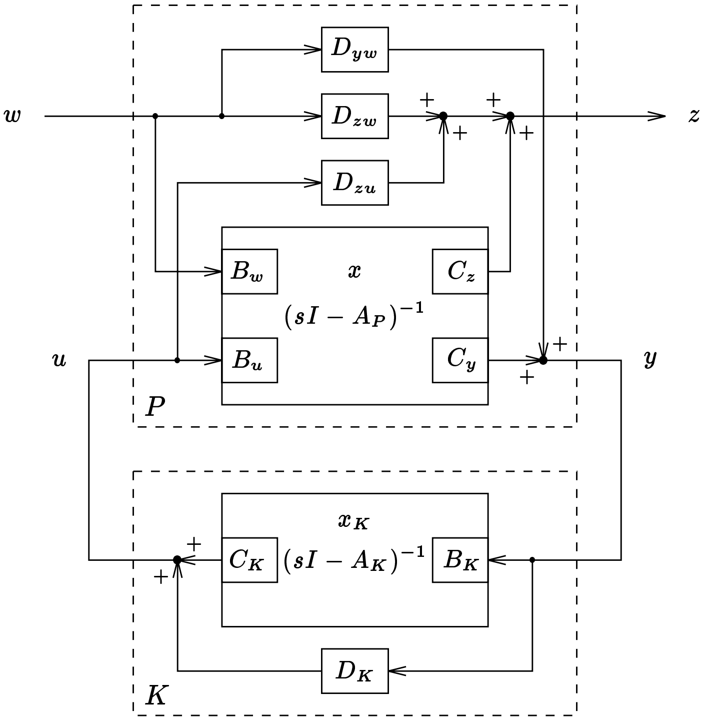


Figure 2.16 State-space realizations of the plant and controller connected to form a state-space realization of the closed-loop system.

Notes and References

Standard Descriptions and Nomenclature for Control Systems

In 1954, the Feedback Control Systems Committee of the AIEE approved an AIEE Proposed Standard of Terminology for Feedback Control Systems, which included a standard for the block diagram (see [AIE51] and [NGK57]).

Two-Input, Two-Output Plant Description

The description of a plant that explicitly shows the exogenous input and the regulated variables is now common, and the symbols w, u, z, and y are becoming standard. They appear, for example, in Francis [FRA87]. Nett [NET86] explicitly discusses the command inputs to the controller and the diagnostic outputs from the controller, which he denotes w' and z', respectively (see figures 2.3 and 2.4).

An example of a diagnostic output is the output prediction error of an observer; a substantial increase in the size of this signal may indicate that the plant is no longer a good model of the system to be controlled, *i.e.*, some system failure has occurred. This idea is treated in the survey by Isermann [ISE84].

Nonlinear Controllers Based on LTI Controllers

Several methods for designing nonlinear controllers for nonlinear plants are based on the design of LTI controllers, *e.g.*,

- *Linear variational method.* An LTI controller is designed for an approximate LTI model of a nonlinear plant (near some equilibrium point). We will see a specific example of this in chapter 10.
- *Gain scheduling.* For a family of equilibrium points, approximate LTI models of a nonlinear plant are developed, and for each of these LTI models an LTI controller is designed. On the basis of the sensed signals, y, an estimate is made of the equilibrium point that is "closest" to the current plant state, and the corresponding LTI controller is "switched in". The overall controller thus consists of a family of LTI controllers along with a selection algorithm; this overall controller is nonlinear.

  In cases where a sensor can readily or directly measure this "closest" equilibrium point, and the family of LTI controllers differ only in some parameter values or "gains", the resulting controller is called *gain scheduled.* For example, many aircraft controllers are gain scheduled; the equilibrium points might be parametrized by altitude and airspeed.
- *Adaptive control.* An *adaptive controller* uses an identification procedure based on the signals y and u to estimate the "closest" equilibrium point, or other system parameters. This estimate is often used for gain scheduling. See for example [AW89, SB88, GS84, ABJ86].
- *Feedback linearization.* In many cases, it is possible to construct a preliminary nonlinear feedback that makes the plant, with the preliminary feedback loop closed, linear and time-invariant, and thus amenable to the methods of this book. This idea is studied in the field of *geometric control theory*; see for example [HSM83]. A clear exposition can be found in chapters 4 and 5 of the book by Isidori [ISI89], which also has many references.

Control System Architectures

A description of different control architectures can be found in, *e.g.*, Lunze [LUN89, P31], Anderson and Moore [AM90, P212], and Maciejowski [MAC89, §1.3]. Discussions of control architectures appear in the articles [DL85] and [NET86].

We noted that the 2-DOF control system is more general than the 1-DOF control system, and yet the 1-DOF control system architecture is widely chosen. In some cases, the actual sensor used is a *differential sensor*, which directly senses the difference between the reference and the system output, so that the 1-DOF plant accurately reflects signals accessible to the controller. But in many cases, the system output signal and the reference input signal are separately sensed, as in the 2-DOF controller, and a *design decision* is made to have the controller process only their difference.

One possible reason for this is that the block diagram in figure 2.11 has been widely used to describe the general *feedback paradigm.* In *The Origins of Feedback Control* [MAY70], for example, Mayr *defines* feedback control as forming u from the error signal e (*i.e.*, the 1-DOF control system). It would be better to refer to the general principle behind the 1-DOF control system as the *error-nulling paradigm*, leaving *feedback paradigm* to refer to the more general scheme depicted in figure 2.2.

The Standard Example Plant

The idea of using a double integrator plant with some excess phase as a simple but realistic typical plant with which to explore control design tradeoffs is taken from a study presented by Stein in [FBS87]. A discretized version of our standard example plant is considered in Barratt and Boyd [BB89].

State-Space Descriptions

Comprehensive texts include Kailath [KAI80], Chen [CHE84]; an earlier text is Zadeh and Desoer [ZD63]. The appendices in the book by Anderson and Moore [AM90] contain all of the material about state-space LTI systems that is needed for this book.

Chapter 3

Controller Design Specifications and Approaches

In this chapter we develop a unified framework for describing the goals of controller design in terms of families of boolean *design specifications.* We show how various approaches, *e.g.*, multicriterion optimization or classical optimization, can be described in this framework.

Just as there are many different architectures or configurations for control systems, there are many different general approaches to expressing the *design goals* and *objectives* for controller design. One example is the *optimal controller paradigm*: the goal is to determine a controller that minimizes a single cost function or objective. In another approach, *multicriterion optimization*, several different cost functions are specified and the goal is to identify controllers that perform mutually well on these goals. The purpose of this chapter is to develop a unified framework for describing design specifications, and to explain how these various approaches can be described using this framework.

Throughout this chapter, we assume that a fixed plant is under consideration: the signals w, u, z, and y have been defined, and the plant transfer matrix P has been determined. As described in chapter 2, the definitions of the exogenous input w and the regulated variables z should contain enough signals that we can express every goal of the controller design in terms of H, the closed-loop transfer matrix from w to z.

3.1 Design Specifications

We begin by defining the basic or atomic notion of a *design specification*:

Definition 3.1: *A* design specification $\mathcal{D}$ *is a boolean function or test on the closed-loop transfer matrix* H*.*

Thus, for each candidate closed-loop transfer matrix H, a design specification $\mathcal{D}$ is either *satisfied* or *not satisfied*: design specifications are simple tests, with a "pass" or "fail" outcome. A design specification is a *predicate* on transfer matrices, *i.e.*, a function that takes an $n_z \times n_w$ transfer matrix as argument and returns a boolean result:

$$\mathcal{D} : \mathcal{H} \longrightarrow \{\text{PASS, FAIL}\},$$

where $\mathcal{H}$ denotes the set of all $n_z \times n_w$ transfer matrices.

It may seem strange to the reader that we express design specifications in terms of the closed-loop transfer matrix H instead of the controller transfer matrix K, since we really design the controller transfer matrix K, and not the resulting closed-loop transfer matrix H. Of course, to each candidate controller K there corresponds the resulting closed-loop transfer matrix H (given by the formula (2.7) in definition 2.3), which either passes or fails a given design specification; hence, we can think of a design specification as inducing a boolean test on the transfer matrices of candidate controllers. In fact, we will sometimes abuse notation by saying "the controller K satisfies $\mathcal{D}$", meaning that the corresponding closed-loop transfer matrix H satisfies $\mathcal{D}$.

Such a PASS/FAIL test on candidate controllers would be *logically* equivalent to the design specification, which is a PASS/FAIL test on closed-loop transfer matrices, and perhaps more natural. We will see in chapter 6, however, that there are important *geometric* advantages to expressing design specifications in terms of the closed-loop transfer matrix H and not directly in terms of the controller K.

3.1.1 Some Examples

Some possible design specifications for the standard example described in section 2.4 are:

- *Maximum step response overshoot.* $\mathcal{D}_{\rm os}$ will denote the design specification "the step response overshoot from the reference signal to $y_{\rm p}$ is less than 10%".

  Let us express $\mathcal{D}_{\rm os}$ more explicitly in terms of H, the 2×3 closed-loop transfer matrix. The reference signal r is the third exogenous input (w_3), and $y_{\rm p}$ is the first regulated variable (z_1), so H_{13} is the closed-loop transfer function from the reference signal to $y_{\rm p}$, and its step response is given by

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{H_{13}(j\omega)}{j\omega} e^{j\omega t}\, d\omega,$$

for $t \geq 0$. Thus, $\mathcal{D}_{\rm os}$ can be expressed as

$$\mathcal{D}_{\rm os} \;:\; \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{H_{13}(j\omega)}{j\omega} e^{j\omega t}\, d\omega \leq 1.1 \quad \text{for all } t \geq 0.$$

- *Maximum RMS actuator effort.* $\mathcal{D}_{\rm act\_eff}$ will denote the design specification: "the RMS value of the actuator signal due to the sensor and process noises is less than 0.1".

  We shall see in chapters 5 and 8 that $\mathcal{D}_{\rm act\_eff}$ can be expressed as

  $$\mathcal{D}_{\rm act\_eff} \;:\; \frac{1}{2\pi}\int_{-\infty}^{\infty} \left(|H_{21}(j\omega)|^2 S_{\rm proc}(\omega) + |H_{22}(j\omega)|^2 S_{\rm sensor}(\omega)\right) d\omega \leq 0.1^2,$$

  where $S_{\rm proc}$ and $S_{\rm sensor}$ are the power spectral densities of the noises $n_{\rm proc}$ and $n_{\rm sensor}$, respectively.

- *Closed-loop stability.* $\mathcal{D}_{\rm stable}$ will denote the design specification: "the closed-loop transfer matrix H is achieved by a controller that stabilizes the plant". (The precise meaning of $\mathcal{D}_{\rm stable}$ can be found in chapter 7.)

A detailed understanding of these design specifications is not necessary in this chapter; the important point here is that given any 2×3 transfer matrix H, each of the design specifications $\mathcal{D}_{\rm os}$, $\mathcal{D}_{\rm act\_eff}$, and $\mathcal{D}_{\rm stable}$ is either *true* or *false*. This is independent of our knowing *how* to verify whether a design specification holds or not for a given transfer matrix. For example, the reader may not yet know of any explicit procedure for determining whether a transfer matrix H satisfies $\mathcal{D}_{\rm stable}$; nevertheless it is a valid design specification.

These design specifications should be contrasted with the associated informal design goals "the step response from the reference signal to $y_{\rm p}$ should not overshoot too much" and "the sensor and process noises should not cause u to be too large". These are *not* design specifications, even though these qualitative goals may better capture the designer's intentions than the precise design specifications $\mathcal{D}_{\rm os}$ and $\mathcal{D}_{\rm act\_eff}$. Later in this chapter we will discuss how these informal design goals can be better expressed using families of design specifications.

3.1.2 Comparing and Ordering Design Specifications

In some cases, design specifications can be compared. We say that design specification $\mathcal{D}_1$ is *tighter* or *stronger* than $\mathcal{D}_2$ (or, $\mathcal{D}_2$ is *looser* or *weaker* than $\mathcal{D}_1$) if all transfer matrices that satisfy $\mathcal{D}_1$ also satisfy $\mathcal{D}_2$. We will say that $\mathcal{D}_1$ is *strictly tighter* or *strictly stronger* than $\mathcal{D}_2$ if it is tighter and in addition there is a transfer matrix that satisfies $\mathcal{D}_2$ but not $\mathcal{D}_1$.

An obvious but extremely important fact is that given two design specifications, it is not always possible to compare them: it is possible that neither is a stronger specification. The ordering of design specifications by strength is a *partial ordering*, not a *linear* or *total ordering.* We can draw a *directed graph* of the relations between different specifications by connecting strictly weaker specifications to stronger ones by arrows, and deleting the arrows that follow from transitivity (*i.e.*, $\mathcal{D}_1$ is stronger than $\mathcal{D}_2$ which is stronger than $\mathcal{D}_3$ implies $\mathcal{D}_1$ is stronger than $\mathcal{D}_3$). An example of a linear ordering is shown in figure 3.1(a): every pair of specifications can be compared since there is a directed path connecting any two specifications. By comparison, a partial but not linear ordering of design specifications is shown in figure 3.1(b). Specification $\mathcal{D}_{\rm D}$ is tighter than $\mathcal{D}_{\rm C}$, which in turn is tighter than both $\mathcal{D}_{\rm A}$ and $\mathcal{D}_{\rm B}$. Similarly, specification $\mathcal{D}_{\rm G}$ is tighter than $\mathcal{D}_{\rm E}$, which is itself tighter than $\mathcal{D}_{\rm C}$. However, specifications $\mathcal{D}_{\rm E}$ and $\mathcal{D}_{\rm D}$ cannot be compared; nor can the design specifications $\mathcal{D}_{\rm G}$ and $\mathcal{D}_{\rm D}$.

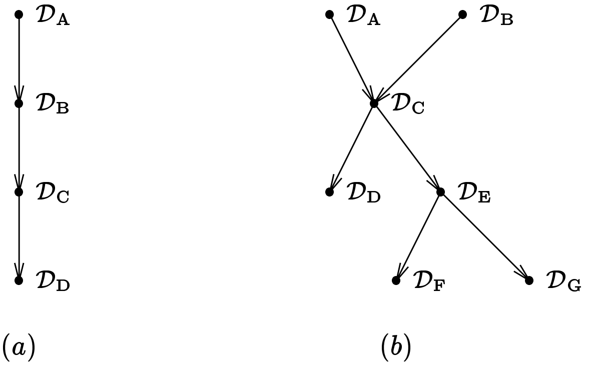


Figure 3.1 In these directed graphs each node represents a specification. Arrows connect a weaker specification to a tighter one. The graph in (a) shows a linear ordering: all specifications can be compared. The graph in (b) shows a partial, but not linear, ordering: some specifications can be compared (*e.g.*, $\mathcal{D}_{\rm D}$ is tighter than $\mathcal{D}_{\rm A}$), but others cannot (*e.g.*, $\mathcal{D}_{\rm F}$ is neither weaker nor tighter than $\mathcal{D}_{\rm D}$).

We may form new design specifications from other design specifications using any boolean operation, for example *conjunction*, meaning joint satisfaction:

$$\mathcal{D}_1 \wedge \mathcal{D}_2 \quad : \quad H \text{ satisfies } \mathcal{D}_1 \text{ and } \mathcal{D}_2.$$

The new design specification $\mathcal{D}_1 \wedge \mathcal{D}_2$ is tighter than both $\mathcal{D}_1$ and $\mathcal{D}_2$.

We say that a design specification is *infeasible* or *inconsistent* if it is not satisfied by any transfer matrix; it is *feasible* or *consistent* if it is satisfied by at least one transfer matrix. We say that the set of design specifications $\{\mathcal{D}_1, \ldots, \mathcal{D}_L\}$ is *jointly infeasible* or *jointly inconsistent* if the conjunction $\mathcal{D}_1 \wedge \cdots \wedge \mathcal{D}_L$ is infeasible; if the conjunction $\mathcal{D}_1 \wedge \cdots \wedge \mathcal{D}_L$ is feasible, then we say that the set of design specifications $\{\mathcal{D}_1, \ldots, \mathcal{D}_L\}$ is *jointly feasible* or *achievable.*

3.2 The Feasibility Problem

Given a specific set of design specifications, we can pose the feasibility problem of determining whether all of our design specifications can be simultaneously satisfied:

Definition 3.2: Feasibility controller design problem: *given a set of design specifications* $\{\mathcal{D}_1, \ldots, \mathcal{D}_L\}$*, determine whether it is jointly feasible. If so, find a closed-loop transfer matrix* H *that meets the design specifications* $\mathcal{D}_1, \ldots, \mathcal{D}_L$.

Like design specifications, the feasibility problem has a boolean outcome. If this outcome is negative, we say the set of design specifications $\{\mathcal{D}_1, \ldots, \mathcal{D}_L\}$ is *too tight*, *infeasible*, or *unachievable*; if the outcome of the feasibility problem is positive, we say that $\{\mathcal{D}_1, \ldots, \mathcal{D}_L\}$ is *achievable* or *feasible.* It is only for ease of interpretation that we pose the feasibility problem in terms of sets of design specifications, since it is equivalent to feasibility of the single design specification $\mathcal{D}_1 \wedge \cdots \wedge \mathcal{D}_L$.

The feasibility problem is not meant by itself to capture the whole controller design problem; rather it is meant to be a *basic* or *atomic problem* that we can use to describe other, more subtle, formulations of the controller design problem. In the rest of this chapter we will describe various design approaches in terms of families of feasibility problems. This is exactly like our definition of the standard plant, which is meant to be a standard form in which to describe the many possible architectures for controllers. The motivation is the same—to provide the means to sensibly compare apparently different formulations.

3.3 Families of Design Specifications

A single fixed set of design specifications usually does not adequately capture the notion of "suitable" or "satisfactory" system performance, which more often involves tradeoffs among competing desirable qualities, and specifications with varying degrees of *hardness.* Hardness is a quality that describes how firmly the designer insists on a specification, or how flexible the designer is in accepting violations of a design specification. Thus, the solution of a single, fixed, feasibility problem may be of limited utility. These vaguer notions of satisfactory performance can be modeled by considering *families* of related feasibility problems. The designer can then choose among those sets of design specifications that are achievable.

To motivate this idea, we consider again the specifications described in section 3.1.1. Suppose first that the solution to the feasibility problem with the set of specifications $\{\mathcal{D}_{\rm os},\ \mathcal{D}_{\rm act\_eff},\ \mathcal{D}_{\rm stable}\}$ is affirmative, and that H simultaneously satisfies $\mathcal{D}_{\rm os}$, $\mathcal{D}_{\rm act\_eff}$, and $\mathcal{D}_{\rm stable}$. If the set of specifications $\{\mathcal{D}_{\rm os},\ \mathcal{D}_{\rm act\_eff},\ \mathcal{D}_{\rm stable}\}$ adequately captures our notion of a satisfactory design, then we are done. But the designer may wonder how much smaller the overshoot and actuator effort can be made than the original limits of 10% and 0.1, respectively. In other words, the designer will often want to know not just that the set of specifications $\{\mathcal{D}_{\rm os},\ \mathcal{D}_{\rm act\_eff},\ \mathcal{D}_{\rm stable}\}$

is achievable, but in addition how much these specifications could be tightened while remaining achievable.

A similar situation occurs if the solution to the feasibility problem with the design specifications $\{\mathcal{D}_{\mathrm{os}},\ \mathcal{D}_{\mathrm{act\_eff}},\ \mathcal{D}_{\mathrm{stable}}\}$ is negative. In this case the designer might like to know how "close" this set of design specifications is to achievable. For example, how much larger would the limit on overshoot have to be made for this set of design specifications to be achievable?

Questions of this sort can be answered by considering *families of design specifications*, which are often indexed by numbers. In our example above, we could consider a *family* of overshoot specifications indexed, or parametrized, by the allowable overshoot. For $a_1 \geq 0$ we define

$$\mathcal{D}_{\mathrm{os}}^{(a_1)} \ : \ \text{overshoot of step response from } r \text{ to } y_{\mathrm{p}} \ \leq a_1\%.$$

Note that $\mathcal{D}_{\mathrm{os}}^{(a_1)}$ is a *different specification* for each value of a_1. Similarly, a family of actuator effort specifications for $a_2 \geq 0$ is

$$\mathcal{D}_{\mathrm{act\_eff}}^{(a_2)} \ : \ \text{RMS deviation of } u \text{ due to the sensor and actuator noises } \ \leq a_2.$$

Thus for each $a_1 \geq 0$ and $a_2 \geq 0$ we have a different set of design specifications,

$$\mathcal{D}^{(a_1,a_2)} \triangleq \mathcal{D}_{\mathrm{os}}^{(a_1)} \wedge \mathcal{D}_{\mathrm{act\_eff}}^{(a_2)} \wedge \mathcal{D}_{\mathrm{stable}}.$$

Suppose we solve the feasibility problem with design specifications $\mathcal{D}^{(a_1,a_2)}$ for each a_1 and a_2, shading the area in the (a_1, a_2) plane corresponding to feasible $\mathcal{D}^{(a_1,a_2)}$, as shown in figure 3.2. We call the shaded region in figure 3.2, with some abuse of notation, a *plot of achievable specifications in performance space.*

From this plot we can better answer the vague questions posed above. Our original set of design specifications corresponds to $a_1 = 10\%$ and $a_2 = 0.1$, shown as $\mathcal{D}_{\mathrm{X}}$ in figure 3.2. From figure 3.2 we may conclude, for example, that we may tighten the overshoot specification to $a_1 = 6\%$ and still have an achievable set of design specifications ($\mathcal{D}_{\mathrm{Y}}$ in figure 3.2), but if we further tighten the overshoot specification to $a_1 = 4\%$, we have a set of specifications ($\mathcal{D}_{\mathrm{Z}}$) that is too tight. Alternatively, we could tighten the actuator effort specification to $a_2 = 0.06$ and also have an achievable set of design specifications ($\mathcal{D}_{\mathrm{W}}$).

By considering the feasibility problem for *families* of design specifications, we have considerably more information than if we only consider one fixed set of specifications, and therefore are much better able to decide what a satisfactory design is. We will return to a discussion of this general idea, and this particular example, after we describe a general and common method for parametrizing design specifications.

3.4 Functional Inequality Specifications

The parametrized families $\mathcal{D}_{\mathrm{os}}^{(a_1)}$ and $\mathcal{D}_{\mathrm{act\_eff}}^{(a_2)}$ described above have the same general form. Each of these families involves a particular quantity (overshoot and actuator

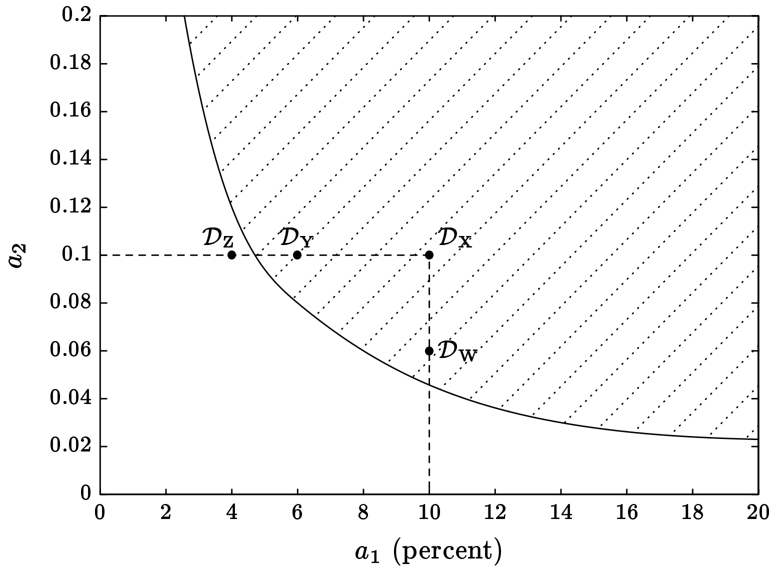


Figure 3.2 A plot of achievable specifications in performance space: the region where the set of design specifications $\{\mathcal{D}_{\rm os}^{(a_1)}, \mathcal{D}_{\rm act\_eff}^{(a_2)}, \mathcal{D}_{\rm stable}\}$ is achievable is shaded. The specification $\mathcal{D}_{\rm X}$ corresponds to $a_1 = 10\%$ and $a_2 = 0.1$, the original design specifications. The specification $\mathcal{D}_{\rm Y}$ represents a tightening of the overshoot specification over $\mathcal{D}_{\rm X}$ to $a_1 = 6\%$ and is an achievable specification, whereas the further tightening of the overshoot specification to $a_1 = 4\%$ ($\mathcal{D}_{\rm Z}$) is *not* achievable.

effort, respectively) for which a smaller value is "better", or at least, a tighter specification. The parameter (a_1 and a_2, respectively) is simply a *limit* on the associated quantity.

This notion of "quantity" we make more precise as a *functional* on transfer matrices. A functional is just a function that assigns to each $n_z \times n_w$ transfer matrix H a real number (or possibly ∞).

Definition 3.3: *A* functional ϕ *(on transfer matrices) is a function*

$$\phi \;:\; \mathcal{H} \rightarrow \mathbf{R} \cup \{\infty\}.$$

In contrast to a design specification, which a transfer matrix either passes or fails, a functional can be thought of as *measuring* some property or quality of the transfer matrix, without judgment. Many functionals can be interpreted as measuring the *size* of a transfer matrix or some of its entries (the subject of chapters 4 and 5). Allowing functionals to return the value $+\infty$ for some transfer matrices will be convenient for us.

Two functionals related to our example above are the overshoot,

$$\phi_{\rm os}(H) \triangleq \sup_{t\geq 0}\left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{H_{13}(j\omega)}{j\omega}e^{j\omega t}\,d\omega - 1\right),$$

and the RMS response at u due to the sensor and actuator noises,

$$\phi_{\rm act\_eff}(H) \triangleq \left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\left(|H_{21}(j\omega)|^2 S_{\rm proc}(\omega) + |H_{22}(j\omega)|^2 S_{\rm sensor}(\omega)\right)d\omega\right)^{1/2}.$$

The families of design specifications $\mathcal{D}_{\rm os}^{(a_1)}$ and $\mathcal{D}_{\rm act\_eff}^{(a_2)}$ can be expressed as the *functional inequality specifications*:

$$\mathcal{D}_{\rm os}^{(a_1)} \quad : \quad \phi_{\rm os}(H) \leq a_1,$$

and

$$\mathcal{D}_{\rm act\_eff}^{(a_2)} \quad : \quad \phi_{\rm act\_eff}(H) \leq a_2.$$

More generally, we have:

Definition 3.4: *A* functional inequality specification *is a specification of the form*

$$\mathcal{D}_{\phi}^{(a)} \quad : \quad \phi(H) \leq a, \tag{3.1}$$

where ϕ is a functional on transfer matrices and a is a constant.

Thus a can be interpreted as the maximum allowable value of the functional ϕ; by varying a, the expression (3.1) sweeps out a family of design specifications.

The family of specifications given by a functional inequality (3.1) is linearly ordered: $\mathcal{D}_{\phi}^{(a)}$ is a stronger specification than $\mathcal{D}_{\phi}^{(b)}$ if $a \leq b$. A functional can also allow us to make *quantitative comparisons* of the specifications $\mathcal{D}_{\phi}^{(a)}$ and $\mathcal{D}_{\phi}^{(b)}$. For example $\mathcal{D}_{\rm act\_eff}^{(0.1)}$ is not merely a tighter specification than $\mathcal{D}_{\rm act\_eff}^{(0.2)}$; we can say that it is twice as tight, at least as measured by the functional $\phi_{\rm act\_eff}$.

In the next two sections we describe two common methods for capturing the notion of a "suitable" design using families of feasibility problems.

3.5 Multicriterion Optimization

In multicriterion optimization, we have a hard constraint $\mathcal{D}_{\rm hard}$ and *objective functionals* or *criteria* $\phi_1, \ldots, \phi_L$. Each objective represents a *soft* goal in the design: if the closed-loop transfer matrix H satisfies $\mathcal{D}_{\rm hard}$, it is generally desirable to have each $\phi_i(H)$ small, although we have not (yet) set any priorities among the objective functionals.

We consider the family of design specifications, indexed by the parameters $a_1, \ldots, a_L$, given by

$$\mathcal{D}^{(a_1,\ldots,a_L)} \quad : \quad \mathcal{D}_{\text{hard}} \wedge \mathcal{D}_{\phi_1}^{(a_1)} \wedge \cdots \wedge \mathcal{D}_{\phi_L}^{(a_L)}. \tag{3.2}$$

$\mathcal{D}_{\text{hard}}$ is the conjunction of our hard constraints—performance specifications about which we are inflexible. Every specification we consider, $\mathcal{D}^{(a_1,\ldots,a_L)}$, is stronger than $\mathcal{D}_{\text{hard}}$. The remaining specifications are functional inequalities for the criteria. The basic goal of multicriterion optimization is to identify specifications $\mathcal{D}^{(a_1,\ldots,a_L)}$ that are on the boundary between achievable specifications and unachievable specifications:

Definition 3.5: *The specification $\mathcal{D}^{(a_1,\ldots,a_L)}$ is* Pareto optimal *or* noninferior *if the specification $\mathcal{D}^{(\tilde{a}_1,\ldots,\tilde{a}_L)}$ is achievable whenever $\tilde{a}_1 > a_1, \ldots, \tilde{a}_L > a_L$, and is unachievable whenever $\tilde{a}_1 < a_1, \ldots, \tilde{a}_L < a_L$. A closed-loop transfer matrix that satisfies a Pareto optimal specification is called a Pareto optimal transfer matrix or design.*

This is illustrated in figure 3.3 for our particular example. The points on the boundary between the achievable and unachievable specifications, shown with a dashed line in figure 3.3, are precisely the Pareto optimal specifications.

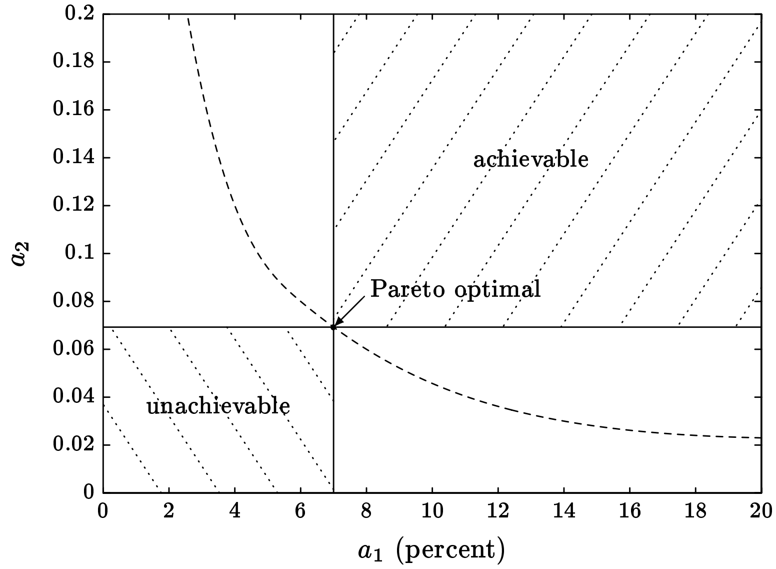


Figure 3.3 A specification is Pareto optimal if smaller limits in the functional inequalities yield unachievable specifications and larger limits in the functional inequalities yield achievable specifications. Such specifications and on the boundary between achievable and unachievable specifications.

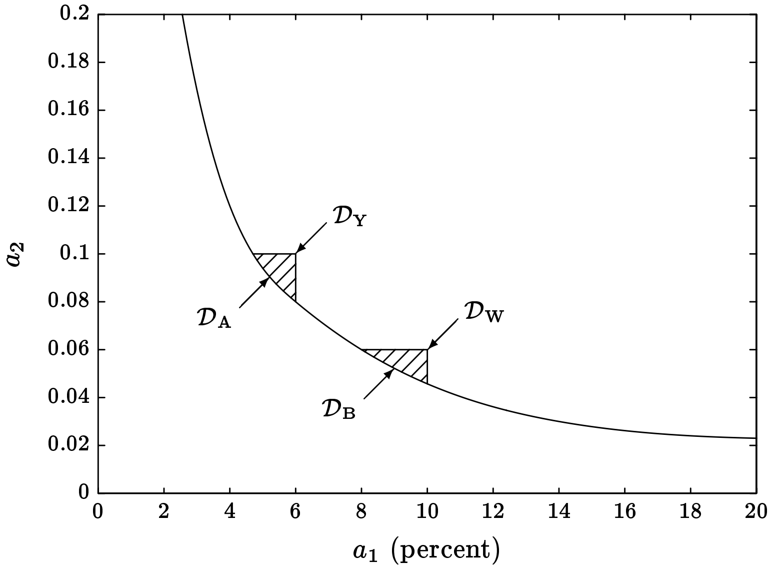


Figure 3.4 The region of achievable specifications that are tighter than the specification marked $\mathcal{D}_Y$ is shaded, and similarly for $\mathcal{D}_W$. The specifications $\mathcal{D}_A$ and $\mathcal{D}_B$ are Pareto optimal.

Let us illustrate this concept further. Figure 3.2 is redrawn in figure 3.4 with the specifications $\mathcal{D}_Y$ and $\mathcal{D}_W$ shown, along with two new specifications, $\mathcal{D}_A$ and $\mathcal{D}_B$. The region of achievable specifications that are tighter than the specification $\mathcal{D}_Y$ is shaded, and similarly for $\mathcal{D}_W$. To see the significance of these regions, consider a closed-loop transfer matrix H with $(\phi_{\rm os}(H), \phi_{\rm act\_eff}(H))$ lying in the shaded region corresponding to $\mathcal{D}_Y$, and a closed-loop transfer matrix $\tilde{H}$ with $\phi_{\rm os}(\tilde{H}) = 6\%$, $\phi_{\rm act\_eff}(\tilde{H}) = 0.1$ (*i.e.*, $\tilde{H}$ just satisfies $\mathcal{D}_Y$). H is clearly a better design, in the sense that it has less overshoot *and* it has a lower actuator effort than $\tilde{H}$. $\mathcal{D}_Y$ is not Pareto optimal precisely because there are designs that are better in both overshoot and actuator effort. In general, an achievable but not Pareto optimal specification is one that can be strictly strengthened. The specifications $\mathcal{D}_A$ and $\mathcal{D}_B$ are Pareto optimal.

Yet another interpretation of the Pareto optimal specifications is in terms of the partial ordering of the design specifications. Figure 3.5 shows the partial ordering of the design specifications from figures 3.2 and 3.4, together with the boundary between achievable and unachievable specifications. For each linearly ordered chain of specifications that contains a Pareto optimal specification, all the weaker specifications will be achievable and all the tighter specifications will be unachievable. In terms of the partial ordering among specifications by strength, Pareto optimal specifications are the *minimal* achievable specifications.

The boundary between achievable and unachievable specifications is called the

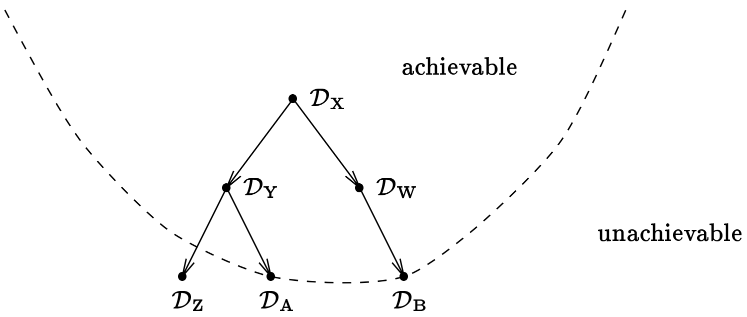


Figure 3.5 A directed graph showing the ordering (in the sense of strength) between the specifications in figures 3.2 and 3.4. The boundary between achievable and unachievable specifications is shown with a dashed line.

tradeoff curve (between the objectives ϕ_{os} and $\phi_{\mathrm{act\_eff}}$). We say that the specification $\mathcal{D}_{\mathrm{A}}$ *trades off* lower overshoot against higher actuator effort, as compared to $\mathcal{D}_{\mathrm{B}}$. More generally, with L specifications, we have a *tradeoff surface* (between the functionals $\phi_1, \ldots, \phi_L$) in performance space.

The idea of a tradeoff among various competing objectives is very important. Figure 3.6 shows two different possible regions of achievable specifications. In plot (a), the tradeoff curve is nearly straight, meaning that the overshoot and actuator effort are tightly coupled, and that we must "give up on one to improve the other". In plot (b), the tradeoff curve is quite bent, which means that we can do very well in terms of actuator effort and overshoot simultaneously: we give up only a little bit in each objective to do well in both. In this case we might say that the two functionals are nearly independent.

We note that Pareto optimal specifications themselves may be either achievable or unachievable. If a Pareto optimal specification is unachievable, however, there are arbitrarily close specifications that are achievable, so, in practice, it is irrelevant whether or not the Pareto optimal specifications can be achieved.

3.6 Optimal Controller Paradigm

In the classical optimization paradigm a family of design specifications indexed by a *single* parameter α is considered:

$$\mathcal{D}^{(\alpha)} \quad : \quad \mathcal{D}_{\mathrm{hard}} \wedge \mathcal{D}_{\mathrm{obj}}^{(\alpha)} \tag{3.3}$$

where $\mathcal{D}_{\mathrm{obj}}$ is the functional inequality specification

$$\mathcal{D}_{\mathrm{obj}}^{(\alpha)} \quad : \quad \phi_{\mathrm{obj}}(H) \le \alpha.$$

As in multicriterion optimization, $\mathcal{D}_{\mathrm{hard}}$ is the conjunction of our hard constraints. Unlike general multicriterion optimization, the family of specifications we consider,

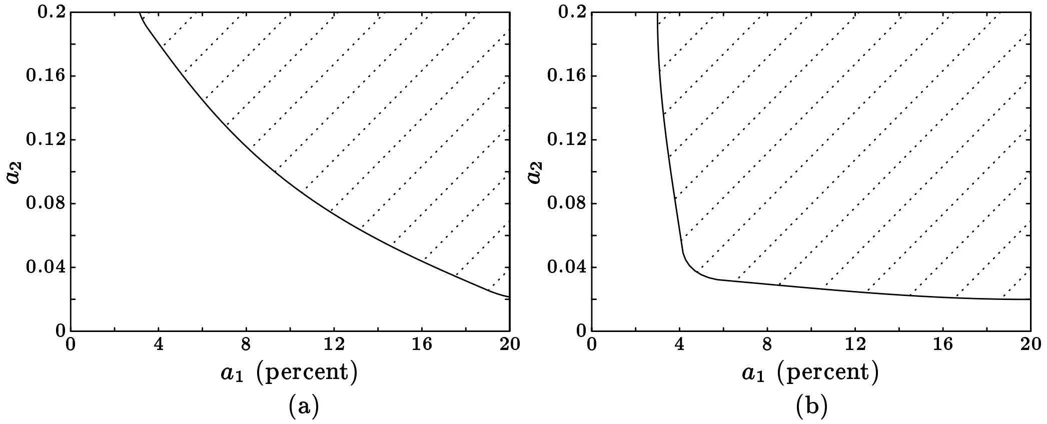


Figure 3.6 Two possible tradeoff curves. In (a), the designer must trade off any reduction in actuator effort with an increase in the step response overshoot, and vice versa. In (b), it is possible to have low overshoot and actuator effort; the specification at the knee of the tradeoff curve is not much worse than minimizing each separately.

$\mathcal{D}^{(\alpha)}$, is *linearly ordered.* Any two specifications of this form can be compared; $\mathcal{D}^{(\alpha)}$ is tighter than $\mathcal{D}^{(\beta)}$ if $\alpha \leq \beta$.

We simply seek the tightest of these specifications that is achievable; more precisely, we seek the critical value $\alpha_{\rm crit}$ such that $\mathcal{D}^{(\alpha)}$ is achievable for $\alpha > \alpha_{\rm crit}$ and unachievable for $\alpha < \alpha_{\rm crit}$. Of course, this is the unique Pareto optimal specification if we consider this a multicriterion optimization problem with one criterion. A transfer matrix $H_{\rm opt}$ that minimizes $\phi_{\rm obj}(H)$, that is, satisfies $\mathcal{D}^{(\alpha_{\rm crit})}$, is called a $\phi_{\rm obj}$-optimal design.

The classical optimization paradigm is more commonly expressed as

$$\alpha_{\rm crit} = \min_{H \text{ satisfies } \mathcal{D}_{\rm hard}} \phi_{\rm obj}(H), \tag{3.4}$$

$$H_{\rm opt} = \mathop{\arg\min}_{H \text{ satisfies } \mathcal{D}_{\rm hard}} \phi_{\rm obj}(H), \tag{3.5}$$

where the notation *arg min* denotes a minimizer of $\phi_{\rm obj}$, *i.e.*, $\alpha_{\rm crit} = \phi_{\rm obj}(H_{\rm opt})$.

While the classical optimization paradigm is formally a special case of multicriterion optimization, it is used quite differently. The objective functional $\phi_{\rm obj}$ is usually not just one of many single soft goals, as is each objective functional in multicriterion optimization; rather it generally represents some sort of combination of the many important functionals into one single functional. In the following sections we survey some of the common methods used to combine functionals into the single objective functional $\phi_{\rm obj}$.

3.6.1 Weighted-Sum Objective

One common method is to add the various functionals, after they have been multiplied by weights. We form

$$\phi_{\rm obj}(H) = \lambda_1\phi_1(H) + \cdots + \lambda_L\phi_L(H), \tag{3.6}$$

where λ_i are nonnegative numbers, called *weights*, which assign relative "values" among the functionals ϕ_i. We refer to an objective functional of the form (3.6) as a *weighted-sum* objective; the vector λ with components λ_i is called the *weight vector.* One method for choosing the weights is to scale each functional by a *typical* or *nominal* value:

$$\lambda_i = \frac{1}{\phi_i^{\rm nom}},$$

where $\phi_i^{\rm nom}$ represents some nominal value of the functional ϕ_i. We can think of these nominal values as including the (possibly different) physical units of each functional, so that each term in the sum (3.6) is dimensionless (and hence, they can be sensibly added).

For our example the designer might choose

$$\phi_{\rm os}^{\rm nom} = 10\% = 0.1, \qquad \phi_{\rm act\_eff}^{\rm nom} = 0.05,$$

so that

$$\lambda_1 = \frac{1}{0.1} = 10, \qquad \lambda_2 = \frac{1}{0.05} = 20, \tag{3.7}$$

and the objective functional is

$$\phi_{\rm obj}(H) = 10\phi_{\rm os}(H) + 20\phi_{\rm act\_eff}(H). \tag{3.8}$$

In figure 3.7(a), the lines of constant objective (3.8) in the performance plane are shown. Along each of these constant objective lines, the tradeoff is exactly increasing (or decreasing) $\phi_{\rm os}$ by 10%, while decreasing (or increasing) $\phi_{\rm act\_eff}$ by 0.05. The constant objective line $\phi_{\rm obj} = 1.91$ is tangent to the tradeoff curve; the $\phi_{\rm obj}$-optimal specification is the intersection of this line with the tradeoff curve. This specification is $a_1 = 10.78\%$, $a_2 = 0.0415$.

In figure 3.7(b), the constant objective lines and the optimum specification are shown for the weights $\lambda_1 = 20$, $\lambda_2 = 10$. These weights put more emphasis on the step response overshoot than do our original weights (3.8); the corresponding optimum specification is $a_1 = 4.86\%$, $a_2 = 0.097$, which represents a tradeoff of smaller overshoot for larger actuator effort over the optimum specification for the weights (3.8).

In the general case, we consider hyperplanes in performance space with a normal vector λ. Each such hyperplane consists of specifications that yield the same value of the objective functional $\phi_{\rm obj}$ given by (3.6). We can interpret the optimization

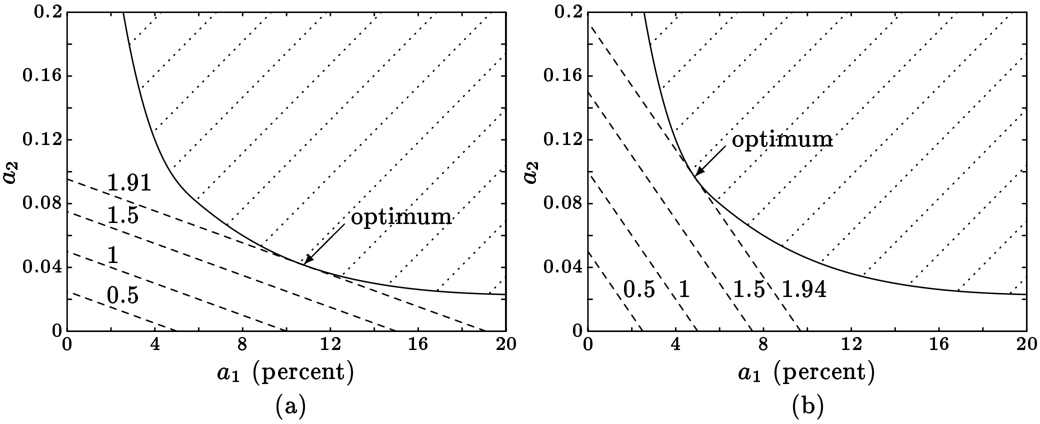


Figure 3.7 Lines of constant objective functional (3.6) are shown, together with the optimizing design. In (a) the weights are $\lambda_1 = 10$ and $\lambda_2 = 20$ and the smallest achievable objective functional value is 1.91. In (b) the weights are $\lambda_1 = 20$ and $\lambda_2 = 10$ and the smallest achievable objective functional value is 1.94. In both cases the smaller objective functional values shown lie below the performance boundary and are therefore not achievable.

problem (3.4) geometrically in performance space as follows: "go as far as possible in the direction $-\lambda$ while staying in the region of achievable specifications". More precisely,

$$\alpha_{\text{crit}} = \min\left\{\lambda^T a \;\middle|\; \mathcal{D}^{(a)} \text{ is achievable }\right\}.$$

Figure 3.7 suggests another interpretation of the classical optimization problem with objective (3.6). If the tradeoff boundary is smooth near the optimal specification a_{opt}, then a first order approximation of the tradeoff surface is given by the tangent hyperplane

$$\left\{a \;\middle|\; \lambda^T(a - a_{\text{opt}}) = 0\right\}. \tag{3.9}$$

We can use (3.9) to estimate nearby specifications on the tradeoff surface. For example, the approximate optimal tradeoff between two functionals ϕ_i and ϕ_j (with all other specifications fixed) is given by

$$\lambda_i \delta\phi_i \approx -\lambda_j \delta\phi_j,$$

where $\delta\phi_i$ represents the change in the functional ϕ_i along the tradeoff surface. Since the weights are positive, $\delta\phi_i$ and $\delta\phi_j$ have different signs, meaning that we must give up performance in one functional (*e.g.*, $\delta\phi_i > 0$) to obtain better performance in the other ($\delta\phi_j < 0$). For small changes, the ratio of the two changes is nearly the inverse ratio of the weights:

$$-\frac{\delta\phi_i}{\delta\phi_j} \approx \frac{\lambda_j}{\lambda_i}.$$

Thus, when the classical optimization paradigm is applied to the weighted-sum objective (3.6), the resulting specification is on the tradeoff surface, and the local tradeoffs between the different functionals are given by the inverse ratios of the corresponding weights.

The solution of each classical optimization problem with the weighted-sum objective (3.6) is a Pareto optimal specification for the multicriterion optimization problem with criteria $\phi_1, \ldots, \phi_L$. However, there can be Pareto optimal specifications that are not optimal for any selection of weights in the classical optimization problem with weighted-sum objective. Figure 3.8 shows an example of this. We will see in chapter 6 that in many cases the Pareto optimal specifications are exactly the same as the specifications that are optimal for some selection of weights in the classical optimization problem with weighted-sum objective.

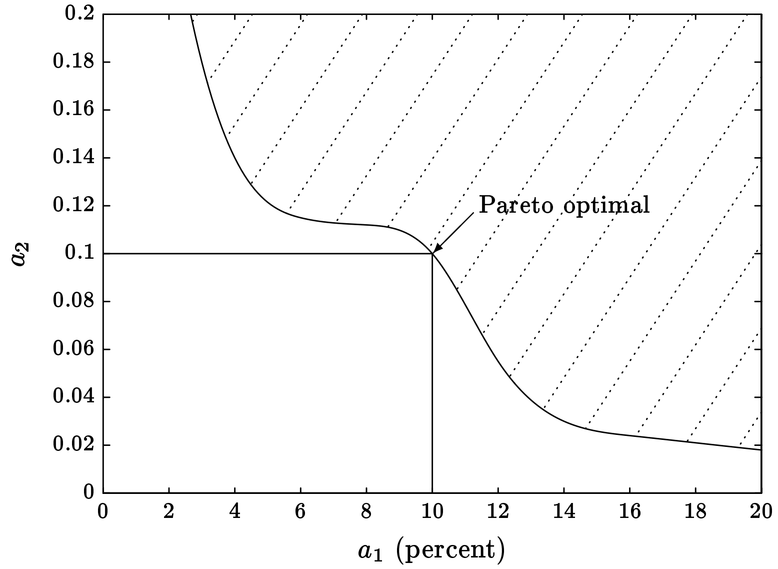


Figure 3.8 An example of a tradeoff curve with a Pareto optimal specification that is not optimal for any selection of weights in the classical optimization problem with weighted-sum objective. It *is* optimal for the weighted-max objective with weights $\lambda_1 = \lambda_2 = 10$.

3.6.2 The Dual Function

An important function defined in terms of the weighted-sum objective is the *dual objective* or *dual function* associated with the functionals $\phi_1, \ldots, \phi_L$ and the hard constraint $\mathcal{D}_{\text{hard}}$. It is defined as the mapping from the nonnegative weight vector $\lambda \in \mathbf{R}_+^L$ into the resulting minimum weighted-sum objective:

$$\psi(\lambda) \overset{\Delta}{=} \min\left\{\lambda_1\phi_1(H) + \cdots + \lambda_L\phi_L(H) \;\middle|\; H \text{ satisfies } \mathcal{D}_{\text{hard}}\right\}. \tag{3.10}$$

We will encounter the dual function in chapter 6 and chapters 12–15.

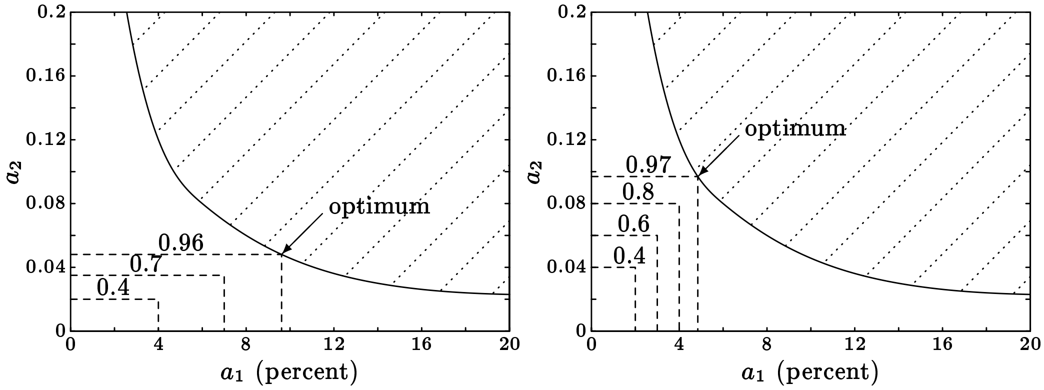
3.6.3 Weighted-Max Objective

Another approach, sometimes called *minimax design*, is to form the objective functional as the maximum of the weighted functionals:

$$\phi_{\text{obj}}(H) = \max\{\lambda_1\phi_1(H), \ldots, \lambda_L\phi_L(H)\}, \tag{3.11}$$

where λ_i are nonnegative. We refer to (3.11) as a *weighted-max* objective. As before, the weights are meant to express the designer's preferences among the functionals.

In figure 3.9 the curves of constant weighted-max objective are shown for the same two sets of weights as figure 3.7. In figure 3.9(a) the weights are $\lambda_1 = 10$, $\lambda_2 = 20$; the constant objective curve $\phi_{\text{obj}} = 0.96$ touches the tradeoff curve at $a_1 = 9.6\%$, $a_2 = 0.048$. In figure 3.9(b) the weights are $\lambda_1 = 20$, $\lambda_2 = 10$; the constant objective curve $\phi_{\text{obj}} = 0.97$ touches the tradeoff curve at $a_1 = 4.85\%$, $a_2 = 0.097$.



(a) (b)

Figure 3.9 Lines of constant minimax objective functional (3.11) are shown, together with the optimizing design. In (a) the weights are $\lambda_1 = 10$ and $\lambda_2 = 20$ and the smallest achievable objective functional value is 0.96. In (b) the weights are $\lambda_1 = 20$ and $\lambda_2 = 10$ and the smallest achievable objective functional value is 0.97. In both cases the smaller objective functional values shown lie below the performance boundary and are therefore not achievable.

This minimax approach always produces Pareto optimal specifications, and vice versa: every Pareto optimal specification arises as the solution of the minimax problem (3.11) for some choice of weights. For example, the Pareto optimal specification shown in figure 3.8 is optimal for the minimax specification with weights $\lambda_1 = \lambda_2 = 10$.

In one variation on the weighted-max objective (3.11), a constant offset is subtracted from each functional:

$$\phi_{\mathrm{obj}}(H) = \max\left\{\lambda_1(\phi_1(H) - \gamma_1), \ldots, \lambda_L(\phi_L(H) - \gamma_L)\right\}. \tag{3.12}$$

3.7 General Design Procedures

In this section we describe some design *procedures* that have been suggested and successfully used. We suppose that $\phi_1, \ldots, \phi_L$ are our criteria. From our discussion of multicriterion optimization, we know that, among the achievable specifications, the Pareto optimal ones are wise choices. It remains to choose one of these, *i.e.*, to "search" the tradeoff surface for a suitable design. This is generally done *interactively*, often by repeatedly adjusting the weights in a weighted-sum or weighted-max objective and evaluating the resulting optimal design.

3.7.1 Initial Designs

We have already mentioned one sensible set of initial weights for either the weighted-sum or weighted-max objectives: each λ_i is the inverse of a nominal value or a maximum value for the functional ϕ_i.

Several researchers have suggested the following method for determining reasonable initial weights and offsets for the offset weighted-max objective (3.12). The designer first decides what "good" and "bad" values would be for each objective; we will use G_i and B_i to denote these values. One choice is to let G_i be the minimum value of the functional ϕ_i alone. We then form the objective

$$\phi_{\mathrm{obj}}(H) = \max\left\{\frac{\phi_1(H) - G_1}{B_1 - G_1}, \ldots, \frac{\phi_L(H) - G_L}{B_L - G_L}\right\}, \tag{3.13}$$

which is the maximum of the "normalized badness" of H: $\phi_{\mathrm{obj}}(H) \approx 1$ means that at least one criterion is near a bad value; $\phi_{\mathrm{obj}}(H) \approx 0.1$ means that the worst criterion is only 10% of the way away from its good value, towards its bad value.

Another approach, called *goal programming*, starts with some "goal" values G_i for the functionals $\phi_1, \ldots, \phi_L$. The designer then determines the closest achievable specification to this goal specification, using, *e.g.*, a weighted Euclidean distance in performance space.

3.7.2 Design Iterations

Perhaps the most common method used to find a new and, one hopes, more suitable design is informal adjustment ("tweaking") of the weights in the weighted-sum or weighted-max objective for the classical optimization problem. The designer picks one criterion, say $\phi_{i_{\mathrm{bad}}}$, whose current optimal value is deemed unacceptably large, and then increases the associated weight $\lambda_{i_{\mathrm{bad}}}$. One drawback of this method is that if more than one weight is adjusted in a single iteration, it can be difficult to predict the resulting effect on the optimal values of the criteria.

Using the objective (3.13), a design iteration consists of a re-evaluation of what good and bad values are for each objective, based on the current and previous optimal values of the criteria. Similarly, in goal programming the designer can change the goal specification, or change the norm used to determine the closest achievable specification.

A procedure called the *satisficing tradeoff method* uses the objective (3.13). In this method, the B_i are called *aspiration levels*, so the goal is to find a design with $\phi_{\mathrm{obj}}(H) \approx 1$, if possible, which means that our aspirations have been met or exceeded. At each iteration, new aspiration levels are set according to various heuristic rules, *e.g.*, they should be lower for those functionals that are currently too large, and they must be larger for at least one functional that the designer feels can be worsened a bit.

Notes and References

The Feasibility Problem

In [ZA73], Zakian and Al-Naib formulate the design problem as a set of functional inequalities that must be satisfied:

> In the design of dynamical systems, such as control systems, electrical networks and other analogous systems, it is convenient to formulate the design problem in terms of the inequalities
>
> $$\phi_i(p) \leq C_i, \qquad i = 1, 2, \ldots, m$$
>
> [p represents the design]. Some of the $\phi_i(p)$ are related to the dynamical behavior of the system ... and are called functionals.

If the C_i are fixed, then Zakian and Al-Naib's formulation is what we have called the feasibility problem; if, on the other hand, the C_i are treated as parameters that the designer can vary, then the "wise" choices for the parameters are the Pareto optimal values.

The Classical Optimal Controller Paradigm

Wiener and Kolmogorov were the first to explicitly study the optimal controller paradigm. Linear controller design before that time had consisted mostly of rules for synthesizing controllers, *e.g.*, PID tuning rules [ZN42] or root locus methods [EVA50], or analysis useful in designing controllers, *e.g.*, Bode [BOD45]. In the 1957 book by Newton, Gould, and Kaiser [NGK57], the optimal controller paradigm is called the "analytical design procedure", which they describe as (p.31):

> In place of a relatively simple statement of the allowable error, the analytical design procedure employs a more or less elaborate *performance index.* The objective of the performance index is to encompass in a single number a quality measure for the performance of the system.

Multicriterion Optimization and Pareto Optimality

The notion of Pareto optimality is first explicitly described in Pareto's 1896 treatise on Economics [PAR96]; since then it has been extensively studied in the Econometrics literature, *e.g.*, Von Neumann and Morgenstern [NM53] and Debreu [DEB59]. The latter book contains many plots exactly like our plots of achievable specifications, tradeoff curves, and so on. Two recent texts on multicriterion optimization are Sawaragi, Nakayama, and Tanino [SNT85] and Luc [LUC89]. [SNT85] covers many practical aspects of multicriterion optimization and decision making; [LUC89] is a clear and complete description of vector optimization, covers topics such as convexity, quasiconvexity, and duality, and has an extensive bibliography. Brayton and Spence [BS80, CH7] has a general discussion of multicriterion optimization similar to ours.

The article [WIE82] describes satisficing decision making, gives a mathematically rigorous formulation, and has a large set of references. The procedure for selecting weights and offsets in (3.13) of section 3.7 is described in Nye and Tits [NT86], Fan et al. [FWK89], and also the book by Sawaragi, Nakayama, and Tanino [SNT85].

An early reference in the control literature to multicriterion optimization is the article [ZAD63] by Zadeh, published right at the rise in prominence of the optimal controller paradigm, which was spurred by the work of Kalman and others on the linear quadratic regulator (see chapter 12 and [AM90]). The recent book by Ng [NG89] has a general discussion of multicriterion optimization for control system design, concentrating on the "quality of cooperation between the designer and his computer".

The Example Figures

The plots of achievable specifications in performance space for our standard example are fictitious (in fact, we never specified the power spectral densities $S_{\rm proc}$ and $S_{\rm sensor}$, so our objective functionals are not even fully defined). Later in this book we will see many *real* tradeoff curves that relate to our standard example; see for example chapters 12 and 15.

Part II

ANALYTICAL TOOLS

Chapter 4

Norms of Signals

Many of the goals of controller design can be expressed in terms of the *size* of various signals, *e.g.*, tracking error signals should be made "small", while the actuator signals should not be "too large". In this chapter we explore some of the ways this notion of the size of a signal can be made precise, using *norms*, which generalize the notion of Euclidean length.

4.1 Definition

There are many ways to describe the size of a signal or to express the idea that a signal is small or large. For example, the fraction of time that the magnitude of a signal exceeds some given threshold can serve as a measure of the size of the signal; we could define "small" to mean that the threshold is exceeded less than 1% of the time. Among the many methods to measure the size of a signal, those that satisfy certain geometric properties have proven especially useful. These measures of size are called *norms.*

The geometric properties that norms satisfy are expressed in the framework of a vector space; roughly speaking, we have a notion of how to add two signals, and how to multiply or scale a signal by a scalar (see the Notes and References).

Definition 4.1: *Suppose V is a vector space and $\phi : V \to \mathbf{R}_+ \cup \{\infty\}$. ϕ is a* norm *on V if it satisfies*

Nonnegativity: $\phi(v) \geq 0$,

Homogeneity: *for* $\phi(v) < \infty$, $\phi(\alpha v) = |\alpha|\phi(v)$,

Triangle inequality: $\phi(v + w) \leq \phi(v) + \phi(w)$,

for all $\alpha \in \mathbf{R}$ and $v, w \in V$.

We warn the reader that this definition differs slightly from the standard definition of a norm; see the Notes and References at the end of this chapter.

We will generally use the notation

$$\|v\|_{\mathrm{x}} \stackrel{\Delta}{=} \phi(v),$$

where x is some distinguishing mark or mnemonic for ϕ. This notation emphasizes that a norm is a generalization of the absolute value for real or complex numbers, and the Euclidean length of a vector. Note that we allow norms to take on the value $+\infty$, just as we allow our functionals on transfer matrices to do. We interpret $\|v\| = \infty$ as "v is infinitely large" as measured by the norm $\|\cdot\|$.

In the next few sections we survey some common norms used to measure the size of signals. The verification that these norms do in fact satisfy the required properties is left as an exercise for the reader (alternatively, the reader can consult the references).

4.2 Common Norms of Scalar Signals

4.2.1 Peak

One simple but strict interpretation of "the signal u is small" is that it is small *at all times*, or equivalently, its maximum or peak absolute value is small. The *peak* or $\mathbf{L}_\infty$ *norm* of u is defined as

$$\|u\|_\infty \stackrel{\Delta}{=} \sup_{t\geq 0} |u(t)|.$$

An example of a signal u and its peak $\|u\|_\infty$ is shown in figure 4.1.

The peak norm of a signal is useful in specifying a strict limit on the absolute value of a signal, *e.g.*, the output current of a power amplifier, or the tracking error in a disk drive head positioning system.

The peak norm of a signal depends entirely on the extreme or large values the signal takes on. If the signal occasionally has large values, $\|u\|_\infty$ will be large; a statistician would say $\|u\|_\infty$ depends on *outliers* or "rare events" in the signal u. We shall soon see other norms that depend to a lesser extent on occasional large signal values.

It is useful to imagine how various signal norms might be measured. A full wave rectifier circuit that measures the peak of a voltage signal is shown in figure 4.2.

The peak norm can be used to describe a signal about which very little is known, or willing to be assumed, other than some bound on its peak or worst case value. Such a description is called an *unknown-but-bounded* model of a signal: we assume only $\|u\|_\infty \leq M$. An example is *quantization error*, the difference between a signal and its uniformly quantized value. This error can be modeled as unknown but bounded by one-half of the quantization interval.

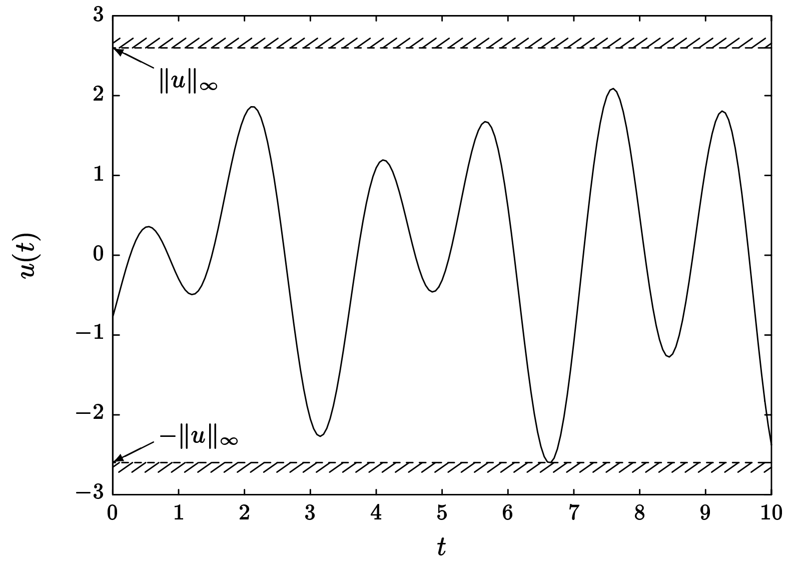


Figure 4.1 A signal u and its peak norm $\|u\|_\infty$.

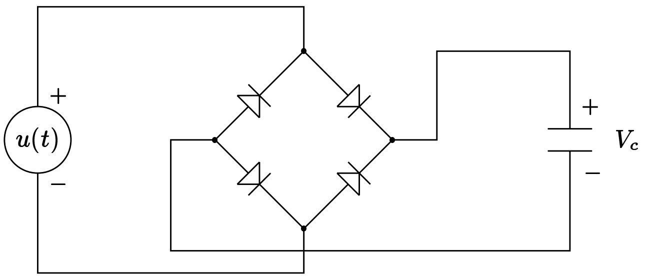


Figure 4.2 With ideal diodes and an ideal capacitor the voltage on the capacitor, V_c, tends to $\|u\|_\infty$ for t large.

A variation on the peak norm is the eventual peak or steady-state peak:

$$\|u\|_{\mathrm{ss}\infty} \overset{\Delta}{=} \limsup_{t\to\infty} |u(t)| = \lim_{T\to\infty} \sup_{t\geq T} |u(t)|.$$

The steady-state peak norm measures only *persistent* large excursions of the signal; unlike the peak norm, it is unaffected by transients, *i.e.*, the addition of a signal that decays to zero:

$$\|u + u_{\mathrm{transient}}\|_{\mathrm{ss}\infty} = \|u\|_{\mathrm{ss}\infty} \qquad \text{if} \quad \lim_{t\to\infty} u_{\mathrm{transient}}(t) = 0.$$

4.2.2 Root-Mean-Square

A measure of a signal that reflects its eventual, average size is its *root-mean-square* (RMS) *value*, defined by

$$\|u\|_{\mathrm{rms}} \overset{\Delta}{=} \left(\lim_{T\to\infty} \frac{1}{T} \int_0^T u(t)^2\, dt \right)^{1/2}, \tag{4.1}$$

provided the limit exists (see the Notes and References). This is a classical notion of the size of a signal, widely used in many areas of engineering. An example of a signal u and its RMS value $\|u\|_{\mathrm{rms}}$ is shown in figure 4.3.

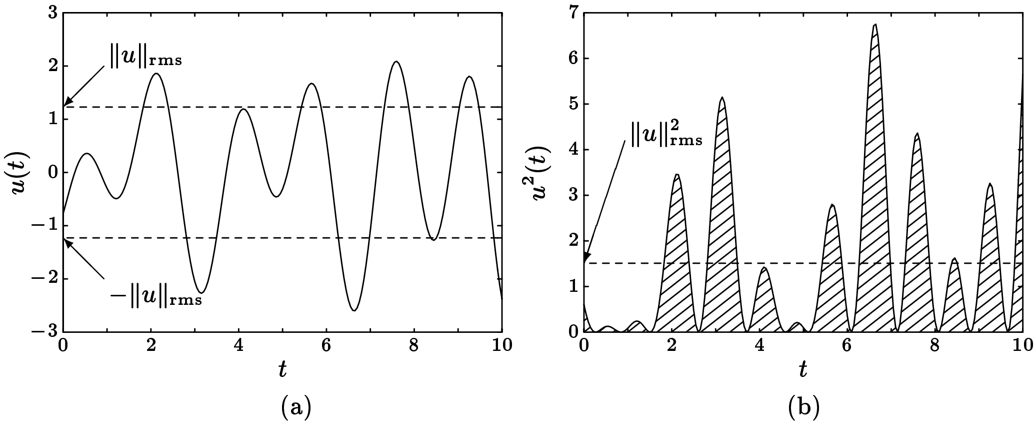


Figure 4.3 A signal u and its RMS value $\|u\|_{\mathrm{rms}}$ is shown in (a). $\|u\|^2_{\mathrm{rms}}$ is the average area under u^2, as shown in (b).

In an early RMS ammeter, shown in figure 4.4, the torque on the rotor is proportional to the square of the current; its large rotational inertia, the torsional spring, and some damping, make the rotor deflection approximately proportional to the mean-square current, $\|u\|^2_{\mathrm{rms}}$.

Another useful conceptual model for $\|u\|_{\mathrm{rms}}$ is in terms of the average power dissipated in a resistive load driven by a voltage u, as in figure 4.5. The instantaneous

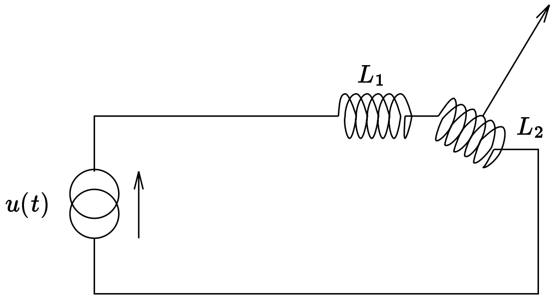


Figure 4.4 An early RMS ammeter consists of a stator L_1 and a rotor L_2, which is fitted with a needle and restoring spring. The deflection of the needle will be approximately proportional to $\|u\|_{\rm rms}^2$, the square of the RMS value of the input current u.

power dissipated in the load resistor is simply $u(t)^2/R$; the (long term) temperature rise of the large thermal mass above ambient temperature is proportional to the average power dissipation in the load, $\|u\|_{\rm rms}^2/R$. This conceptual model shows that the RMS measure is useful in specifying signal limits that are due to steady-state thermal considerations such as maximum power dissipation and temperature rise. For example, the current through a voice coil actuator might be limited by a maximum allowable steady-state temperature rise for the voice coil; this specification could be expressed as an RMS limit on the voice coil current.

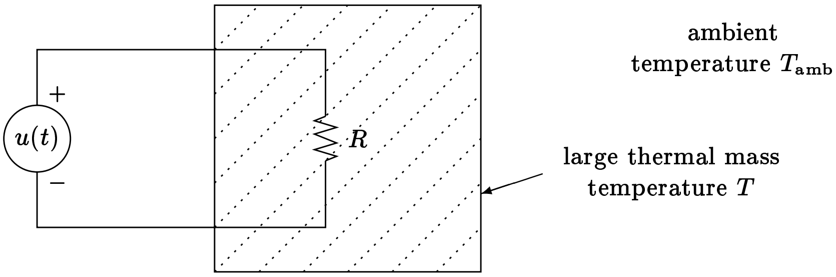


Figure 4.5 If u varies much faster than the thermal time constant of the mass, then the long term temperature rise of the mass is proportional to the average power in u, *i.e.* $T - T_{\rm amb} \propto \|u\|_{\rm rms}^2$, where $T_{\rm amb}$ is the ambient temperature, and T is the temperature of the mass.

Even if the RMS norm of a signal is small, the signal may occasionally have large peaks, provided the peaks are not too frequent and do not contain too much energy. In this sense, $\|u\|_{\rm rms}$ is less affected than $\|u\|_\infty$ by large but infrequent values of the signal. We also note that the RMS norm is a *steady-state* measure of a signal; the RMS value of a signal is not affected by any transient. In particular, a signal with small RMS value can be very large for some initial time period.

4.2.3 Average-Absolute Value

A measure that puts even less emphasis on large values of a signal (indeed, the minimum emphasis possible to still be a norm) is its *average-absolute value*, defined by

$$\|u\|_{\mathrm{aa}} \overset{\Delta}{=} \lim_{T\to\infty} \frac{1}{T}\int_0^T |u(t)|\,dt, \tag{4.2}$$

provided the limit exists (see the Notes and References). An example of a signal u and its average-absolute norm $\|u\|_{\mathrm{aa}}$ is shown in figure 4.6. $\|u\|_{\mathrm{aa}}$ can be measured with the circuit shown in figure 4.7 (*c.f.* the peak detector circuit in figure 4.2).

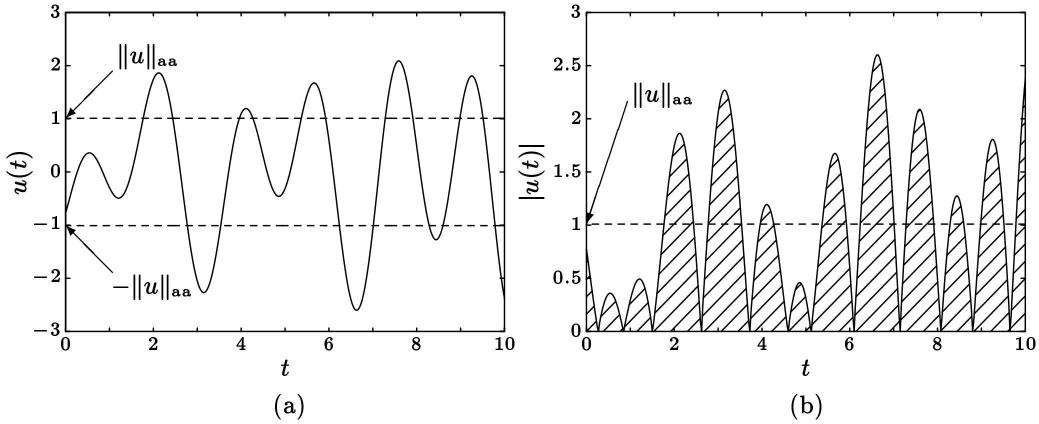


Figure 4.6 A signal u and its average-absolute value $\|u\|_{\mathrm{aa}}$ is shown in (a). $\|u\|_{\mathrm{aa}}$ is found by finding the average area under $|u|$, as shown in (b).

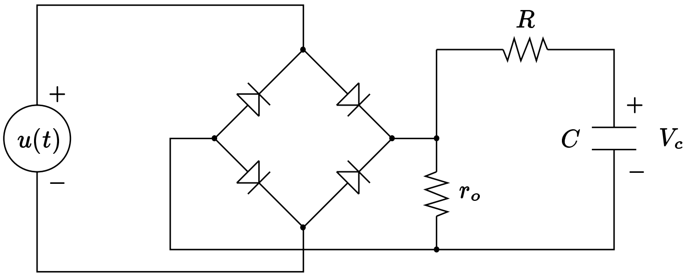


Figure 4.7 If u varies much faster than the time constant RC then V_c will be nearly proportional to the average of the peak of the input voltage u, so that $V_c = \|u\|_{\mathrm{aa}}$. The resistor $r_o \ll R$ ensures that the output impedance of the bridge is low at all times.

The average-absolute norm $\|u\|_{\mathrm{aa}}$ is useful in measuring *average fuel* or *resource use*, when the fuel or resource consumption is proportional to $|u(t)|$. In contrast, the

RMS norm $\|u\|_{\mathrm{rms}}^2$ is useful in measuring average *power*, which is often proportional to $u(t)^2$. Examples of resource usage that might be measured with the average-absolute norm are rocket fuel use, compressed air use, or power supply demand in a conventional class B amplifier, as shown in figure 4.8.

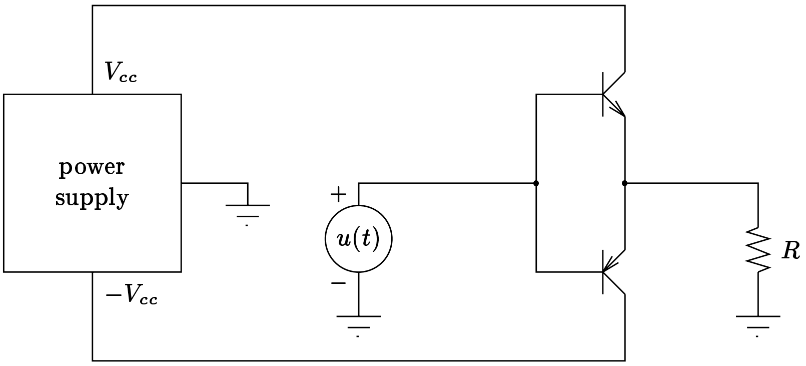


Figure 4.8 Idealized version of a class B power amplifier, with no bias circuitry shown. Provided the amplifier does not clip, *i.e.*, $\|u\|_\infty \le V_{cc}$, the average power supplied by the power supply is proportional to $\|u\|_{\mathrm{aa}}$, the average-absolute norm of u; the average power dissipated in the load R is proportional to $\|u\|_{\mathrm{rms}}^2$, the square of the RMS norm of u.

4.2.4 Norms of Stochastic Signals

For a signal modeled as a stationary stochastic process, the measure of its size most often used is

$$\|u\|_{\mathrm{rms}} \stackrel{\Delta}{=} \left(\mathbf{E}\, u(t)^2\right)^{1/2}. \tag{4.3}$$

Because the process is stationary, the expression in (4.3) does not depend on t. For stochastic signals that approach stationarity as time goes on, we define

$$\|u\|_{\mathrm{rms}} \stackrel{\Delta}{=} \left(\lim_{t\to\infty} \mathbf{E}\, u(t)^2\right)^{1/2}.$$

If the signal u is ergodic, then its RMS norm can be computed either by (4.3) or (4.1): with probability one the deterministic and stochastic RMS norms are equal.

The RMS norm can be expressed in terms of the autocorrelation of u,

$$R_u(\tau) \stackrel{\Delta}{=} \mathbf{E}\, u(t)u(t+\tau), \tag{4.4}$$

or its power spectral density,

$$S_u(\omega) \stackrel{\Delta}{=} \int_{-\infty}^{\infty} R_u(\tau)e^{-j\omega\tau}\, d\tau, \tag{4.5}$$

as follows:

$$\|u\|_{\mathrm{rms}}^2 = R_u(0) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_u(\omega)\, d\omega. \tag{4.6}$$

We can interpret the last integral as follows: the average power in the signal is the integral of the contribution at each frequency.

For stochastic signals, the analogs of the average-absolute or peak norm are less often encountered than the RMS norm. For u stationary, we define the average-absolute norm as

$$\|u\|_{\mathrm{aa}} \stackrel{\Delta}{=} \mathbf{E}\,|u(t)|,$$

which for ergodic signals agrees (with probability one) with our deterministic definition of $\|u\|_{\mathrm{aa}}$. We interpret $\|u\|_{\mathrm{aa}}$ as the expected or mean resource consumption.

The analog of the steady-state peak of u is the *essential sup* norm,

$$\|u\|_{\mathrm{ess\_sup}} \stackrel{\Delta}{=} \inf\left\{a \;\middle|\; \mathbf{Prob}(|u(t)| \geq a) = 0\right\},$$

or equivalently, the smallest number a such that with probability one, $|u(t)| \leq a$. Under some mild technical assumptions about u, this agrees with probability one with the steady-state peak norm of u defined in section 4.2.1.

4.2.5 Amplitude Distributions

We can think of the steady-state peak norm, RMS norm, and average-absolute norm as differing in the relative weighting of large versus small signal values: the steady-state peak norm is entirely dependent on the large values of a signal; the RMS norm is less dependent on the large values, and the average-absolute norm less still.

This idea can be made precise by considering the notion of the *amplitude distribution* $\mathbf{F}_u(a)$ of a signal u, which is, roughly speaking, the fraction of the time the signal exceeds the limit a, or the probability that the signal exceeds the limit a at some particular time.

We first consider stationary ergodic stochastic signals. The amplitude distribution is just the probability distribution of the absolute value of the signal:

$$\mathbf{F}_u(a) \stackrel{\Delta}{=} \mathbf{Prob}(|u(t)| \geq a).$$

Since u is stationary, this expression does not depend on t.

We can also express $\mathbf{F}_u(a)$ in terms of the fraction of time the absolute value of the signal exceeds the threshold a. Consider the time interval $[0,T]$. Over this time interval, the signal u will spend some fraction of the total time T with $|u(t)| \geq a$. $\mathbf{F}_u(a)$ is the limit of this fraction as $T \to \infty$:

$$\mathbf{F}_u(a) = \lim_{T\to\infty} \frac{\mu\{t \mid 0 \leq t \leq T,\ |u(t)| \geq a\}}{T}, \tag{4.7}$$

where $\mu(\cdot)$ denotes the total length (Lebesgue measure) of a subset of the real line. These two ideas are depicted in figure 4.9. The amplitude distribution of the signal in figure 4.9 is shown in figure 4.10.

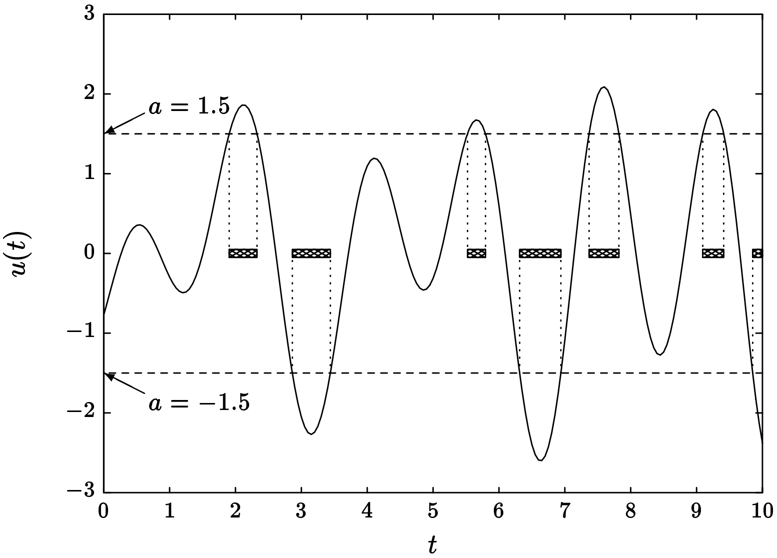


Figure 4.9 Example of calculating $F_u(1.5)$ for the signal u in figure 4.1. For $T = 10$, $\mu\{t \mid 0 \leq t \leq T,\ |u(t)| \geq 1.5\}$ is the length of the shaded intervals, and this length divided by T approximates $F_u(1.5)$.

This last interpretation of the amplitude distribution in terms of the fraction of time the signal exceeds any given threshold allows us to extend the notion of amplitude distribution to some deterministic (non-stochastic) signals. For a deterministic u, we define $\mathbf{F}_u(a)$ to be the limit (4.7), provided this limit exists (it need not). All of the results of this section hold for a suitably restricted set of deterministic signals, if we use this definition of amplitude distribution. There are many more technical details in such a treatment of deterministic signals, however, so we continue under the assumption that u is a stationary ergodic stochastic process.

Clearly $\mathbf{F}_u(a) = 0$ for $a > \|u\|_{\mathrm{ss}\infty}$, and $\mathbf{F}_u(a)$ increases to one as a decreases to zero. Informally, we think of $|u(t)|$ as spending a large fraction of time where the slope of $\mathbf{F}_u(a)$ is sharp; if $\mathbf{F}_u$ decreases approximately linearly, we say $|u(t)|$ is approximately uniformly distributed in amplitude. Figure 4.11 shows two signals and their amplitude distribution functions.

We can compute the steady-state peak, RMS, and average-absolute norms of u directly from its amplitude distribution. We have already seen that

$$\|u\|_{\mathrm{ss}\infty} = \sup\{a \mid \mathbf{F}_u(a) > 0\}. \tag{4.8}$$

The steady-state peak of a signal is therefore the value of a at which the graph of the amplitude distribution first becomes zero.

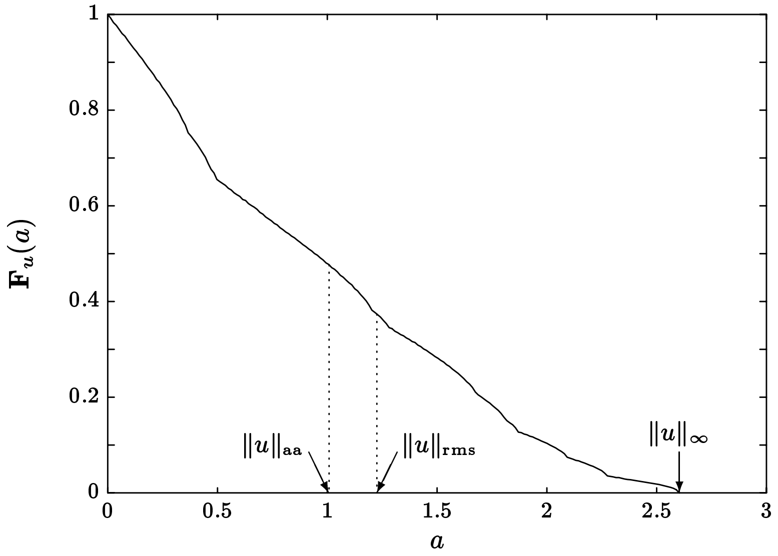


Figure 4.10 The amplitude distribution $\mathbf{F}_u$ of the signal u shown in figure 4.1, together with the values $\|u\|_{\rm aa}$, $\|u\|_{\rm rms}$, and $\|u\|_\infty$.

From elementary probability theory we have

$$\|u\|_{\rm aa} = \mathbf{E}\,|u(t)| = \int_0^\infty \mathbf{F}_u(a)\,da. \tag{4.9}$$

Thus, the average-absolute norm of a signal is the total area under the amplitude distribution function.

Since the amplitude distribution function of u^2 is $\mathbf{F}_{u^2}(a) = \mathbf{F}_u(\sqrt{a})$, equation (4.9) yields

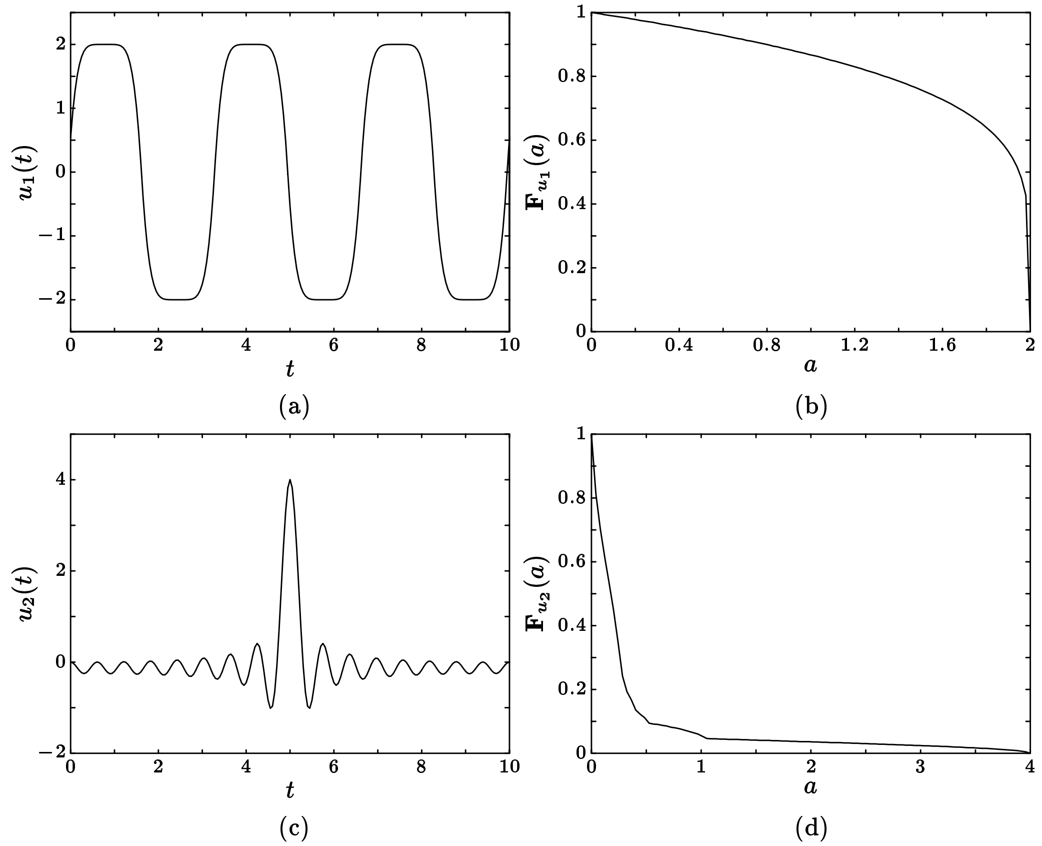
$$\mathbf{E}\,u(t)^2 = \int_0^\infty \mathbf{F}_{u^2}(a)\,da = \int_0^\infty \mathbf{F}_u(\sqrt{a})\,da = \int_0^\infty 2a\mathbf{F}_u(a)\,da,$$

so that we can express the RMS norm as:

$$\|u\|_{\rm rms}^2 = \int_0^\infty 2a\mathbf{F}_u(a)\,da. \tag{4.10}$$

Thus, the average power in the signal is the integral of its amplitude distribution function times $2a$. Just as we interpret formula (4.6) as expressing the average power in the signal as the integral of the contributions at all frequencies, we may interpret (4.10) as expressing the average power in the signal as the integral of the contributions from all possible signal amplitudes.

Comparison of the three formulas (4.8), (4.9), and (4.10) show that the three norms simply put different emphasis on large and small signal values: the steady-state peak norm puts all of its emphasis on large values; the RMS norm puts



(a) (b)

(c) (d)

Figure 4.11 Examples of periodic signals are shown in (a) and (c). Their respective amplitude distribution functions are shown in (b) and (d). The signal in (a) spends most of its time near its peaks; the amplitude distribution falls rapidly near $a = \|u_1\|_\infty$. The signal in (c) spends most of its time near 0; the amplitude distribution falls rapidly near $a = 0$.

linearly weighted emphasis on signal amplitudes; and the average-absolute norm puts uniform weight on all signal amplitudes.

4.2.6 $\mathbf{L}_2$: Square Root Total Energy

The previous sections dealt with the sizes of signals that persist. In this section and the next we examine some norms appropriate for transient signals, which decay to zero as time progresses; such signals have zero as their steady-state peak, RMS, and average-absolute norms.

The total energy or $\mathbf{L}_2$ norm of a signal is defined by

$$\|u\|_2 \stackrel{\Delta}{=} \left(\int_0^\infty u(t)^2\,dt\right)^{1/2}.$$

This norm is the appropriate analog of the RMS norm for decaying signals, *i.e.*, signals with finite *total energy* as opposed to finite *steady-state power.*

A useful conceptual model for the $\mathbf{L}_2$ norm is shown in figure 4.12. Here we think of u as a driving voltage to a resistive load immersed in a thermal mass. This thermal mass is isolated (adiabatic), unlike the mass of figure 4.5, which is connected to the ambient temperature (a very large thermal mass) through a finite thermal resistance. The eventual temperature rise of the isolated thermal mass is proportional to $\|u\|_2^2$.

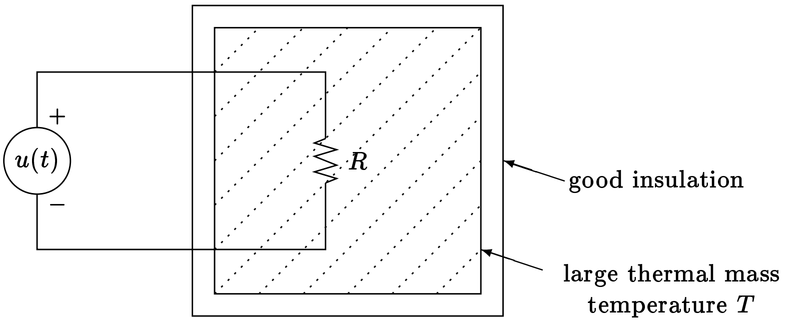


Figure 4.12 The long term temperature rise of the mass is proportional to the total energy in u, *i.e.* $T - T_{\text{init}} \propto \|u\|_2^2$, where T_{init} is the initial temperature of the mass, and T is the final temperature of the mass.

As a practical example, suppose that u represents the current through a voice coil drive during a step input in commanded position, and the thermal time constant of the voice coil is longer than the time over which u is large. Then $\|u\|_2^2$ is a measure of the temperature rise in the voice coil during a step command input.

By Parseval's theorem, the $\mathbf{L}_2$ norm can be computed as an $\mathbf{L}_2$ norm in the frequency domain:

$$\|u\|_2 = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty} |U(j\omega)|^2\, d\omega\right)^{1/2}. \tag{4.11}$$

4.2.7 $\mathbf{L_1}$: Total Fuel or Resource Consumption

The $\mathbf{L_1}$ norm of a signal is defined as

$$\|u\|_1 \triangleq \int_0^{\infty} |u(t)|\, dt.$$

Just as the $\mathbf{L_2}$ norm measures the total energy in a signal, while the RMS norm measures its average power, the $\mathbf{L_1}$ norm of a signal can be thought of as measuring a total resource consumption, while the average-absolute norm measures a steady-state average resource consumption. For example, if u represents the compressed gas flow through a nozzle during a particular spacecraft maneuver, then $\|u\|_1$ is proportional to the total gas consumed during the maneuver.

4.2.8 Frequency Domain Weights

The norms described above can be combined with an initial linear transformation that serves to emphasize or de-emphasize certain aspects of a signal. Typically, this initial transformation consists of passing the signal through an LTI filter with transfer function W, which we refer to as a *frequency domain weight*, as shown in figure 4.13.

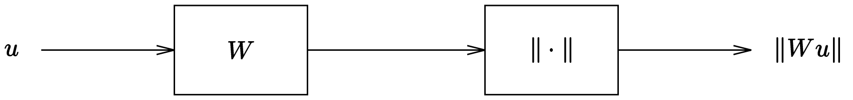


Figure 4.13 A frequency domain weighted norm is computed by passing the signal u through an LTI filter W, and then determining the (unweighted) norm of this filtered signal.

The idea is that the weighting filter makes the norm more "sensitive" (*i.e.*, assign larger values) to signals that have a large power spectral density at those frequencies where $|W(j\omega)|$ is large. This idea can be made precise for the W-weighted RMS norm, which we will denote $\|\cdot\|_{W,\mathrm{rms}}$. The power spectral density of the filtered signal Wu is

$$S_{Wu}(\omega) = S_u(\omega)|W(j\omega)|^2,$$

so the RMS norm of Wu is

$$\|u\|_{W,\mathrm{rms}} = \|Wu\|_{\mathrm{rms}} = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty} S_u(\omega)|W(j\omega)|^2\, d\omega\right)^{1/2}. \tag{4.12}$$

Thus, the weight emphasizes the power spectral density of u where $|W(j\omega)|$ is large, meaning it contributes more to the integral, and de-emphasizes the power spectral density of u where $|W(j\omega)|$ is small. We note from (4.12) that the weighted norm depends only on the *magnitude* of the weighting transfer function W, and not its phase. This is not true of all weighted norms.

We can view the effect of the weight W as changing the relative importance of different frequencies in the total power integral (4.12). We can also interpret the W-weighted RMS norm in terms of the average power conceptual model shown in figure 4.5, by changing the resistive load R to a frequency-dependent load, *i.e.*, a more general passive *admittance* $G(s)$. The load admittance G is related to the weight W by the *spectral factorization*

$$\frac{G(s)+G(-s)}{2} = W(s)W(-s), \tag{4.13}$$

so that $\Re G(j\omega) = |W(j\omega)|^2$. Since the average power dissipated in G at a frequency ω is proportional to the real part (resistive component) of $G(j\omega)$, we see that the total average power dissipated in G is given by the square of (4.12), or the square of the W-weighted RMS norm of u.

For example, suppose that $W(s) = (1+\sqrt{2}s)/(1+s)$, which gives up to 3dB emphasis at frequencies above $\sqrt{2}$. The load admittance for this weight is $G(s) = (1+2s)/(1+s)$, which we realize as the parallel connection of a 1Ω resistor and the series connection of a 1Ω resistor and a $1F$ capacitor, as shown in figure 4.14.

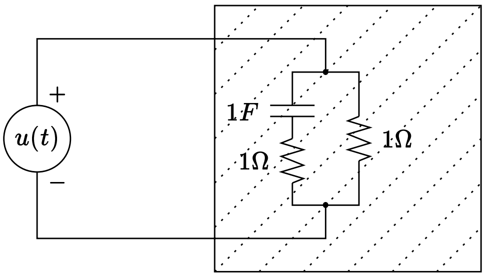


Figure 4.14 The average power dissipated in the termination admittance $G(s) = (1+2s)/(1+s)$ is $\|u\|^2_{W,\mathrm{rms}}$, the square of the W-weighted RMS norm of the driving voltage u, where $W(s) = (1+\sqrt{2}s)/(1+s)$.

Frequency domain weights are used less often with the other norms, mostly because their effect is harder to understand than the simple formula (4.12). One common exception is the *maximum slew rate*, which is the peak norm used with the weight $W(s) = s$, a differentiator:

$$\|u\|_{\mathrm{slew\_rate}} = \|\dot{u}\|_\infty. \tag{4.14}$$

Maximum slew-rate specifications occur frequently, especially on actuator signals. For example, an actuator signal may represent the position of a large valve, which is opened and closed by a motor that has a maximum speed. A graphical interpretation of the slew-rate constraint $\|u\|_{\mathrm{slew\_rate}} \leq 1$ is shown in figure 4.15.

The peak norm is sometimes used with higher order differentiators:

$$\|u\|_{\mathrm{acc}} \stackrel{\Delta}{=} \left\| \frac{d^2u}{dt^2} \right\|_\infty,$$

$$\|u\|_{\mathrm{jerk}} \stackrel{\Delta}{=} \left\| \frac{d^3u}{dt^3} \right\|_\infty.$$

Weights that are successively higher order differentiators yield the amusing *snap*, *crackle*, and *pop* norms.

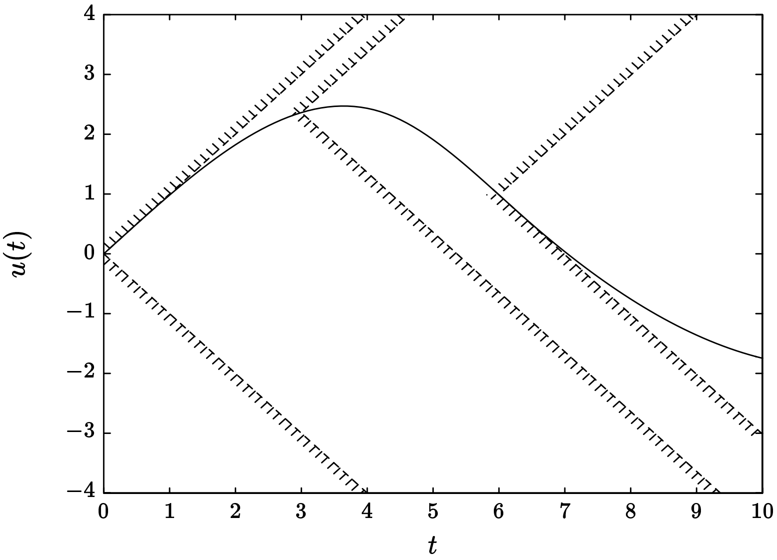


Figure 4.15 An interpretation of the maximum slew-rate specification $\|\dot{u}\| \leq 1$: at every time t the graph of u must evolve within a cone, whose sides have slopes of ±1. Examples of these cones are shown for $t = 0,\ 3,\ 6$; u is slew-rate limited at $t = 0$ and $t = 6$, since $\dot{u}(0) = 1$ and $\dot{u}(6) = -1$.

4.2.9 Time Domain Weights

If the initial linear transformation consists of multiplying the signal by some given function of time $w(t)$, we refer to w as a *time domain weight.* One example is the ITAE (integral of time multiplied by absolute error) norm from classical control, defined as

$$\|u\|_{\mathrm{itae}} \stackrel{\Delta}{=} \int_0^\infty t|u(t)|\,dt.$$

This is simply the $\mathbf{L}_1$ norm of u with the time domain weight $w(t) = t$, which serves to emphasize the signal u at large times, and de-emphasize the signal at small times.

One commonly used family of time domain weights is the family of *exponential weightings*, which have the form $w(t) = \exp(at)$. If a is positive, such a weighting exponentially emphasizes the signal at large times. This may be appropriate for measuring the size of a decaying signal. Alternatively, we can think of a specification such as $\|\tilde{u}\|_\infty \le M$ (where $\tilde{u}(t) = \exp(at)u(t)$, and $a > 0$) as enforcing a rate of decay in the signal u at least as fast as $\exp(-at)$. An example of a signal u and the exponentially scaled signal $u(t)e^{-t}$ is shown in figure 4.16.

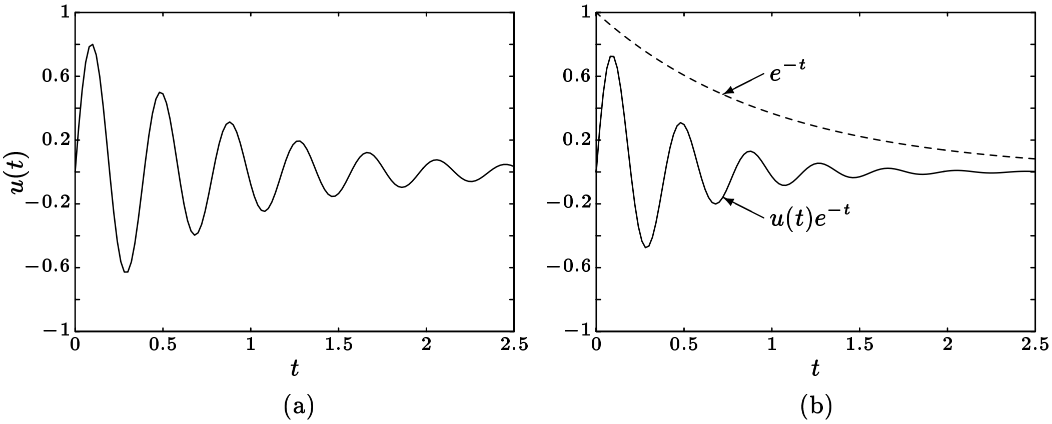


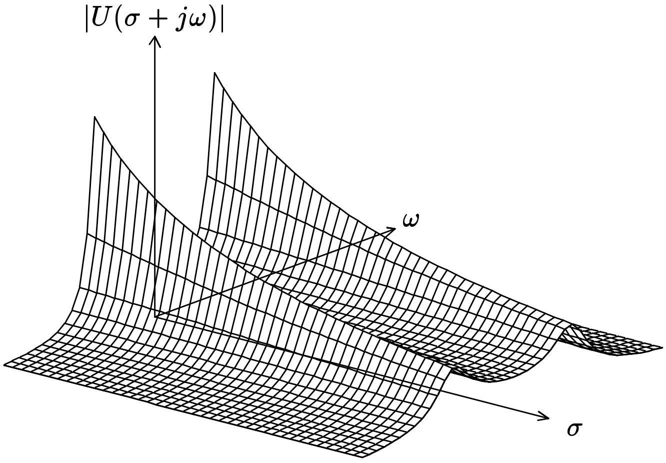
Figure 4.16 A signal u is shown in (a). The exponentially weighted $\mathbf{L}_2$ norm of u is the $\mathbf{L}_2$ norm of the signal $\tilde{u}(t) = u(t)e^{-t}$, which is shown in (b).

If a is negative, then the signal is exponentially de-emphasized at large times. This might be useful to measure the size of a diverging or growing signal, where the value of an unweighted norm is infinite.

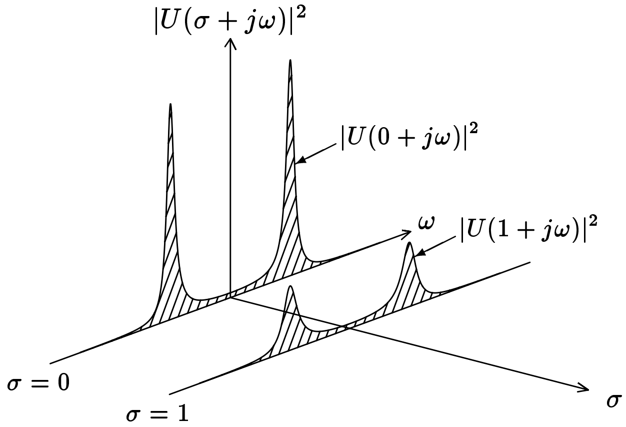
There is a simple frequency domain interpretation of the exponentially weighted $\mathbf{L}_2$ norms. If the a-exponentially weighted $\mathbf{L}_2$ norm of a signal u is finite then its Laplace transform $U(s)$ is analytic in the region $\{s \mid \Re s > -a\}$, and in fact

$$\|\tilde{u}\|_2^2 = \int_0^\infty (\exp(at)u(t))^2\, dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |U(-a + j\omega)|^2\, d\omega \tag{4.15}$$

(recall $\tilde{u}(t) = \exp(at)u(t)$). The only difference between (4.15) and (4.11) is that the integral in the a-exponentially weighted norm is *shifted* by the weight exponent a. The frequency domain calculations of the $\mathbf{L}_2$ norm and the a-exponentially weighted $\mathbf{L}_2$ norm for the signal in figure 4.16(a) are shown in figure 4.17.



(a)



(b)

Figure 4.17 An exponentially weighted $\mathbf{L}_2$ norm can be calculated in the frequency domain. Consider $\|u\|_{2,-1} = \|\tilde{u}\|_2$, where $\tilde{u}(t) = e^{-t}u(t)$ is shown in figure 4.16(b). $|U(\sigma + j\omega)|$, the magnitude of the Laplace transform of u, is shown in (a) for $\sigma \geq 0$. As shown in (b), $\|u\|_2^2$ is proportional to the area under $|U(j\omega)|^2$, and $\|u\|_{2,-1}^2$ is proportional to the area under $|U(1 + j\omega)|^2$.

4.3 Common Norms of Vector Signals

The norms for scalar signals described in section 4.2 have natural extensions to vector signals. We now suppose that $u : \mathbf{R}_+ \to \mathbf{R}^n$, *i.e.*, $u(t) \in \mathbf{R}^n$ for $t \geq 0$.

4.3.1 Peak

The peak, or $\mathbf{L}_\infty$ norm, of a vector signal is defined to be the maximum peak of any of its components:

$$\|u\|_\infty \stackrel{\Delta}{=} \max_{1\leq i\leq n} \|u_i\|_\infty = \sup_{t\geq 0} \max_{1\leq i\leq n} |u_i(t)|$$

(note the different uses of the notation $\|\cdot\|_\infty$). Thus, "$\|u\|_\infty$ is small" means that *every* component of u is *always* small. A two-input peak detector is shown in figure 4.18.

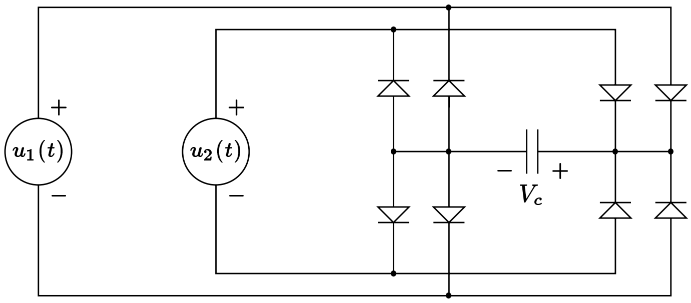


Figure 4.18 With ideal diodes and an ideal capacitor, the voltage on the capacitor, V_c, tends to $\|u\|_\infty$ for t large, where $u = [u_1\ u_2]^T$ is a vector of two signals.

4.3.2 RMS

We define the RMS norm of a vector signal as

$$\|u\|_{\rm rms} \stackrel{\Delta}{=} \left(\lim_{T\to\infty} \frac{1}{T} \int_0^T u(t)^T u(t)\, dt \right)^{1/2}, \tag{4.16}$$

provided this limit exists (see the Notes and References). For an ergodic wide-sense stationary stochastic signal this can be expressed

$$\|u\|_{\rm rms}^2 = \mathbf{E}\,\|u(t)\|_2^2 = \mathbf{Tr}\, R_u(0) = \mathbf{Tr}\, \frac{1}{2\pi} \int_{-\infty}^{\infty} S_u(\omega)\, d\omega.$$

(For vector signals, the autocorrelation is defined by

$$R_u(\tau) \stackrel{\Delta}{=} \mathbf{E}\, u(t)u(t+\tau)^T$$

(*c.f.* (4.4))). For such signals we have

$$\|u\|_{\mathrm{rms}} = \left(\sum_{i=1}^{n} \|u_i\|_{\mathrm{rms}}^2 \right)^{1/2}, \tag{4.17}$$

i.e., $\|u\|_{\mathrm{rms}}$ is the square root of the sum of the mean-square values of the components of u.

A conceptual model for the RMS value of a vector voltage signal is shown in figure 4.19.

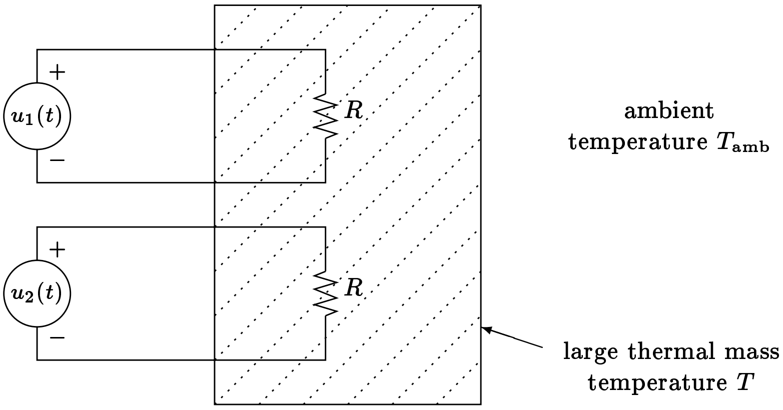


Figure 4.19 If u_1 and u_2 vary much faster than the thermal time constant of the mass, then the long term temperature rise of the mass is proportional to the average power in the vector signal u, *i.e.* $T - T_{\mathrm{amb}} \propto \|u\|_{\mathrm{rms}}^2$, where T_{amb} is the ambient temperature, and T is the temperature of the mass.

4.3.3 Average-Absolute

The average-absolute norm of a vector signal is defined by

$$\|u\|_{\mathrm{aa}} \stackrel{\Delta}{=} \limsup_{T\to\infty} \frac{1}{T} \int_0^T \sum_{i=1}^{n} |u_i(t)|\, dt.$$

This measures the average total resource consumption of all the components of u.

4.3.4 $\mathbf{L}_2$ and $\mathbf{L}_1$ Norms

The $\mathbf{L}_2$ and $\mathbf{L}_1$ norms of a vector signal are defined by

$$\|u\|_2 \triangleq \left(\int_0^\infty \sum_{i=1}^n u_i(t)^2 \, dt \right)^{1/2} = \left(\sum_{i=1}^n \|u_i\|_2^2 \right)^{1/2},$$

$$\|u\|_1 \triangleq \int_0^\infty \sum_{i=1}^n |u_i(t)| \, dt = \sum_{i=1}^n \|u_i\|_1.$$

4.3.5 Scaling and Weighting

A very important concept associated with vector signal norms is *scaling*, which can be thought of as assigning relative weights to the different components of the vector signal. For example, suppose that u_1 represents the output voltage and u_2 the output current of a power amplifier that voltage saturates (clips) at ±100V and current limits at ±2A. One appropriate measure of the peak of this signal is

$$\|u\|_{D,\infty} = \sup_{t\geq 0} \max \left\{ \frac{|u_1(t)|}{100}, \frac{|u_2(t)|}{2} \right\} = \|Du\|_\infty$$

where

$$D = \left[\begin{array}{cc} 1/100 & 0 \\ 0 & 1/2 \end{array} \right].$$

D is referred to as a *scaling* matrix, and $\|u\|_{D,\infty}$ the D-scaled peak of u. We can interpret $\|u\|_{D,\infty}$ as the size of u, relative to amplifier voltage or current overload: $\|u\|_{D,\infty} = 0.5$ indicates 6dB of headroom before the amplifier overloads; $\|u\|_{D,\infty} = 1.0$ indicates that the amplifier will just saturate or current limit.

When the different components of a vector signal u represent different physical quantities, as in the example above, the use of an appropriate scaling matrix is crucial. It is useful to think of the scaling matrix as including the translation factors among the various physical units of the components of u: for our example above,

$$D = \left[\begin{array}{cc} 1/100\mathrm{V} & 0 \\ 0 & 1/2\mathrm{A} \end{array} \right],$$

which properly renders $\|u\|_{D,\infty}$ unitless.

A simple rule-of-thumb is to use scale factors that are inversely proportional to what we might consider typical, nominal, or maximum acceptable values for that component of the signal. The scaling in our example above is based on this principle, using the maximum acceptable (overload) values. We have already encountered this idea in sections 3.6.1 and 3.6.3.

For average-absolute or $\mathbf{L}_1$ norms, the scale factors can be interpreted as relative costs or values of the resources or commodities represented by the different components of the signal. For RMS or $\mathbf{L}_2$ norms, the scale factors might represent different termination resistances in the conceptual model in figure 4.19.

Scaling is a very special form of weighting, since it consists of applying a linear transformation to the signal before computing its norm; the linear transformation is just multiplication by a diagonal matrix. Of course, it is also possible to multiply the signal vector by a nondiagonal matrix before computing the norm, as in

$$\|u\|_{A,2} = \|Au\|_2,$$

where A is some matrix. One familiar example of this is the weighted $\mathbf{L}_2$ norm,

$$\|u\|_{A,2} = \left(\int_0^\infty u(t)^T R u(t)\, dt \right)^{1/2},$$

where $R = A^T A$. A is called a (constant) *weight matrix.* Constant weight matrices can be used to emphasize some directions in $\mathbf{R}^n$ while de-emphasizing others, whereas with (diagonal) scaling matrices we are restricted to directions aligned with the axes. An example of this distinction is shown in figure 4.20 for the constraint

$$\|u\|_{A,\infty} = \|Au\|_\infty \le 1.$$

When A is diagonal the signal $u(t)$ is constrained to lie in a rectangle at each time t. For a general matrix A, the signal $u(t)$ is constrained to lie in a trapezoid.

More generally, we can preprocess the signal by a *weighting transfer matrix* W that has n columns, as in

$$\|u\|_{W,\mathrm{rms}} = \|Wu\|_{\mathrm{rms}}.$$

Often, W is square, *i.e.*, it has n rows as well. W might emphasize different "directions" at different frequencies.

4.4 Comparing Norms

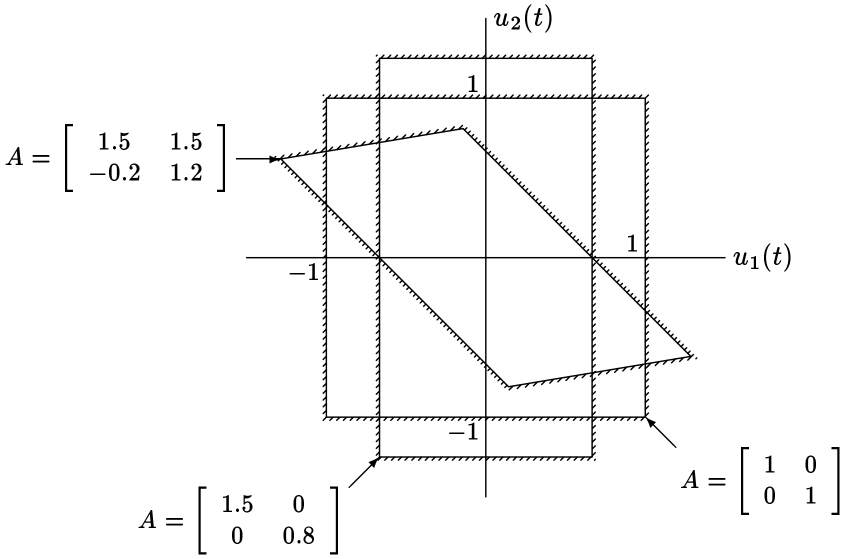
We have seen many norms for signals. A natural question is: how different can they be? Intuition suggests that since these different norms each measure the "size" of a signal, they should generally agree about whether a signal is "small" or "large". This intuition is generally false, however.

For scalar signals we have

$$\|u\|_\infty \ge \|u\|_{\mathrm{rms}} \ge \|u\|_{\mathrm{aa}}; \tag{4.18}$$

for vector signals with n components we have the generalization

$$\|u\|_\infty \ge \frac{1}{\sqrt{n}}\|u\|_{\mathrm{rms}} \ge \frac{1}{n}\|u\|_{\mathrm{aa}}. \tag{4.19}$$



$u_2(t)$
$u_1(t)$
1
1
-1
-1
$A = \begin{bmatrix} 1.5 & 1.5 \\ -0.2 & 1.2 \end{bmatrix}$
$A = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.8 \end{bmatrix}$
$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Figure 4.20 The constraint $\|u\|_{A,\infty} \leq 1$ requires that the signal $u(t)$ lie inside a trapezoid at all times t. Three weighting matrices A are shown, together with the corresponding limits on the signal u.

The first inequality in (4.18) can be shown by replacing u^2 by the upper bound $\|u\|_\infty^2$:

$$\begin{aligned} \|u\|_{\mathrm{rms}}^2 &= \lim_{T\to\infty} \frac{1}{T}\int_0^T u(t)^2\,dt \\ &\leq \lim_{T\to\infty} \frac{1}{T}\int_0^T \|u\|_\infty^2\,dt \\ &= \|u\|_\infty^2. \end{aligned}$$

The second inequality in (4.18) follows from the Cauchy-Schwarz inequality:

$$\begin{aligned} \|u\|_{\mathrm{aa}} &= \lim_{T\to\infty} \frac{1}{T}\int_0^T 1|u(t)|\,dt \\ &\leq \lim_{T\to\infty} \left(\frac{1}{T}\int_0^T 1^2\,dt\right)^{1/2} \left(\frac{1}{T}\int_0^T |u(t)|^2\,dt\right)^{1/2} \\ &= \|u\|_{\mathrm{rms}}. \end{aligned}$$

It can also be shown that

$$\|u\|_\infty \geq \|u\|_{\mathrm{ss}\infty} \geq \|u\|_{\mathrm{rms}} \geq \|u\|_{\mathrm{aa}}.$$

Another norm inequality, that gives a lower bound for $\|u\|_{\mathrm{aa}}$, is

$$\|u\|_{\mathrm{rms}}^2 \leq \|u\|_{\mathrm{aa}}\|u\|_\infty. \tag{4.20}$$

This inequality can be understood by considering the power amplifier shown in figure 4.8, with a 1Ω load resistance. If we have $V_{cc} = \|u\|_\infty$, then the amplifier does not saturate (clip), and the voltage on the load is the input voltage, $u(t)$. The average power delivered to the load is therefore $\|u\|_{\mathrm{rms}}^2$. The average power supply current is $\|u\|_{\mathrm{aa}}$, so the average power delivered by the power supply is $\|u\|_{\mathrm{aa}} V_{cc}$, which is $\|u\|_{\mathrm{aa}} \|u\|_\infty$. Of course, the average power delivered to the load does not exceed the average power drain on the power supply (the difference is dissipated in the transistors), so we have $\|u\|_{\mathrm{rms}}^2 \leq \|u\|_{\mathrm{aa}} \|u\|_\infty$, which is (4.20).

If the signal u is close to a *switching signal*, which means that it spends most of its time near its peak value, then the values of the peak, steady-state peak, RMS and average-absolute norms will be close. The *crest factor* of a signal gives an indication of how much time a signal spends near its peak value. The crest factor is defined as the ratio of the steady-state peak value to the RMS value of a signal:

$$\mathbf{CF}(u) = \frac{\|u\|_{\mathrm{ss}\infty}}{\|u\|_{\mathrm{rms}}}.$$

Since $\|u\|_{\mathrm{ss}\infty} \geq \|u\|_{\mathrm{rms}}$, the crest factor of a signal is at least 1. The crest factor is a measure of how rapidly the amplitude distribution of the signal increases below $a = \|u\|_{\mathrm{ss}\infty}$; see figure 4.11. The two signals in figure 4.11 have crest factors of 1.15 and 5.66, respectively.

The crest factor can be combined with the upper bound in (4.20) to give a bound on how much the RMS norm exceeds the average-absolute norm:

$$\frac{\|u\|_{\mathrm{rms}}}{\|u\|_{\mathrm{aa}}} \leq \mathbf{CF}(u). \tag{4.21}$$

Notes and References

For general references on vector spaces, norms of signals, or norms in general, see the Notes and References for the next chapter. The text [WH85] by Wong and Hajek covers stochastic signals.

The bold $\mathbf{L}$ in the symbols $\mathbf{L}_1$, $\mathbf{L}_2$, and $\mathbf{L}_\infty$ stands for the mathematician H. Lebesgue.

Our Definition of Norm

Our definition 4.1 differs from the standard definition of a norm in two ways: first, we allow norms to take on the value $+\infty$; and second, we do not require $\|v\| > 0$ for nonzero v (which is called the *definiteness* property). The standard term for what we call a norm is *seminorm.* We will not need the finiteness or definiteness properties of norms; in fact, the only property of norms that we use in this book is convexity (see chapter 6). Our less formal usage of the term norm allows us to give a less technical discussion of norms of signals.

The mathematically sophisticated reader can form a standard norm from each of our seminorms by first restricting it to the subspace of all signals for which $\|u\|$ is finite, and then forming the quotient space, modulo the subspace of all signals for which $\|u\|$ is zero. This process is discussed in any mathematics text covering norms, *e.g.*, Kolmogorov and Fomin [KF75] and Aubin [AUB79]. As an example, $\|\cdot\|_{\mathrm{ss}\infty}$ is a standard norm on the vector space of equivalence classes of eventually bounded signals, where the equivalence classes consist of signals that differ by a transient, *i.e.*, signals that converge to each other as $t \to \infty$.

Some Mathematical Notes

There are signals for which the RMS value (4.1) or average-absolute value (4.2) are not defined because the limits in these expressions fail to exist: for example, $u(t) = \cos\log(1+t)$. These norms will *always* be defined if we substitute lim sup for lim in the definitions (4.1) and (4.2). With this generalized definition of the RMS and average-absolute norm, many but not all of the properties discussed in this chapter still hold. For example, with lim sup substituted for lim in the definition of the RMS norm of a vector signal (given in (4.16)), equation (4.17) need not hold.

We also note that the integral defining the power spectral density (equation (4.5)) need not exist. In this case the process has a *spectral measure.*

Spectral Factorization of Weights

Youla [YOU61] developed a spectral factorization analogous to (4.13) for transfer *matrices*, which yields an interpretation of the W-weighted RMS norm as the square root of the total average power dissipated in a passive n-port admittance $G(s)$. In [AND67], Anderson showed how this spectral factorization for transfer matrices can be computed using state-space methods, by solving an algebraic Riccati equation. See also section 7.3 of Francis [FRA87].

Chapter 5

Norms of Systems

A notion closely related to the size of a *signal* is the size of a *transfer function* or LTI system. In this chapter we explore some of the ways this notion of size of a system can be made precise.

5.1 Paradigms for System Norms

In this chapter we discuss methods of measuring the size of an LTI system with input w, output z, and transfer matrix H, shown in figure 5.1. Many commonly used norms for LTI systems can be interpreted as examples of several general methods for measuring the size of a system in terms of the norms of its input and output signals. In the next few subsections we describe these general methods.

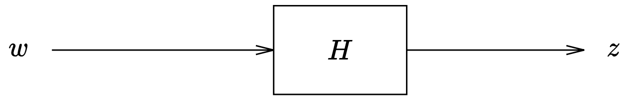


Figure 5.1 A linear system with input w and output z.

We leave as an exercise for the reader the verification that the norms we will encounter in this chapter do satisfy the required properties (*i.e.*, definition 4.1).

5.1.1 Norm of a Particular Response

The simplest general method for measuring the size of a system is to measure the size of its response to a *particular input signal* $w_{\rm part}$, *e.g.*, a unit impulse, a unit step, or a stochastic signal with a particular power spectral density. If we use the norm $\|\cdot\|_{\rm output}$ to measure the size of the response, as shown in figure 5.2, we define

$$\|H\|_{\rm part} \stackrel{\Delta}{=} \|Hw_{\rm part}\|_{\rm output}.$$

(It can be shown that this functional satisfies all of the properties required for a norm, as our notation suggests.)

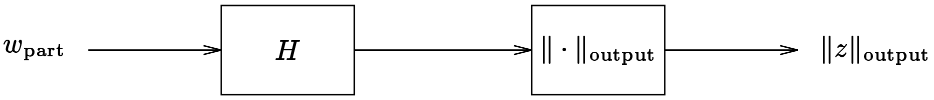


Figure 5.2 The size of a transfer matrix H can be measured by applying a particular signal w_{part}, and measuring the size of the output with some suitable signal norm $\|\cdot\|_{\mathrm{output}}$.

5.1.2 Average Response Norm

A general method for measuring the size of a system, that directly takes into account the response of the system to many input signals (and not just one particular input signal), is to measure the *average* size of the response of H to a specific *probability distribution of input signals.* If $\|\cdot\|_{\mathrm{output}}$ measures the size of the response, we define

$$\|H\|_{\mathrm{avg}} \stackrel{\Delta}{=} \mathop{\mathbf{E}}_{w} \|Hw\|_{\mathrm{output}},$$

where $\mathbf{E}_w$ denotes expectation with respect to the distribution of input signals.

5.1.3 Worst Case Response Norm

Another general method for measuring the size of a system, that takes into account the response of the system to many input signals, is to measure the worst case or largest norm of the response of H to a specific *collection of input signals.* If $\|\cdot\|_{\mathrm{output}}$ measures the size of the response, we define

$$\|H\|_{\mathrm{wc}} \stackrel{\Delta}{=} \sup_{w\in\mathcal{W}} \|Hw\|_{\mathrm{output}},$$

where $\mathcal{W}$ denotes the collection of input signals.

5.1.4 Gain of a System

An important special case of a worst case norm is a *gain*, defined as the largest ratio of the norm of the output to the norm of the input. If $\|\cdot\|$ is used to measure the size of both the input and output signals, we define

$$\|H\|_{\mathrm{gn}} \stackrel{\Delta}{=} \sup_{\|w\|\neq 0} \frac{\|Hw\|}{\|w\|}. \qquad (5.1)$$

The gain $\|H\|_{\rm gn}$ is therefore the maximum factor by which the system can scale the size (measured by the norm $\|\cdot\|$) of a signal flowing through it. The gain can also be expressed as a worst case response norm:

$$\|H\|_{\rm gn} = \sup_{\|w\|\leq 1} \|Hw\|.$$

If the transfer matrix H is not square, we cannot really use the same norm to measure the input and output signals, since they have different numbers of components. In such cases we rely on our naming conventions to identify the "same" norm to be used for the input and the output. For example, the RMS gain of a 2×3 transfer matrix is defined by (5.1), where the norm in the numerator is the RMS norm of a vector signal with 2 components, and the norm in the denominator is the RMS norm of a vector signal with 3 components. It is also possible to define a more general gain with different types of norms on the input and output, but we will not use this generalization.

5.2 Norms of SISO LTI Systems

In this section we describe various norms for single-input, single-output (SISO) systems.

5.2.1 Peak-Step

Our first example of a norm of a system is from the first paradigm: the size of its response to a particular input. The particular input signal is a unit step; we measure the size of the response by its peak norm. We define:

$$\|H\|_{\rm pk\_step} \stackrel{\Delta}{=} \|s\|_\infty$$

where $s(t)$ denotes the step response of H; we will refer to $\|H\|_{\rm pk\_step}$ as the *peak-step* norm of H. This would be an appropriate measure if, say, w represents a set-point command in a control system (a signal that might be expected to change values only occasionally), and z represents some actuator signal, say, a motor voltage. In this case, $\|H\|_{\rm pk\_step}$ (multiplied by the maximum possible changes in the set-point command) would represent a good approximation of the maximum motor voltage (due to set-point changes) that might be encountered in the operation of the system.

5.2.2 RMS Response to a Particular Noise Input

A common measure of the size of a transfer function is the RMS value of its output when its input is some particular stationary stochastic process. Suppose that the particular input w has power spectral density $S_w(\omega)$, and H is stable, meaning that all of its poles have negative real part. The power spectral density of the output z of H is then

$$S_z(\omega) = S_w(\omega)|H(j\omega)|^2,$$

and therefore

$$\|z\|_{\mathrm{rms}} = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 S_w(\omega)\, d\omega \right)^{1/2}. \tag{5.2}$$

Thus we assign to H the norm

$$\|H\|_{\mathrm{rms},w} \triangleq \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 S_w(\omega)\, d\omega \right)^{1/2}. \tag{5.3}$$

The right-hand side of (5.3) has the same form as (4.12), with H substituted for W. The interpretations are different, however: in (5.3), w is some fixed signal, and we are measuring the size of the *LTI system* H, whereas in (4.12), W is a fixed weighting transfer function, and we are measuring the size of the *signal* w.

5.2.3 $\mathbf{H}_2$ Norm: RMS Response to White Noise

Consider the RMS response norm above. If $S_w(\omega) \approx 1$ at those frequencies where $|H(j\omega)|$ is significant, then we have

$$\|H\|_{\mathrm{rms},w} \approx \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2\, d\omega \right)^{1/2}.$$

It is convenient to think of such a signal as an approximation of a *white noise* signal, a fictitious input signal with $S_w(\omega) = 1$ for all ω (and thus, infinite power, which we conveniently overlook).

This important norm of a stable system is denoted

$$\|H\|_2 \triangleq \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2\, d\omega \right)^{1/2}$$

(we assign $\|H\|_2 = \infty$ for unstable H), and referred to as the $\mathbf{H}_2$ norm of H.

Thus we have the important fact: *the* $\mathbf{H}_2$ *norm of a transfer function measures the RMS response of its output when it is driven by a white noise excitation.*

The $\mathbf{H}_2$ norm can be given another interpretation. By the Parseval theorem,

$$\|H\|_2 = \left(\int_0^{\infty} h(t)^2\, dt \right)^{1/2} = \|h\|_2,$$

the $\mathbf{L}_2$ norm of the *impulse response* h of the LTI system. Thus we can interpret the $\mathbf{H}_2$ norm of a system as the $\mathbf{L}_2$ norm of its response to the particular input signal δ, a unit impulse.

5.2.4 A Worst Case Response Norm

Let us give an example of measuring the size of a transfer function using the worst case response paradigm. Suppose that not much is known about w except that $\|w\|_\infty \le M_{\mathrm{ampl}}$ and $\|\dot{w}\|_\infty \le M_{\mathrm{slew}}$, *i.e.*, w is bounded by M_{ampl} and slew-rate limited by M_{slew}. If the peak of the output z is critical, a reasonable measure of the size of H is

$$\|H\|_{\mathrm{wc}} \overset{\Delta}{=} \sup\left\{\|Hw\|_\infty \;\middle|\; \|w\|_\infty \le M_{\mathrm{ampl}}, \|\dot{w}\|_\infty \le M_{\mathrm{slew}}\right\}.$$

In other words, $\|H\|_{\mathrm{wc}}$ is the worst case (largest) peak of the output, over all inputs bounded by M_{ampl} and slew-rate limited by M_{slew}.

5.2.5 Peak Gain

The peak gain of an LTI system is

$$\|H\|_{\mathrm{pk\_gn}} \overset{\Delta}{=} \sup_{\|w\|_\infty \ne 0} \frac{\|Hw\|_\infty}{\|w\|_\infty}. \tag{5.4}$$

It can be shown that the peak gain of a transfer function is equal to the $\mathbf{L}_1$ norm of its impulse response:

$$\|H\|_{\mathrm{pk\_gn}} = \int_0^\infty |h(t)|\,dt = \|h\|_1. \tag{5.5}$$

The peak gain of a transfer function is finite if and only if the transfer function is stable.

To establish (5.5) we consider the input signal

$$w(t) = \begin{cases} \mathrm{sgn}(h(T-t)) & \text{for } 0 \le t \le T \\ 0 & \text{otherwise,} \end{cases} \tag{5.6}$$

which has $\|w\|_\infty = 1$ (the sign function, $\mathrm{sgn}(\cdot)$, has the value 1 for positive arguments, and -1 for negative arguments). The output at time T is

$$\begin{aligned} z(T) &= \int_0^T w(T-t)h(t)\,dt \\ &= \int_0^T \mathrm{sgn}(h(t))h(t)\,dt \\ &= \int_0^T |h(t)|\,dt, \end{aligned}$$

which converges to $\|h\|_1$ as $T \to \infty$. So for large T (and H stable, so that $\|H\|_{\mathrm{pk\_gn}} < \infty$), the signal (5.6) yields $\|z\|_\infty/\|w\|_\infty$ near $\|h\|_1$; it is also possible to show that there is a signal w such that $\|z\|_\infty/\|w\|_\infty = \|h\|_1$.

The peak gain of a system can also be expressed in terms of its step response s:

$$\|H\|_{\mathrm{pk\_gn}} = \mathbf{Tv}(s),$$

where $\mathbf{Tv}(f)$, the *total variation* of a function f, is defined by

$$\mathbf{Tv}(f) \stackrel{\Delta}{=} \sup_{0 \le t_1 \le \cdots \le t_N} \sum_{i=1}^{N-1} |f(t_i) - f(t_{i+1})| \,.$$

Roughly speaking, $\mathbf{Tv}(f)$ is the sum of all consecutive peak-to-valley differences in f; this is shown in figure 5.3.

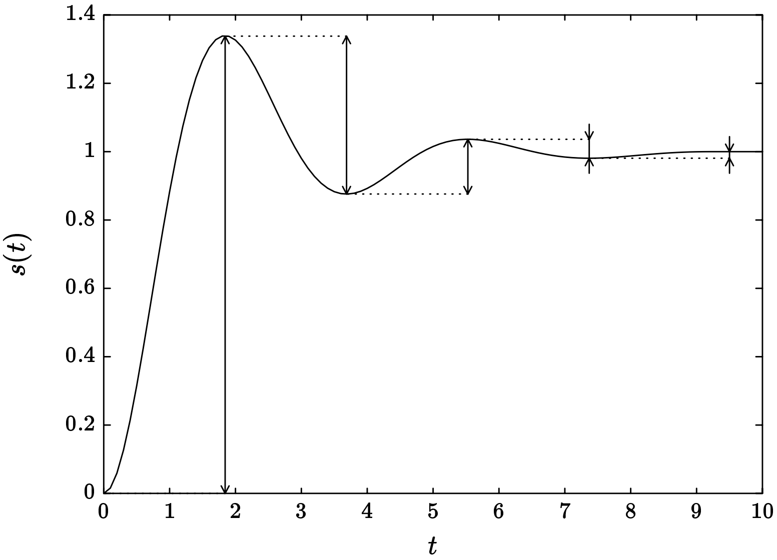


Figure 5.3 The peak gain of a transfer function is equal to the total variation of its step response s, *i.e.*, the sum of all the consecutive peak-to-valley differences (shown as arrows) of s.

It turns out that the peak gain of a SISO transfer function is also the average-absolute gain:

$$\|H\|_{\mathrm{pk\_gn}} = \|H\|_{\mathrm{aa\_gn}} \stackrel{\Delta}{=} \sup_{\|w\|_{\mathrm{aa}} \neq 0} \frac{\|Hw\|_{\mathrm{aa}}}{\|w\|_{\mathrm{aa}}}. \tag{5.7}$$

5.2.6 $\mathbf{H}_\infty$ Norm: RMS Gain

An important norm of a transfer function is its RMS gain:

$$\|H\|_{\mathrm{rms\_gn}} \stackrel{\Delta}{=} \sup_{\|w\|_{\mathrm{rms}} \neq 0} \frac{\|Hw\|_{\mathrm{rms}}}{\|w\|_{\mathrm{rms}}}. \tag{5.8}$$

The RMS gain of a transfer function turns out to coincide with its $\mathbf{L}_2$ gain,

$$\|H\|_{\mathrm{rms\_gn}} = \sup_{\|w\|_2 \neq 0} \frac{\|Hw\|_2}{\|w\|_2}, \tag{5.9}$$

and is equal to the maximum magnitude of the transfer function,

$$\|H\|_{\mathrm{rms\_gn}} = \sup_{\omega} |H(j\omega)|, \tag{5.10}$$

when H is stable; for unstable H we have $\|H\|_{\mathrm{rms\_gn}} = \infty$. For this reason, the RMS gain is sometimes called the *maximum magnitude norm* or the *Chebychev norm* of a transfer function. We note that the right-hand side of (5.10) can be interpreted as a worst case response norm of H: it is the largest steady-state peak of the response of H to any unit amplitude sinusoid ($w(t) = \cos \omega t$).

Equations (5.8–5.10) show that four reasonable interpretations of "the transfer function H is small" coincide:

- the RMS value of its output is always small compared to the RMS value of its input;
- the total energy of its output is always small compared to the total energy of its input;
- the transfer function $H(j\omega)$ has a small magnitude at all frequencies;
- the steady-state peak of the response to a unit amplitude sinusoid of any frequency is small.

The RMS gain of a transfer function H can be expressed as its maximum magnitude in the right half of the complex plane:

$$\|H\|_{\mathrm{rms\_gn}} = \|H\|_\infty \stackrel{\Delta}{=} \sup_{\Re s > 0} |H(s)|, \tag{5.11}$$

which is called the $\mathbf{H}_\infty$ norm of H. (Note the very different meaning from $\|w\|_\infty$, the $\mathbf{L}_\infty$ norm of a signal.)

Let us establish (5.10) in the case where w is stochastic and H is stable; it is not hard to establish it in general. Let $S_w(\omega)$ denote the power spectral density of w. The power spectral density of the output z is then $S_z(\omega) = |H(j\omega)|^2 S_w(\omega)$, and therefore

$$\begin{aligned}
\|z\|_{\mathrm{rms}}^2 &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_z(\omega)\, d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} |H(j\omega)|^2 S_w(\omega)\, d\omega \\
&\leq \sup_{\omega} |H(j\omega)|^2 \frac{1}{2\pi}\int_{-\infty}^{\infty} S_w(\omega)\, d\omega \\
&= \|H\|_\infty^2 \|w\|_{\mathrm{rms}}^2.
\end{aligned}$$

Thus we have for all w with nonzero RMS value

$$\frac{\|Hw\|_{\mathrm{rms}}}{\|w\|_{\mathrm{rms}}} \le \|H\|_\infty,$$

which shows that $\|H\|_{\mathrm{rms\_gn}} \le \|H\|_\infty$. By concentrating the power spectral density of w near a frequency $\omega_{\max}$ at which $|H(j\omega_{\max})| \approx \|H\|_\infty$, we have

$$\frac{\|Hw\|_{\mathrm{rms}}}{\|w\|_{\mathrm{rms}}} \approx \|H\|_\infty.$$

(Making this argument precise establishes (5.10).)

We can contrast the $\mathbf{H}_\infty$ and $\mathbf{H}_2$ norms by considering the associated inequality specifications. Equation (5.11) implies that the $\mathbf{H}_\infty$ norm-bound specification

$$\|H\|_\infty \le M \tag{5.12}$$

is equivalent to the specification:

$$\|Hw\|_{\mathrm{rms}} \le M \quad \text{for all } w \text{ with } \ \|w\|_{\mathrm{rms}} \le 1. \tag{5.13}$$

In contrast, the $\mathbf{H}_2$ norm-bound specification

$$\|H\|_2 \le M \tag{5.14}$$

is equivalent to the specification

$$\|Hw\|_{\mathrm{rms}} \le M \quad \text{for } w \text{ a white noise.} \tag{5.15}$$

The $\mathbf{H}_\infty$ norm is often combined with a frequency domain weight:

$$\|H\|_{W,\infty} = \|WH\|_\infty.$$

If the weighting transfer function W and its inverse W^{-1} are both stable, the specification $\|H\|_{W,\infty} \le 1$ can be expressed in the more classical form: H is stable and

$$|H(j\omega)| \le |W(j\omega)|^{-1} \quad \text{for all } \ \omega,$$

depicted in figure 5.4(b).

5.2.7 Shifted $\mathbf{H}_\infty$ Norm

A useful generalization of the $\mathbf{H}_\infty$ norm of a transfer function is its a-exponentially weighted $\mathbf{L}_2$ norm gain, defined by

$$\|H\|_{\infty,a} \stackrel{\Delta}{=} \sup_{\|\tilde{w}\|_2 \ne 0} \frac{\|\tilde{z}\|_2}{\|\tilde{w}\|_2},$$

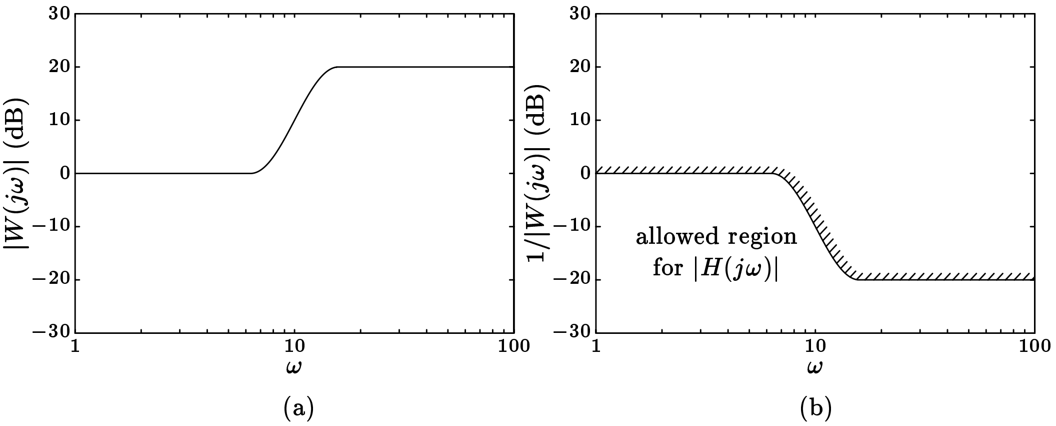


Figure 5.4 An example of a frequency domain weight W that enhances frequencies above $\omega = 10$ is shown in (a). The specification $\|WH\|_\infty \le 1$ requires that the magnitude of $H(j\omega)$ lie below the curve $1/|W(j\omega)|$, as shown in (b). In particular, $|H(j\omega)|$ must be below -20dB for $\omega \ge 12$.

where $\tilde{w}(t) = e^{at}w(t)$, $\tilde{z}(t) = e^{at}z(t)$, and $z = Hw$. It can be shown that

$$\|H\|_{\infty,a} = \sup_{\Re s > -a} |H(s)| = \|H_a\|_\infty, \tag{5.16}$$

where $H_a(s) = H(s-a)$. H_a is called the a-shifted transfer function formed from H, and the norm $\|\cdot\|_{\infty,a}$ is called the a-shifted $\mathbf{H}_\infty$ norm of a transfer function. This is shown in figure 5.5.

From (5.16), we see that the a-shifted $\mathbf{H}_\infty$ norm of a transfer function is finite if and only if the real parts of the poles of H are less than $-a$. For $a < 0$, then, the shifted $\mathbf{H}_\infty$ norm can be used to measure the size of some unstable transfer functions, for example,

$$\|1/(s-1)\|_{\infty,-2} = 1$$

(whereas $\|1/(s-1)\|_\infty = \infty$, meaning that its RMS gain is infinite). On the other hand if $a > 0$, the a-shifted $\mathbf{H}_\infty$ norm-bound specification

$$\|H\|_{\infty,a} \le M$$

(or even just $\|H\|_{\infty,a} < \infty$) guarantees that the poles of H lie to the left of the line $\Re s = -a$ in the complex plane.

We can interpret the shifted transfer function $H(s-a)$ as follows. Given a block diagram for H that consists of integrators (transfer function $1/s$), summing blocks, and scaling amplifiers, we replace each integrator with a transfer function $1/(s-a)$ (called a "leaky integrator" when $a < 0$). The result is a block diagram of $H(s-a)$, as shown in figure 5.6. In circuit theory, where H is some network function, this is called a *uniform loading* of H.

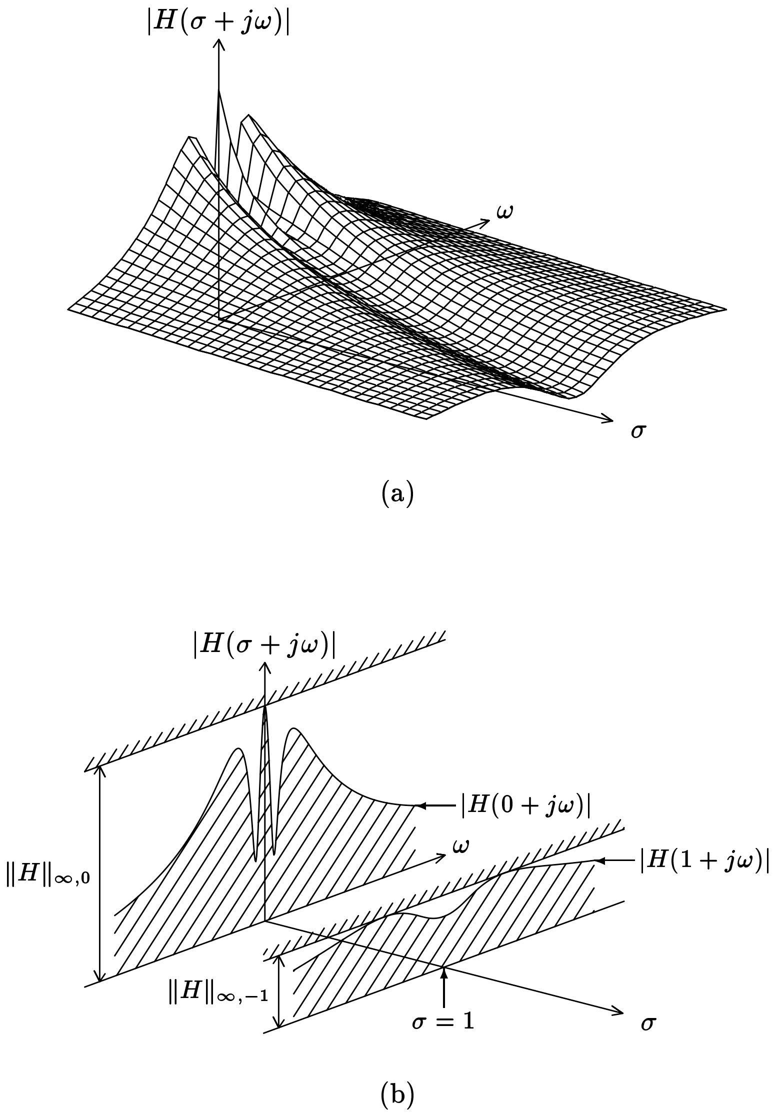


Figure 5.5 The magnitude of a transfer function H is shown in (a). The $\mathbf{L}_2$ gain of the system is the peak magnitude of H along the line $s = j\omega$. The exponentially weighted $(a = -1)$ $\mathbf{L}_2$ gain of the system is the peak magnitude of H along the line $s = 1 + j\omega$, as shown in (b).

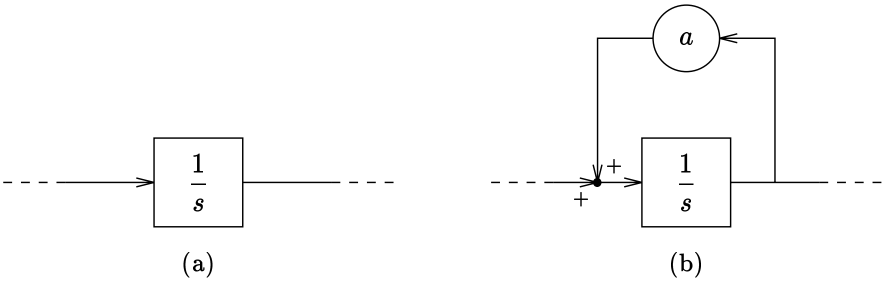


Figure 5.6 A realization of $H(s-a)$ can be formed by taking each integrator in a realization of $H(s)$, shown in (a), and adding a feedback of a, as shown in (b).

5.2.8 Hankel Norm

The *Hankel norm* of a transfer function is a measure of the effect of its past input on its future output, or the amount of energy that can be stored in and then retrieved from the system. It is given by

$$\|H\|_{\mathrm{hankel}} \\ \triangleq \sup \left\{ \left(\int_T^\infty z(t)^2\,dt \right)^{1/2} \;\middle|\; \int_0^T w(t)^2\,dt \le 1,\ w(t)=0,\ t>T\ge 0 \right\}.$$

We can think of w in this definition as an excitation that acts over the time period $0 \le t \le T$; the response or *ring* of the system after the excitation has stopped is $z(t)$ for $t > T$. An example of a past excitation and the resulting ring is shown in figure 5.7.

It is useful to think of the map from the excitation ($w(t)$ for $0 \le t \le T$) to ring ($z(t)$ for $t \ge T$) as consisting of two parts: first, the mapping of the excitation into the *state* of the system at $t = T$; and then, the mapping from the state of the system at $t = T$ (which "summarizes" the total effect on the future output that the excitation can have) into the output for $t \ge T$. This interpretation will come up again when we describe a method for computing $\|H\|_{\mathrm{hankel}}$.

5.2.9 Example 1: Comparing Two Transfer Functions

In this section we will consider various norms of the two transfer functions

$$H_{13}^{(\mathrm{a})}(s) = \frac{-44.1s^3+334s^2+1034s+390}{s^6+20s^5+155s^4+586s^3+1115s^2+1034s+390}, \qquad (5.17)$$

$$H_{13}^{(\mathrm{b})}(s) = \frac{-220s^3+222s^2+19015s+7245}{s^6+29.1s^5+297s^4+1805s^3+9882s^2+19015s+7245}. \qquad (5.18)$$

These transfer functions are the I/O transfer functions (T) achieved by the controllers $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$ in the standard plant example of section 2.4. The step

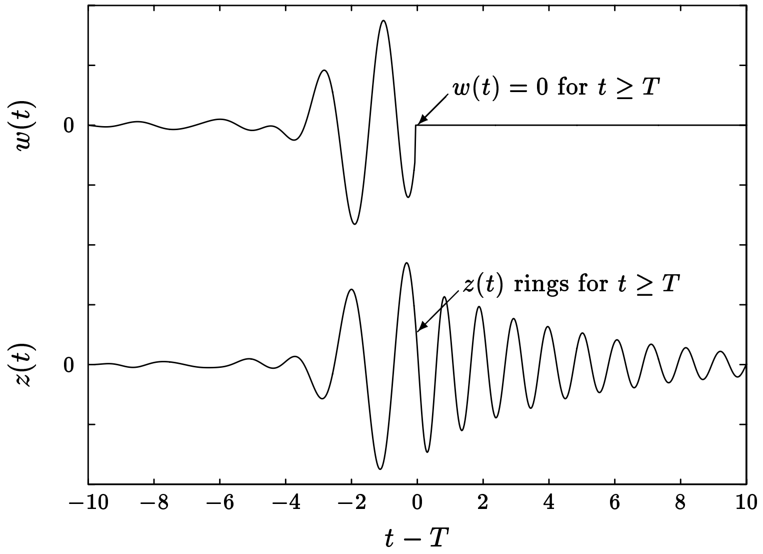


Figure 5.7 The Hankel norm of a transfer function is the largest possible square root energy in the output z for $t \geq T$, given a unit-energy excitation w that stops at $t = T$.

responses of $H_{13}^{(a)}$ and $H_{13}^{(b)}$ are shown in figure 5.8 and their frequency response magnitudes in figure 5.9. The values of various norms of $H_{13}^{(a)}$ and $H_{13}^{(b)}$ are shown in table 5.1.

From the first row of table 5.1 we see that the peak of the response of $H_{13}^{(a)}$ to a step input is about the same as $H_{13}^{(b)}$. Thus, in the sense of peak step response, $H_{13}^{(a)}$ is about the same size as $H_{13}^{(b)}$.

If $H_{13}^{(a)}$ and $H_{13}^{(b)}$ are driven by white noise, the RMS value of the output of $H_{13}^{(a)}$ is less than half that of $H_{13}^{(b)}$ (second row). Figure 5.10 shows an example of the

Norm	$H_{13}^{(a)}$	$H_{13}^{(b)}$	figure
$\lVert\cdot\rVert_{\mathrm{pk\_step}}$	1.36	1.40	5.8
$\lVert\cdot\rVert_2$	1.17	2.69	5.10
$\lVert\cdot\rVert_{\mathrm{wc}}$	1.60	1.68	5.11
$\lVert\cdot\rVert_{\mathrm{pk\_gn}}$	1.74	4.93	5.12
$\lVert\cdot\rVert_\infty$	1.47	3.72	5.9
$\lVert\cdot\rVert_{\mathrm{hankel}}$	1.07	2.04	5.13

Table 5.1 The values of six different norms of $H_{13}^{(a)}$ and $H_{13}^{(b)}$.

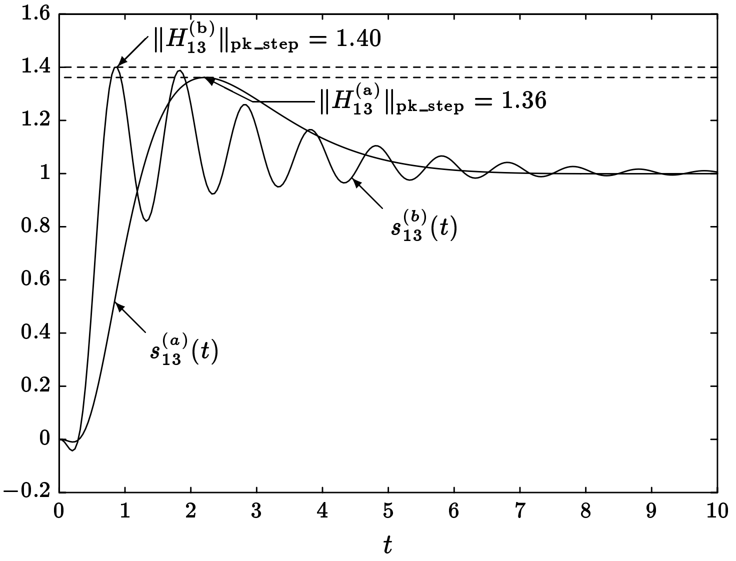


Figure 5.8 The step responses of the transfer functions in (5.17) and (5.18). Note that $\|H_{13}^{(\mathrm{a})}\|_{\mathrm{pk\_step}} = 1.36$, and $\|H_{13}^{(\mathrm{b})}\|_{\mathrm{pk\_step}} = 1.40$.

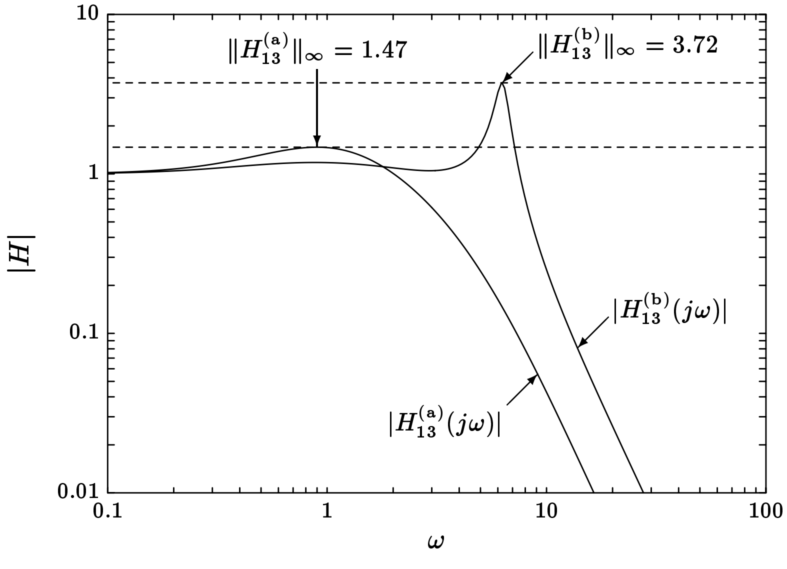


Figure 5.9 The magnitudes of the transfer functions in (5.17) and (5.18). Note that $\|H_{13}^{(\mathrm{a})}\|_\infty = 1.47$, and $\|H_{13}^{(\mathrm{b})}\|_\infty = 3.72$.

outputs of $H_{13}^{(\mathrm{a})}$ and $H_{13}^{(\mathrm{b})}$ with a white noise excitation. Thus, in the sense of RMS response to white noise, we can say that $H_{13}^{(\mathrm{a})}$ is about half the size of $H_{13}^{(\mathrm{b})}$.

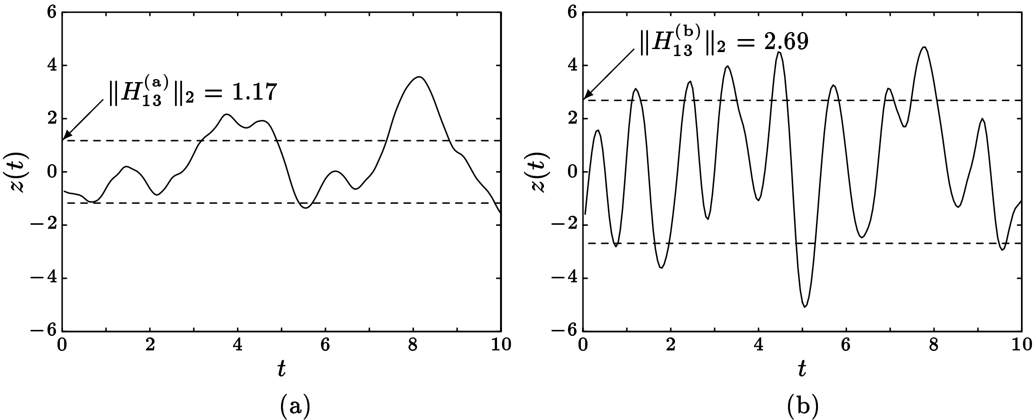


Figure 5.10 (a) shows a sample of the steady-state response of $H_{13}^{(\mathrm{a})}$ to a white noise excitation, together with the value $\|H_{13}^{(\mathrm{a})}\|_2 = 1.17$. (b) shows a sample of the steady-state response of $H_{13}^{(\mathrm{b})}$ to a white noise excitation, together with the value $\|H_{13}^{(\mathrm{b})}\|_2 = 2.69$.

From the third row, we see that the worst case response of $H_{13}^{(\mathrm{a})}$ to inputs bounded and slew-rate limited by 1 is similar to that of $H_{13}^{(\mathrm{b})}$. Amplitude and slew-rate limited input waveforms that produce outputs with peak values close to these worst case values are shown in figure 5.11. Thus, in the sense of maximum peak output in response to inputs bounded and slew limited by 1, $H_{13}^{(\mathrm{a})}$ is about the same size as $H_{13}^{(\mathrm{b})}$.

From the fourth row, we see that the peak output of $H_{13}^{(\mathrm{b})}$ with a worst case input bounded by 1 is almost three times larger than $H_{13}^{(\mathrm{a})}$. This is expected from the step response total variation expression for the peak gain (see figures 5.3 and 5.8). Input waveforms that produce outputs close to these worst case values are shown in figure 5.12. Thus, in the sense of maximum peak output in response to inputs bounded by 1, $H_{13}^{(\mathrm{a})}$ is less than one third the size of $H_{13}^{(\mathrm{b})}$.

From the fifth row, we see that the RMS gain of $H_{13}^{(\mathrm{b})}$ is more than twice as large as the RMS gain of $H_{13}^{(\mathrm{a})}$. This can be seen from figure 5.9; input signals that result in the largest possible ratio of RMS response to RMS input are sinusoids at frequencies $\omega = 0.9$ (for $H_{13}^{(\mathrm{a})}$) and $\omega = 6.3$ (for $H_{13}^{(\mathrm{b})}$).

Finally, the worst case square root energy in the output of $H_{13}^{(\mathrm{a})}$ after its unit-energy input signal is turned off is about 48% lower than the worst case for $H_{13}^{(\mathrm{b})}$ (sixth row of table 5.1). Thus, in the sense of square root energy gain from past inputs to future outputs we can say that $H_{13}^{(\mathrm{a})}$ is about half the size of $H_{13}^{(\mathrm{b})}$. Fig-

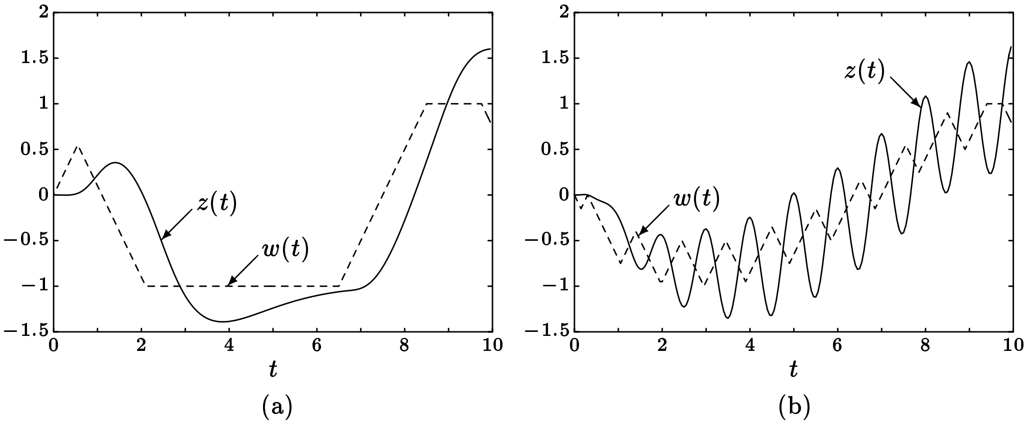


Figure 5.11 (a) shows an input signal w with $\|w\|_\infty = 1$ and $\|\dot{w}\|_\infty = 1$ that drives the output of $H_{13}^{(\mathrm{a})}$ close to $\|H_{13}^{(\mathrm{a})}\|_{\mathrm{wc}} = 1.60$. (b) shows an input signal w with $\|w\|_\infty = 1$ and $\|\dot{w}\|_\infty = 1$ that drives the output of $H_{13}^{(\mathrm{b})}$ close to $\|H_{13}^{(\mathrm{b})}\|_{\mathrm{wc}} = 1.68$.

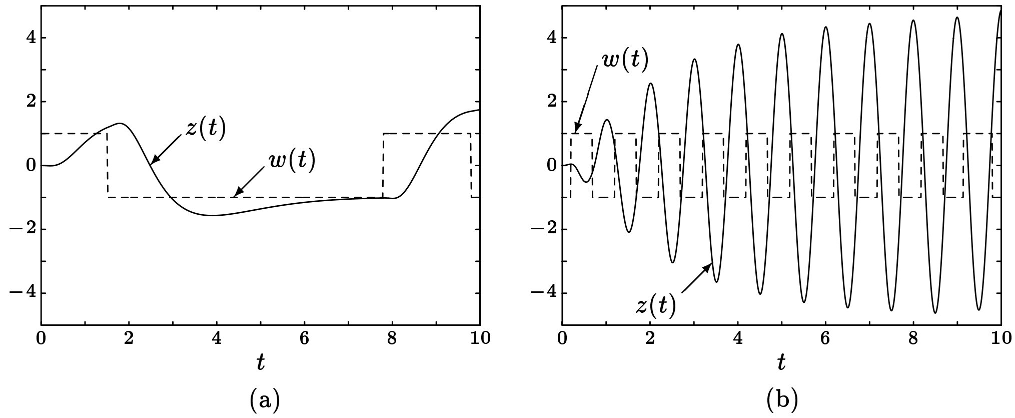


Figure 5.12 (a) shows an input signal w with $\|w\|_\infty = 1$, together with the output z produced when $H_{13}^{(\mathrm{a})}$ is driven by w; $z(10) = 1.74$ is very close to $\|H_{13}^{(\mathrm{a})}\|_{\mathrm{pk\_gn}}$. (b) shows an input signal w with $\|w\|_\infty = 1$, together with the output z produced when $H_{13}^{(\mathrm{b})}$ is driven by w; $z(10) = 4.86$ is close to $\|H_{13}^{(\mathrm{b})}\|_{\mathrm{pk\_gn}} = 4.93$.

ure 5.13 shows unit-energy excitations for $t \leq 5$ that produce square root output energies for $t \geq 5$ close to the Hankel norms of $H_{13}^{(a)}$ and $H_{13}^{(b)}$.

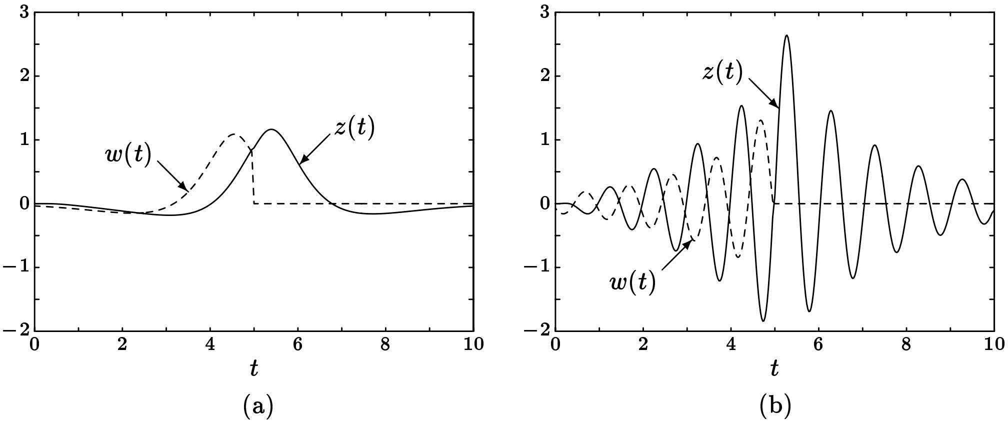


Figure 5.13 (a) shows a unit-energy input signal w that is zero for $t \geq 5$, together with the output z when $H_{13}^{(a)}$ is driven by w. The square root energy in z for $t \geq 5$ is close to $\|H_{13}^{(a)}\|_{\rm hankel} = 1.07$. (b) shows a unit-energy input signal w that is zero for $t \geq 5$, together with the output z when $H_{13}^{(b)}$ is driven by w. The square root energy in z for $t \geq 5$ is close to $\|H_{13}^{(b)}\|_{\rm hankel} = 2.04$.

5.2.10 Example 2: the Gain of an Amplifier Circuit

Consider the band pass filter circuit shown in figure 5.14. We will assume that the opamp saturates at ±14V. The input of the circuit (which is produced by another opamp) is no larger than ±14V (*i.e.*, $\|w\|_\infty \leq 14$). We ask the question: can this filter saturate?

Assuming the opamp does not saturate, the transfer function from w to z is

$$H(s) = \frac{-2s/10^4}{(s/10^4+1)^2} \tag{5.19}$$

The maximum magnitude of this transfer function is 1.0 ($\|H\|_\infty = 1$), so, provided the opamp does not saturate, the RMS value of the filter output does not exceed the RMS value of the filter input. It is tempting to conclude that the opamp in the filter will not saturate.

This conclusion is wrong, however. The peak gain of the transfer function H is $\|H\|_{\rm pk\_gn} = 1.47$, so there are inputs bounded by, say, ±10V that will drive the opamp into saturation. Figure 5.15 gives an example of such an input signal, and the corresponding output that would be produced if the opamp did not saturate. Since it exceeds 14V, the real filter will saturate with this input signal.

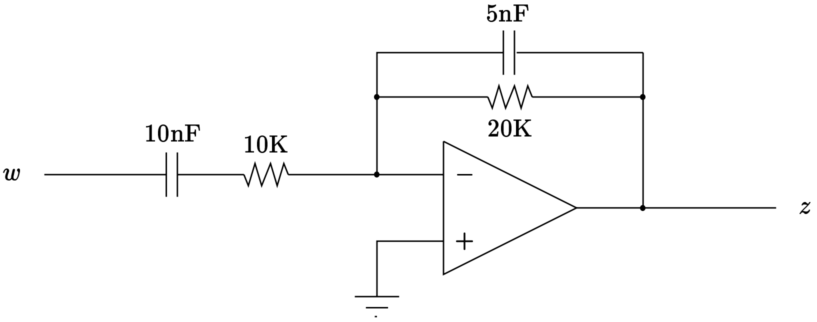


Figure 5.14 A bandpass filter circuit. The amplifier has very large open-loop gain. The output clips at ±14V, and the input lies between ±14V. When the circuit is operating linearly the transfer function from w to z is given by (5.19).

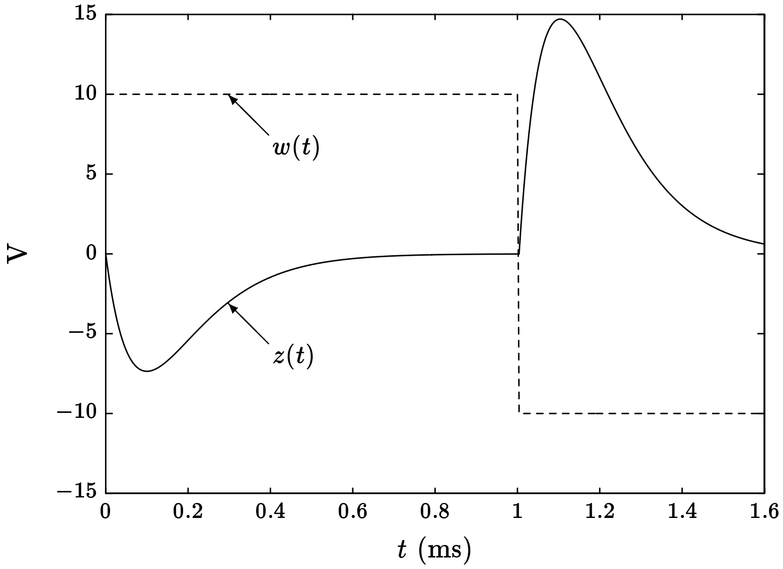


Figure 5.15 If the circuit shown in figure 5.14 did not saturate, the input w shown would produce the output z. Even though $\|w\|_\infty = 10\text{V}$, we have $\|z\|_\infty = 14.7\text{V}$. Thus the input w will drive the real circuit in figure 5.14 into saturation. Of course, $\|z\|_{\text{rms}} \le \|w\|_{\text{rms}}$, since $\|H\|_{\text{rms\_gn}} = 1$.

5.3 Norms of MIMO LTI Systems

Some of the common norms for multiple-input, multiple-output (MIMO) LTI systems can be expressed in terms of the *singular values* of the $n_z \times n_w$ transfer matrix H, which, roughly speaking, give information analogous to the magnitude of a SISO transfer function. The singular values of a matrix $M \in \mathbf{C}^{n_z \times n_w}$ are defined by

$$\sigma_i(M) \triangleq \left(\lambda_i(M^*M)\right)^{1/2}, \qquad i = 1, \ldots, \min\{n_z, n_w\}, \tag{5.20}$$

where $\lambda_i(\cdot)$ denotes the ith largest eigenvalue. The largest singular value (*i.e.*, σ_1) is also denoted $\sigma_{\max}$. A plot of $\sigma_i(H(j\omega))$ is called a *singular value plot*, and is analogous to a Bode magnitude plot of a SISO transfer function (an example is given in figure 5.17).

5.3.1 RMS Response to a Particular Noise Input

Suppose H is stable (*i.e.*, each of its entries is stable), and S_w is the power spectral density matrix of w. Then

$$\|H\|_{\mathrm{rms},w} = \left(\mathbf{Tr}\,\frac{1}{2\pi}\int_{-\infty}^{\infty} H(j\omega)S_w(j\omega)H(j\omega)^*\,d\omega\right)^{1/2}.$$

5.3.2 $\mathbf{H}_2$ Norm: RMS Response to White Noise

If $S_w(\omega) \approx I$ for those frequencies for which $H(j\omega)$ is significant, then this norm is approximately the $\mathbf{H}_2$ norm of a MIMO system:

$$\|H\|_{\mathrm{rms},w} \approx \|H\|_2 \triangleq \left(\mathbf{Tr}\,\frac{1}{2\pi}\int_{-\infty}^{\infty} H(j\omega)H(j\omega)^*\,d\omega\right)^{1/2}. \tag{5.21}$$

The $\mathbf{H}_2$ norm of H is therefore the RMS value of the output when the inputs are driven by independent white noises.

By Parseval's theorem, the $\mathbf{H}_2$ norm can be expressed as

$$\begin{aligned}\|H\|_2 &= \left(\mathbf{Tr}\int_0^{\infty} h(t)h(t)^T\,dt\right)^{1/2}\\ &= \left(\sum_{i=1}^{n_z}\sum_{k=1}^{n_w}\|H_{ik}\|_2^2\right)^{1/2},\end{aligned}$$

where h is the impulse matrix of H. Thus, the $\mathbf{H}_2$ norm of a transfer matrix H is the square root of the sum of the squares of the $\mathbf{H}_2$ norms of its entries.

The $\mathbf{H}_2$ norm can also be expressed in terms of the singular values of the transfer matrix H:

$$\|H\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{i=1}^{n} \sigma_i(H(j\omega))^2 \, d\omega \right)^{1/2}, \tag{5.22}$$

where $n = \min\{n_z, n_w\}$. Thus, the square of the $\mathbf{H}_2$ norm is the total area under the squared singular value plots, on a linear frequency scale. This is shown in figure 5.16.

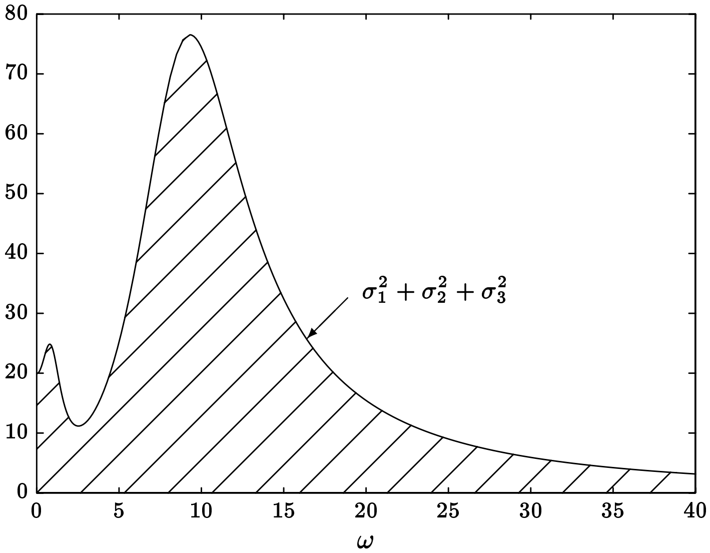


Figure 5.16 For a MIMO transfer matrix, $\|H\|_2^2$ is proportional to the area under a plot of the sum of the squares of the singular values of H (shown here for $\min\{n_z, n_w\} = 3$).

5.3.3 Peak Gain

The peak gain of a MIMO system is

$$\|H\|_{\rm pk\_gn} \stackrel{\Delta}{=} \sup_{\|w\|_\infty \neq 0} \frac{\|Hw\|_\infty}{\|w\|_\infty} = \max_{1 \le i \le n_z} \int_0^\infty \sum_{j=1}^{n_w} |h_{ij}(t)| \, dt. \tag{5.23}$$

For MIMO systems, the peak gain is not the same as the average-absolute gain (*c.f.* (5.7)). The average-absolute gain is

$$\|H\|_{\rm aa\_gn} \stackrel{\Delta}{=} \sup_{\|w\|_{\rm aa} \neq 0} \frac{\|Hw\|_{\rm aa}}{\|w\|_{\rm aa}} = \|H^T\|_{\rm pk\_gn},$$

the peak gain of the LTI system whose transfer matrix is the transpose of H.

5.3.4 RMS Gain

The RMS gain of a MIMO system is important for several reasons, one being that it is readily computed from state-space equations. The RMS gain of a MIMO transfer matrix is

$$\|H\|_{\mathrm{rms\_gn}} = \|H\|_\infty \stackrel{\Delta}{=} \sup_{\Re s>0} \sigma_{\max}(H(s)),$$

the $\mathbf{H}_\infty$ norm of a transfer matrix (*c.f.* (5.11), the analogous definition for transfer functions). Thus, $\|H\|_\infty < \infty$ if and only if the transfer matrix H is stable. For stable H, we can express the $\mathbf{H}_\infty$ norm as the maximum of the maximum singular value over all frequencies:

$$\|H\|_\infty = \sup_\omega \sigma_{\max}(H(j\omega)),$$

as shown in figure 5.17. Note that the other singular values do not affect $\|H\|_\infty$.

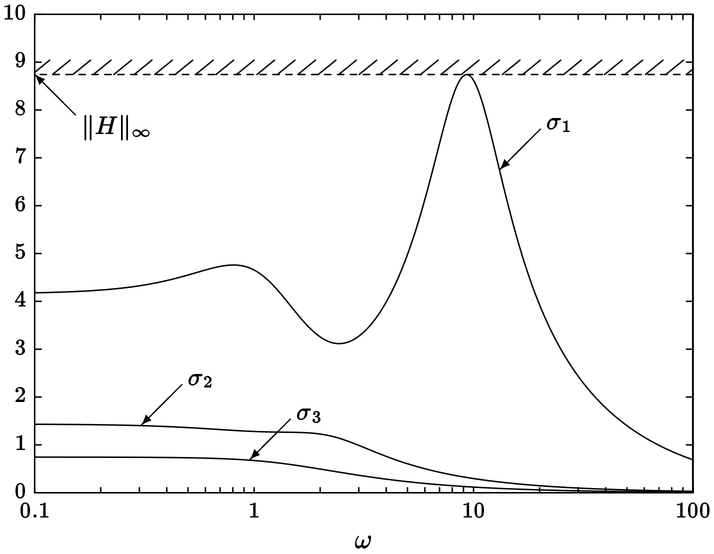


Figure 5.17 The $\mathbf{H}_\infty$ norm of a stable transfer matrix H is the maximum over frequency of the maximum singular value, σ_1. The other singular values, $\sigma_2, \ldots, \sigma_n$, do not affect the $\mathbf{H}_\infty$ norm.

5.3.5 Entropy of a System

In this section we describe a measure of the size of a MIMO system, which is *not* a norm, but is closely related to the $\mathbf{H}_2$ norm and the $\mathbf{H}_\infty$ norm. For $\gamma > 0$ we define the γ-*entropy* of the system with transfer matrix H as

$$I_\gamma(H) \triangleq \begin{cases} -\dfrac{\gamma^2}{2\pi}\displaystyle\int_{-\infty}^{\infty} \log\det\left(I - \gamma^{-2}H(j\omega)H(j\omega)^*\right) d\omega & \text{if } \|H\|_\infty < \gamma \\ \infty & \text{if } \|H\|_\infty \geq \gamma \end{cases}$$

(*c.f.* (5.21)). The γ-entropy can also be expressed in terms of the singular values as

$$I_\gamma(H) = \begin{cases} \dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{\infty} \sum_{i=1}^{n} -\gamma^2 \log\left(1 - (\sigma_i(H(j\omega))/\gamma)^2\right) d\omega & \text{if } \|H\|_\infty < \gamma \\ \infty & \text{if } \|H\|_\infty \geq \gamma \end{cases}$$

(*c.f.* (5.22)). This last formula allows us to interpret the γ-entropy of H as a measure of its size, that puts a weight $-\gamma^2 \log(1 - (\sigma/\gamma)^2)$ on a singular value σ, whereas the $\mathbf{H}_2$ norm uses the weight σ^2. This weight function is shown in figure 5.18, with σ^2 shown for comparison.

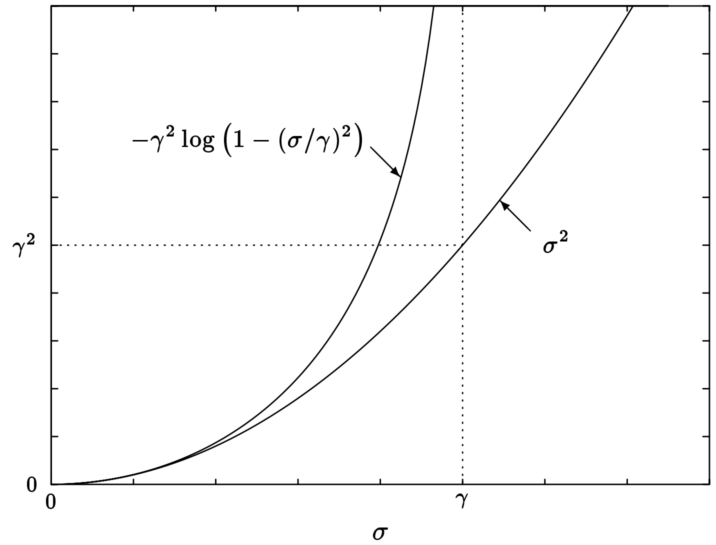


Figure 5.18 The γ-entropy of H is a measure of its size that puts a weight $-\gamma^2 \log(1 - (\sigma/\gamma)^2)$ on a singular value σ, whereas the $\mathbf{H}_2$ norm uses the weight σ^2.

Since these two weight functions are close when σ_i is small compared to γ, we see that

$$\lim_{\gamma\to\infty} \sqrt{I_\gamma(H)} = \|H\|_2. \tag{5.24}$$

From figure 5.18 we can see that

$$\sqrt{I_\gamma(H)} \geq \|H\|_2, \tag{5.25}$$

i.e., the square root of the γ-entropy of a transfer matrix is no smaller than its $\mathbf{H}_2$ norm. We also have the more complicated converse inequality

$$\sqrt{I_\gamma(H)} \le \frac{1}{\alpha}\sqrt{-\log(1-\alpha^2)}\,\|H\|_2, \tag{5.26}$$

where $\alpha = \|H\|_\infty/\gamma < 1$. Thus, the relative increase in the square root of the γ-entropy over the $\mathbf{H}_2$ norm can be bounded by an expression that only depends on how close the $\mathbf{H}_\infty$ norm is to the critical value γ. For example, if $\|H\|_\infty \le \gamma/2$ ($\alpha \le 0.5$), we have $\|H\|_2 \le \sqrt{I_\gamma(H)} \le 1.073\|H\|_2$, *i.e.*, if γ exceeds $\|H\|_\infty$ by 6dB or more, then the square root of the γ-entropy and $\|H\|_2$ cannot differ by more than about 0.6dB. Thus, the $\mathbf{H}_\infty$ norm of a transfer matrix must be near γ for the square root of the γ-entropy to differ much from the $\mathbf{H}_2$ norm.

An important property of the entropy is that it is readily computed using state-space methods, as we will see in section 5.6.5.

An Interpretation of the Entropy

Recall that the square of the $\mathbf{H}_2$ norm of a transfer function H is the power of its output when it is driven by a white noise. For the case of scalar (*i.e.*, transfer function) H, we can give a similar interpretation of the γ-entropy of H as the average power of its output when it is driven by a white noise, and a certain *random* feedback is connected around it, as shown in figure 5.19.

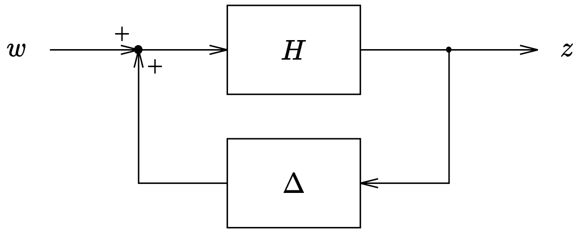


Figure 5.19 The γ-entropy of a transfer function H is the average power of its output z when it is driven by a white noise w and a random feedback Δ is connected around it. The values of the random feedback transfer function Δ are independent at different frequencies, and uniformly distributed on a disk of radius $1/\gamma$ centered at the origin in $\mathbf{C}$.

The transfer function from w to z in figure 5.19, with a particular feedback transfer function Δ connected, is $H/(1-\Delta H)$, so the power of the output signal is

$$\left\|\frac{H}{1-\Delta H}\right\|_2^2.$$

We now assume that Δ is random, with $\Delta(j\omega)$ and $\Delta(j\nu)$ independent for $\omega \ne \nu$, and each $\Delta(j\omega)$ uniformly distributed on the disk of radius $1/\gamma$ in the complex plane. Then we have

$$\mathop{\mathbf{E}}_{\Delta} \left\| \frac{H}{1-\Delta H} \right\|_2^2 = I_\gamma(H), \tag{5.27}$$

where $\mathbf{E}$ denotes expectation over the random feedback transfer functions Δ.

Some feedback transfer functions decrease the power in the output, while other feedback transfer functions increase it; the inequality (5.25) shows that on average, the power in the output is increased by the feedback. The limit (5.24) shows that if the feedback is small, then it has little effect on the output power (indeed, (5.26) shows that the average effect of the feedback on the output power is small unless the $\mathbf{H}_\infty$ norm of the feedback is close to the inverse of the $\mathbf{H}_\infty$ norm of H).

5.3.6 Scaling and Weights

The discussion of section 4.3.5 concerning scalings and weights for norms of vector signals has important implications for norms of MIMO systems; gains especially are affected by the scaling used to measure the input and output vector signals. For example, let us consider the effect of scaling on the RMS gain of a (square) MIMO system. Let D be a scaling matrix (*i.e.*, diagonal). The D-scaled RMS value of Hw is $\|Hw\|_{D,\mathrm{rms}} = \|DHw\|_{\mathrm{rms}}$; the D-scaled RMS value of w is $\|w\|_{D,\mathrm{rms}} = \|Dw\|_{\mathrm{rms}}$. Thus, the D-scaled RMS gain of H is

$$\begin{aligned} \sup_{\|w\|_{\mathrm{rms}}\neq 0} \frac{\|DHw\|_{\mathrm{rms}}}{\|Dw\|_{\mathrm{rms}}} &= \sup_{\|\tilde{w}\|_{\mathrm{rms}}\neq 0} \frac{\|DHD^{-1}\tilde{w}\|_{\mathrm{rms}}}{\|\tilde{w}\|_{\mathrm{rms}}} \\ &= \|DHD^{-1}\|_\infty, \end{aligned}$$

the $\mathbf{H}_\infty$ norm of the diagonally pre- and post-scaled transfer matrix.

More general transfer matrix weights can be applied, *e.g.*, $\|W_{\mathrm{post}} H W_{\mathrm{pre}}\|$.

5.4 Important Properties of Gains

5.4.1 Gain of a Cascade Connection

An important property of gains is that the gain of the product of two transfer matrices can be bounded in terms of the gains of the individual transfer matrices: if $\|\cdot\|_{\mathrm{gn}}$ denotes any gain, then

$$\|H_2 H_1\|_{\mathrm{gn}} \le \|H_2\|_{\mathrm{gn}} \|H_1\|_{\mathrm{gn}}. \tag{5.28}$$

This inequality is easily established; it can be seen from figure 5.20. The property (5.28) does not generally hold for norms of LTI systems that are not gains. For example,

$$H_1 = \frac{3s}{s^2+s+3}$$
$$H_2 = \frac{3s}{s^2+0.5s+2}$$

have $\|H_1\|_{\mathrm{pk\_step}} = 1.18$ and $\|H_2\|_{\mathrm{pk\_step}} = 1.65$, so that $\|H_1\|_{\mathrm{pk\_step}}\|H_2\|_{\mathrm{pk\_step}} = 1.95$, but $\|H_2H_1\|_{\mathrm{pk\_step}} = 2.80 > 1.95$.

On the other hand, $\|H_1\|_\infty = 3$, $\|H_2\|_\infty = 6$, and $\|H_2H_1\|_\infty = 15.3 \le 18$, since the $\mathbf{H}_\infty$ norm is the RMS gain.

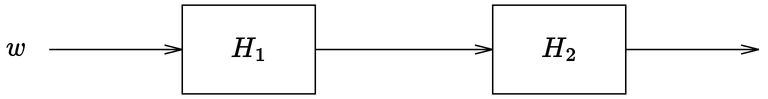


Figure 5.20 The gain of two cascaded transfer matrices is no larger than the product of the gains of the transfer matrices, *i.e.*: $\|H_2H_1\|_{\mathrm{gn}} \le \|H_2\|_{\mathrm{gn}}\|H_1\|_{\mathrm{gn}}$.

5.4.2 Gain of a Feedback Connection

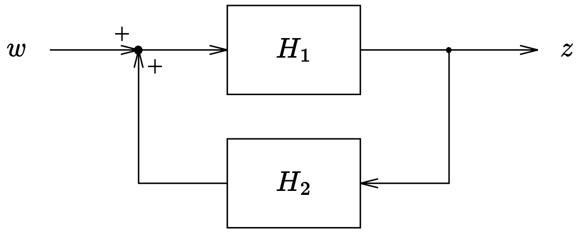


Figure 5.21 Two systems connected in a feedback loop.

Consider the feedback connection shown in figure 5.21. Assuming that this feedback connection is well-posed, meaning that $\det(I - H_1H_2)$ is not identically zero, the transfer matrix from w to z is

$$G = (I - H_1H_2)^{-1}H_1 = H_1(I - H_2H_1)^{-1}. \tag{5.29}$$

A fact that we will need in chapter 10 is that, provided the product of the gains of the two transfer matrices is less than one, the gain of G can be bounded. More precisely, if

$$\|H_1\|_{\mathrm{gn}}\|H_2\|_{\mathrm{gn}} < 1 \tag{5.30}$$

holds, then the feedback connection is well-posed and we have

$$\|G\|_{\mathrm{gn}} \le \frac{\|H_1\|_{\mathrm{gn}}}{1 - \|H_1\|_{\mathrm{gn}}\|H_2\|_{\mathrm{gn}}} \tag{5.31}$$

(note the similarity to (5.29)).

The condition (5.30) is called the "small gain" condition, since, roughly speaking, it limits the gain around the loop in figure 5.21 to less than one. For this reason, the result above is sometimes called the *small gain theorem.*

The small gain theorem can be used to establish stability of a feedback connection, if the gain $\|\cdot\|_{\mathrm{gn}}$ is such that $\|H\|_{\mathrm{gn}} < \infty$ implies that H is stable. The RMS gain, for example, has this property: the small gain condition $\|H_1\|_\infty \|H_2\|_\infty < 1$ implies that the transfer matrix G is stable, that is, all of its poles have negative real part. Similarly, if the gain $\|\cdot\|_{\mathrm{gn}}$ is the a-shifted $\mathbf{H}_\infty$ norm, then the small gain condition $\|H_1\|_{\infty,a}\|H_2\|_{\infty,a} < 1$ implies that the poles of G have real parts less than $-a$.

The small gain theorem is easily shown. Suppose that the small gain condition (5.30) holds. The feedback connection of figure 5.21 means

$$z = H_1(w + H_2 z) = H_1 w + H_1 H_2 z.$$

Using the triangle inequality,

$$\|z\| \le \|H_1 w\| + \|H_1 H_2 z\|,$$

where $\|\cdot\|$ is the norm used for all signals. Using the definition of gain and the property (5.28), we have

$$\|z\| \le \|H_1\|_{\mathrm{gn}}\|w\| + \|H_1 H_2\|_{\mathrm{gn}}\|z\| \le \|H_1\|_{\mathrm{gn}}\|w\| + \|H_1\|_{\mathrm{gn}}\|H_2\|_{\mathrm{gn}}\|z\|,$$

so that

$$\|z\|\left(1 - \|H_1\|_{\mathrm{gn}}\|H_2\|_{\mathrm{gn}}\right) \le \|H_1\|_{\mathrm{gn}}\|w\|.$$

Using the small gain condition, we have

$$\|z\| \le \frac{\|H_1\|_{\mathrm{gn}}}{1 - \|H_1\|_{\mathrm{gn}}\|H_2\|_{\mathrm{gn}}}\|w\|. \tag{5.32}$$

Since (5.32) holds for all signals w, (5.31) follows.

5.5 Comparing Norms

The intuition that different norms for LTI systems should generally agree about which transfer matrices are "small" or "large" is false. There are, however, some general inequalities that the norms we have seen must satisfy.

5.5.1 Some General Inequalities

For convenience we will consider norms of SISO systems, *i.e.*, norms of transfer functions.

Since $\|\cdot\|_{\mathrm{pk\_step}}$ and $\|\cdot\|_\infty$ are each worst case peak norms over input signal sets that have peaks no greater than one (a unit step in the first case, and unit amplitude sinusoids in the second), it follows that these norms will be no larger than the peak gain of the system,

$$\begin{aligned}\|H\|_{\mathrm{pk\_gn}} &\geq \|H\|_\infty,\\ \|H\|_{\mathrm{pk\_gn}} &\geq \|H\|_{\mathrm{pk\_step}}.\end{aligned}$$

From the definition of the Hankel norm, we can see that it cannot exceed the $\mathbf{L}_2$ gain, which we saw is the $\mathbf{H}_\infty$ norm, so we have

$$\|H\|_{\mathrm{hankel}} \leq \|H\|_\infty.$$

It is possible for a system to have a small RMS gain, but a large peak gain. However, if H has n poles then the peak gain of H can be bounded in terms of the Hankel norm, and therefore, the RMS gain:

$$\|H\|_\infty \leq \|H\|_{\mathrm{pk\_gn}} \leq (2n+1)\|H\|_{\mathrm{hankel}} \leq (2n+1)\|H\|_\infty. \tag{5.33}$$

This means that for low order systems, at least, the peak gain, RMS gain, and Hankel norm cannot differ too much.

5.5.2 Approximating Norms: an Example

Consider the worst case norm described in section 5.1.3, with the amplitude bound and slew-rate limit each equal to one:

$$\|H\|_{\mathrm{wc}} = \sup\left\{\|Hw\|_\infty \;\middle|\; \|w\|_\infty \leq 1, \|\dot{w}\|_\infty \leq 1\right\}.$$

Roughly speaking, the bound and slew-rate limit establish a *bandwidth* limit of about one for the input signal w. We might therefore suspect that we can approximate $\|H\|_{\mathrm{wc}}$ by a weighted peak gain, where the weight is some appropriate lowpass filter with a bandwidth near one:

$$\|H\|_{\mathrm{wc}} \approx \|HW\|_{\mathrm{pk\_gn}}.$$

We will show that this intuition is correct: for $W(s) = 1/(2s+1)$, we have

$$\|HW\|_{\mathrm{pk\_gn}} \leq \|H\|_{\mathrm{wc}} \leq 3\|HW\|_{\mathrm{pk\_gn}} \tag{5.34}$$

for all transfer functions H. Thus, $\sqrt{3}\,\|HW\|_{\mathrm{pk\_gn}}$ approximates $\|H\|_{\mathrm{wc}}$ to within ±4.8dB.

To establish (5.34), suppose that w_0 is a signal with

$$\|w_0\|_\infty = 1, \qquad \|HWw_0\|_\infty = \|HW\|_{\mathrm{pk\_gn}}$$

(such a w_0 can be shown to exist). Let $w_1 = Ww_0$. Then we have

$$\|w_1\|_\infty \le \|W\|_{\rm pk\_gn} = 1,$$

and

$$\|\dot{w}_1\|_\infty = \left\| \frac{s}{2s+1} w_0 \right\|_\infty \le \left\| \frac{s}{2s+1} \right\|_{\rm pk\_gn} = 1.$$

Therefore w_1 satisfies the amplitude limit $\|w\|_\infty \le 1$ and slew-rate limit $\|\dot{w}_1\|_\infty \le 1$, so we must have $\|Hw_1\|_\infty \le \|H\|_{\rm wc}$. Since $\|Hw_1\|_\infty = \|HW\|_{\rm pk\_gn}$, this means that

$$\|HW\|_{\rm pk\_gn} \le \|H\|_{\rm wc}.$$

We now establish the right-hand inequality in (5.34). Consider w_2 such that

$$\|w_2\|_\infty \le 1, \quad \|\dot{w}_2\|_\infty \le 1, \quad \|Hw_2\|_\infty = \|H\|_{\rm wc}$$

(such a w_2 can be shown to exist). Define w_3 by

$$w_3 = 2\dot{w}_2 + w_2 = W^{-1} w_2.$$

Then $\|w_3\|_\infty \le 2\|\dot{w}_2\|_\infty + \|w_2\|_\infty = 3$, and $\|HWw_3\|_\infty = \|Hw_2\|_\infty = \|H\|_{\rm wc}$. Hence

$$\|HW\|_{\rm pk\_gn} \ge \frac{\|HWw_3\|_\infty}{\|w_3\|_\infty} = \frac{\|H\|_{\rm wc}}{3},$$

which establishes the right-hand inequality in (5.34).

This example illustrates an interesting tradeoff in the selection of a norm. While $\|H\|_{\rm wc}$ may better characterize the "size" of H in a given situation, the approximation $\|HW\|_{\rm pk\_gn}$ (or even $\|HW\|_\infty$; see the previous section) may be easier to work with, *e.g.*, compute. If $\|w\|_\infty \le 1$ and $\|\dot{w}\|_\infty \le 1$ is only an approximate model of possible w's, and the specification $\|z\|_\infty \le \alpha$ need only hold within a factor of two or so, then

$$\|HW\|_{\rm pk\_gn} \le \sqrt{3}\,\alpha$$

would be an appropriate specification.

5.6 State-Space Methods for Computing Norms

The $\mathbf{H}_2$, Hankel, and $\mathbf{H}_\infty$ norms, and the γ-entropy of a transfer matrix are readily computed from a state-space realization; methods for computing some of the other norms we have seen are described in the Notes and References. In this section we assume that

$$\dot{x} = Ax + Bw, \qquad z = Cx$$

is a minimal realization of the stable transfer matrix H, *i.e.*,

$$H(s) = C(sI - A)^{-1}B.$$

The references cited at the end of this chapter give the generalization of the $\mathbf{H}_\infty$ norm computation method to the case with a feed-through term $(z = Cx + Dw)$.

5.6.1 Computing the $\mathbf{H}_2$ Norm

Substituting the impulse matrix $h(t) = Ce^{At}B$ of H into

$$\|H\|_2^2 = \mathbf{Tr}\left(\int_0^\infty h(t)^T h(t)\, dt\right)$$

we have

$$\begin{aligned}\|H\|_2^2 &= \mathbf{Tr}\left(B^T \int_0^\infty e^{A^T t} C^T C e^{At}\, dt\, B\right)\\ &= \mathbf{Tr}\left(B^T W_{\mathrm{obs}} B\right), \qquad (5.35)\end{aligned}$$

where

$$W_{\mathrm{obs}} \triangleq \int_0^\infty e^{A^T t} C^T C e^{At}\, dt$$

is the *observability Gramian* of the realization, which can be computed by solving the *Lyapunov equation*

$$A^T W_{\mathrm{obs}} + W_{\mathrm{obs}} A + C^T C = 0 \qquad (5.36)$$

(see the Notes and References).

The observability Gramian determines the total energy in the system output, starting from a given initial state, with no input:

$$x(0)^T W_{\mathrm{obs}} x(0) = \int_0^\infty z(t)^T z(t)\, dt,$$

where $\dot{x} = Ax$, $z = Cx$.

Since $\mathbf{Tr}(RS) = \mathbf{Tr}(SR)$, the above derivation can be repeated to give an alternate formula

$$\|H\|_2 = \left(\mathbf{Tr}\left(C W_{\mathrm{contr}} C^T\right)\right)^{1/2}, \qquad (5.37)$$

where

$$W_{\mathrm{contr}} \triangleq \int_0^\infty e^{At} B B^T e^{A^T t}\, dt$$

is the *controllability Gramian*, which can be found by solving the Lyapunov equation

$$AW_{\rm contr} + W_{\rm contr}A^T + BB^T = 0. \tag{5.38}$$

The controllability Gramian determines which points in state-space can be reached using an input with total energy one:

$$\begin{array}{l} \dot{x} = Ax + Bw, \quad x(0) = 0, \quad x(T) = x_d, \quad \int_0^T w(t)^T w(t)\, dt \le 1, \\ \text{for some } T \text{ and } w \\ \quad\Longleftrightarrow \\ x_d^T W_{\rm contr}^{-1} x_d < 1. \end{array}$$

Thus, the points in state-space that can be reached using an excitation with total energy one is given by an ellipsoid determined by $W_{\rm contr}$. (See chapter 14 for more discussion of ellipsoids.)

5.6.2 Computing the Hankel Norm

The Hankel norm is readily computed from the controllability and observability Gramians via

$$\begin{aligned} \|H\|_{\rm hankel} &= \left(\lambda_{\max}(W_{\rm contr}^{1/2} W_{\rm obs} W_{\rm contr}^{1/2})\right)^{1/2} \\ &= \left(\lambda_{\max}(W_{\rm obs} W_{\rm contr})\right)^{1/2}, \end{aligned} \tag{5.39}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue. Roughly speaking, the Gramian $W_{\rm obs}$ measures the energy that can be "retrieved" in the output from the system state, and $W_{\rm contr}$ measures the amount of energy that can be "stored" in the system state using an excitation with a given energy. These are the two "parts" of the mapping from the excitation to the resulting ring that we mentioned in section 5.2.8; (5.39) shows how the Hankel norm depends on the "sizes" of these two parts.

5.6.3 Computing the $\mathbf{H}_\infty$ Norm

There is a simple method for determining whether the inequality specification $\|H\|_\infty < \gamma$ is satisfied. Given $\gamma > 0$ we define the matrix

$$M_\gamma = \begin{bmatrix} A & \gamma^{-1}BB^T \\ -\gamma^{-1}C^TC & -A^T \end{bmatrix}. \tag{5.40}$$

Then we have

$$\|H\|_\infty < \gamma \iff M_\gamma \text{ has no imaginary eigenvalues.} \tag{5.41}$$

Hence we can check whether the specification $\|H\| < \gamma$ is satisfied by forming M_γ and computing its eigenvalues. The equivalence (5.41) can be used to devise an algorithm that computes $\|H\|_\infty$ with guaranteed accuracy, by trying different values of γ; see the Notes and References at the end of this chapter.

The result (5.41) can be understood as follows. $\|H\|_\infty < \gamma$ is true if and only if for all $\omega \in \mathbf{R}$, $\gamma^2 I - H(j\omega)^* H(j\omega)$ is invertible, or equivalently, the transfer matrix

$$G(s) = \left(I - \gamma^{-2} H(-s)^T H(s)\right)^{-1}$$

has no $j\omega$ axis poles. We can derive a realization of G as follows. A realization of $H(-s)^T$ (which is the *adjoint* system) is given by

$$H(-s)^T = \left(B^T\right)\left(sI - \left(-A^T\right)\right)^{-1}\left(-C^T\right).$$

Using this and the block diagram of G shown in figure 5.22, a realization of G is given by $G(s) = C_G(sI - A_G)^{-1}B_G + D_G$, where

$$\begin{aligned} A_G &= M_\gamma \\ B_G &= \begin{bmatrix} B \\ 0 \end{bmatrix} \\ C_G &= \begin{bmatrix} 0 & \gamma^{-1}B^T \end{bmatrix} \\ D_G &= I. \end{aligned}$$

Since $A_G = M_\gamma$, and it can be shown that this realization of G is minimal, the $j\omega$ axis poles of G are exactly the imaginary eigenvalues of M_γ, and (5.41) follows.

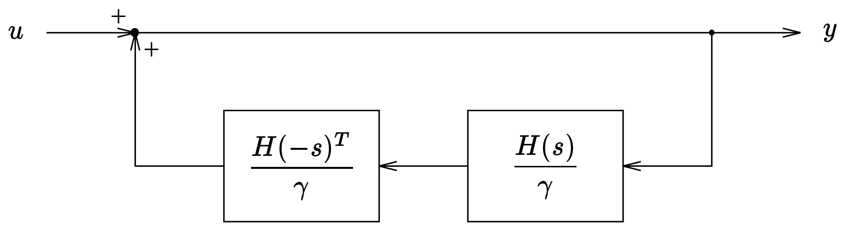


Figure 5.22 A realization of $G(s) = \gamma^2(\gamma^2 I - H(-s)^T H(s))^{-1}$. G has no imaginary axis poles if and only if the inequality specification $\|H\|_\infty < \gamma$ holds.

The condition that M_γ not have any imaginary eigenvalues can also be expressed in terms of a related *algebraic Riccati equation* (ARE),

$$A^T X + XA + \gamma^{-1} XBB^T X + \gamma^{-1} C^T C = 0. \tag{5.42}$$

(Conversely, M_γ is called the *Hamiltonian matrix* associated with the ARE (5.42).) This equation will have a positive definite solution X if and only if M_γ has no imaginary eigenvalues (in which case there will be only one such X). If $\|H\|_\infty < \gamma$, we can compute the positive definite solution to (5.42) as follows. We compute any matrix T such that

$$T^{-1} M_\gamma T = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}$$

where $\tilde{A}_{11}$ is stable (*i.e.*, all of its eigenvalues have negative real part). (One good choice is to compute the ordered Schur form of M_γ; see the Notes and References at the end of this chapter.) We then partition T as

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix},$$

and the solution X is given by

$$X = T_{21}T_{11}^{-1}.$$

The significance of X is discussed in the Notes and References for chapter 10.

5.6.4 Computing the a-Shifted $\mathbf{H}_\infty$ Norm

The results of the previous section can be applied to the realization

$$\dot{x} = (A + aI)x + Bw, \qquad z = Cx$$

of the a-shifted transfer matrix $H(s - a)$. We find that $\|H\|_{\infty,a} < \gamma$ holds if and only if the eigenvalues of A all have real part less than $-a$ and the matrix

$$\begin{bmatrix} A + aI & \gamma^{-1}BB^T \\ -\gamma^{-1}C^TC & -A^T - aI \end{bmatrix}$$

has no imaginary eigenvalues.

5.6.5 Computing the Entropy

To compute the γ-entropy of H, we first form the matrix M_γ in (5.40). If M_γ has any imaginary eigenvalues, then by the result (5.41), $\|H\|_\infty \geq \gamma$ and hence $I_\gamma(H) = \infty$. If M_γ has no imaginary eigenvalues, then we find the positive definite solution X of the ARE (5.42) as described above. The γ-entropy is then

$$I_\gamma(H) = \gamma\,\mathbf{Tr}\left(B^TXB\right). \tag{5.43}$$

From (5.42), the matrix $\tilde{X} = \gamma X$ satisfies the ARE

$$A^T\tilde{X} + \tilde{X}A + \gamma^{-2}\tilde{X}BB^T\tilde{X} + C^TC = 0.$$

Note that as $\gamma \to \infty$, this ARE becomes the Lyapunov equation for the observability Gramian (5.36), so the solution $\tilde{X}$ converges to the observability Gramian W_{obs}. From (5.43) and (5.35) we see again that $I_\gamma(H) \to \|H\|_2^2$ as $\gamma \to \infty$ (see (5.24) in section 5.3.5).

Notes and References

Norms of Signals and Systems for Control System Analysis

The use of norms in feedback system analysis was popularized in the 1960's by researchers such as Zames [ZAM66B], Sandberg [SAN64], Narendra [NG64], and Willems [WIL69], although some norms had been used in control systems analysis before these papers. For example, the square of the $\mathbf{H}_2$ norm is referred to as I_y in the 1957 book [NGK57]; chapter 7 of the 1947 book [JNP47], written by Philips, is entitled *RMS-Error Criterion in Servomechanism Design.*

A thorough reference on norms for signals and systems in the context of control systems is Desoer and Vidyasagar [DV75]. This book contains general and precise definitions of many of the norms in this and the previous chapter. Mathematics texts covering many of the norms we have seen include Kolmogorov and Fomin [KF75] and Aubin [AUB79]. The bold $\mathbf{H}$ in the symbols $\mathbf{H}_2$ and $\mathbf{H}_\infty$ stands for the mathematician G. H. Hardy.

The observation that the total variation of the step response is the peak gain of a transfer function appears in Lunze [LUN89].

Singular Value Plots

Singular value plots are discussed in, for example, Callier and Desoer [CD82A], Maciejowski [MAC89], and Lunze [LUN89]. Analytical properties of singular values and numerical algorithms for computing them are covered in Golub and Van Loan [GL89].

The Entropy Interpretation

The simple interpretation of the entropy, as the average square of the $\mathbf{H}_2$ norm when a random feedback is connected around a transfer function, has not appeared before. We do not know how to generalize this interpretation to the case of a transfer matrix, although it is likely that there is a similar interpretation.

To prove the result (5.27) we consider an arbitrary $h \in \mathbf{C}$, and Δ a complex random variable uniformly distributed on the disk of radius $1/\gamma$. We then have

$$\begin{aligned}\mathbf{E}\left|\frac{h}{1-\Delta h}\right|^2 &= \frac{\gamma^2}{\pi}\int_0^{1/\gamma}\int_0^{2\pi}\left|\frac{h}{1-re^{i\theta}h}\right|^2 r\,d\theta\,dr \\ &= \begin{cases} -\gamma^2\log(1-|h|^2/\gamma^2) & |h|<\gamma \\ \infty & |h|\geq\gamma\end{cases}\end{aligned}$$

(the integration over θ can be evaluated by residues). By integrating over ω, the result (5.27) follows.

Comparing Gains

The result (5.33) is from Boyd and Doyle [BD87].

Small Gain Theorem

The small gain theorem from section 5.4.2 is a standard mathematical result. Applications of this result (and extensions) to feedback system analysis are discussed in Desoer and Vidyasagar [DV75] and Vidyasagar [VID78]; see also chapter 10.

State-Space Norm Computations

Using the controllability or observability Gramian to compute $\mathbf{H}_2$ norms is standard; see for example [FRA87]. The Lyapunov equations that arise are nowadays solved numerically by special methods; see Bartels and Stewart [BS72] and Golub, Nash, and Van Loan [GNL79]. Tables of formulas for the $\mathbf{H}_2$ norm of a transfer function, in terms of its numerator and denominator coefficients, can be found in Appendix E2 of Newton, Gould, and Kaiser [NGK57]. These tables are based on a method that is equivalent to solving the Lyapunov equations. (Professor T. Higgins points out that there are several errors in these tables.)

The result on Hankel norm computation can be found in, *e.g.*, [GLO84, §2.3] and [FRA87]. The result of section 5.6.3 is from Boyd, Balakrishnan, and Kabamba [BBK89]; see also Robel [ROB89] and Boyd and Balakrishnan [BB90]. The method for computing the entropy appears in Mustafa and Glover [MG90] and Glover and Mustafa [GM89].

A discussion of solving the ARE can be found in [AM90]. The method of solving the ARE based on the Schur form is discussed in Laub [LAU79]; see also the articles [AL84, DOO81]. Numerical issues of these and other state-space computations are discussed in Laub [LAU85].

Computing Some Other Norms

Computing the peak gain or peak-step norm from a state-space description of an LTI system is more involved than computing the entropy or the $\mathbf{H}_2$ or $\mathbf{H}_\infty$ norm. Perhaps the simplest method is to numerically integrate (*i.e.*, solve) the state-space equations to obtain the impulse or step response matrix. $\|H\|_{\mathrm{pk\_gn}}$ could then be computed by numerical integration of the integrals in the formula (5.23). Similarly, $\|H\|_{\mathrm{pk\_step}}$ could be determined directly from its definition and the computed step response matrix.

For other norms, *e.g.*, $\|H\|_{\mathrm{wc}}$, there is not even a simple formula like (5.23). Nevertheless it can be computed in several ways; we briefly mention some here. It can be expressed as

$$\|H\|_{\mathrm{wc}} = \sup \left\{ \int_0^\infty h(t)w(t)\,dt \;\middle|\; \|w\|_\infty \le M_{\mathrm{ampl}}, \|\dot{w}\|_\infty \le M_{\mathrm{slew}} \right\}, \qquad (5.44)$$

which is an infinite-dimensional convex optimization problem that can be solved using the methods described in chapters 13–15. Alternatively, the (infinite-dimensional) *dual problem* can be solved:

$$\|H\|_{\mathrm{wc}} = \min_{\substack{\mu \in \mathbf{R} \\ \lambda : \mathbf{R}_+ \to \mathbf{R}}} M_{\mathrm{ampl}} \|\lambda\|_1 + M_{\mathrm{slew}} \left\| \mu + \int_0^t (h(\tau) - \lambda(\tau))\, d\tau \right\|_1 .$$

This dual problem is unconstrained.

The computation of $\|H\|_{\mathrm{wc}}$ can also be formulated as an optimal control problem. We include w as an additional state, so the dynamics are

$$\dot{x} = Ax + Bw, \qquad \dot{w} = u, \qquad z = Cx, \qquad x(0) = w(0) = 0.$$

The peak and slew-rate limits on w can be enforced as the control and state constraints

$$|u(t)| \le M_{\mathrm{slew}}, \qquad |w(t)| \le M_{\mathrm{ampl}}.$$

The objective is then simply $z(T)$ with T free. The solution of this free end-point optimal control problem (which always occurs as $T \to \infty$) yields $\|H\|_{\rm wc}$. This optimal control problem has linear dynamics and convex state and control constraints, so essentially all numerical methods of solution will find the value $\|H\|_{\rm wc}$ (and not some local minimum). See, for example, the books by Pontryagin [PON62], Bryson and Ho [BH75, CH.7], and the survey article by Polak [POL73].

The Figures of Section 5.2.9

The input signals in figure 5.11 were computed by finely discretizing (with first-order hold) the optimization problem

$$\max_{\substack{\|w\|_\infty \le 1 \\ \|\dot{w}\|_\infty \le 1}} \int_0^{10} h(10-t)w(t)\,dt$$

and solving the resulting linear program.

The input signals in figure 5.12 were computed from (5.6) with $T = 10$.

The unit-energy input signals in figure 5.13 give the largest possible square root output energy for $t \ge 5$. These input signals were computed by finding the finite-time controllability Gramian

$$W \stackrel{\Delta}{=} W_{\rm contr}^{[0,5]} = \int_0^5 e^{At}BB^Te^{A^Tt}\,dt = W_{\rm contr} - e^{5A}W_{\rm contr}e^{5A^T}.$$

where $H(s) = C(sI-A)^{-1}B$.

If λ is the largest eigenvalue of $W^{1/2}W_{\rm obs}W^{1/2}$ and z is the corresponding eigenvector, with $\|z\|_2 = 1$, then the input signal

$$w(t) = \begin{cases} B^Te^{A^T(5-t)}W^{-1/2}z & \text{for } 0 \le t \le 5, \\ 0 & \text{otherwise,} \end{cases}$$

has unit energy, and drives the system state to $W^{1/2}z$ at $t = 5$. (It is actually the smallest energy signal that drives the state from the origin to $W^{1/2}z$ in 5 seconds.) The output for $t \ge 5$,

$$z(t) = Ce^{A(t-5)}W^{1/2}z,$$

has square root energy

$$\lambda^{1/2} = \left(\lambda_{\max}(W^{1/2}W_{\rm obs}W^{1/2})\right)^{1/2}$$

(*c.f.* (5.39)).

Chapter 6

Geometry of Design Specifications

In this chapter we explore some geometric properties that design specifications may have, and define the important notion of a *closed-loop convex design specification.* We will see in the sequel that simple and effective methods can be used to solve controller design problems that are formulated entirely in terms of closed-loop convex design specifications.

6.1 Design Specifications as Sets

$\mathcal{H}$ will denote the set of *all* $n_z \times n_w$ closed-loop transfer matrices; we may think of $\mathcal{H}$ as the set of all conceivable candidate transfer matrices for the given plant. Recall from chapter 3 that design specifications are boolean functions or predicates on $\mathcal{H}$. With each design specification $\mathcal{D}_i$ we will associate the set $\mathcal{H}_i$ of all transfer matrices that satisfy it:

$$\mathcal{H}_i = \{H \in \mathcal{H} \mid H \text{ satisfies } \mathcal{D}_i\}.$$

Of course, there is a one-to-one correspondence between subsets of $\mathcal{H}$ (*i.e.*, sets of transfer matrices) and design specifications. For this reason we will also refer to subsets of $\mathcal{H}$ as design specifications. Whether the predicate (*e.g.*, $\phi_{\rm os}(H) \leq 6\%$) or subset ($\{H \mid \phi_{\rm os}(H) \leq 6\%\}$) is meant should be clear from the context, if it matters at all.

The boolean algebra of design specifications mentioned in chapter 3 corresponds exactly to the boolean algebra of subsets, with some of the correspondences listed in table 6.1.

Design specifications	Sets of transfer matrices
H satisfies $\mathcal{D}_1$	$H \in \mathcal{H}_1$
$\mathcal{D}_1$ is stronger than $\mathcal{D}_2$	$\mathcal{H}_1 \subseteq \mathcal{H}_2$
$\mathcal{D}_1$ is weaker than $\mathcal{D}_2$	$\mathcal{H}_1 \supseteq \mathcal{H}_2$
$\mathcal{D}_1 \wedge \mathcal{D}_2$	$\mathcal{H}_1 \cap \mathcal{H}_2$
$\mathcal{D}_1$ is infeasible	$\mathcal{H}_1 = \emptyset$
$\mathcal{D}_1$ is feasible	$\mathcal{H}_1 \neq \emptyset$
$\mathcal{D}_1$ is strictly stronger than $\mathcal{D}_2$	$\mathcal{H}_1 \subseteq \mathcal{H}_2$, $\mathcal{H}_1 \neq \mathcal{H}_2$

Table 6.1 Properties of design specifications and the corresponding sets of transfer matrices.

6.2 Affine and Convex Sets and Functionals

In this section we introduce several important definitions.

We remind the reader that $\mathcal{H}$ is a vector space: roughly speaking, we have a way of adding two of its elements (*i.e.*, $n_z \times n_w$ transfer matrices) and multiplying one by a real scalar. In a vector space, we have the important concepts of a *line* and a *line segment.*

If H, $\tilde{H} \in \mathcal{H}$ and $\lambda \in \mathbf{R}$, we will refer to $\lambda H + (1-\lambda)\tilde{H}$ as an *affine combination* of H and $\tilde{H}$. We may think of an affine combination as lying on the line passing through H and $\tilde{H}$, provided $H \neq \tilde{H}$. If $0 \leq \lambda \leq 1$, we will refer to the affine combination $\lambda H + (1-\lambda)\tilde{H}$ as a *convex combination* of H and $\tilde{H}$. We may think of a convex combination as lying on the line segment between H and $\tilde{H}$. The number λ measures the fraction of the line segment that we move from $\tilde{H}$ towards H to yield $\lambda H + (1-\lambda)\tilde{H}$. This can be seen in figure 6.1.

Definition 6.1: *$\mathcal{H}_1 \subseteq \mathcal{H}$ is* affine *if for any H, $\tilde{H} \in \mathcal{H}_1$, and any $\lambda \in \mathbf{R}$, $\lambda H + (1-\lambda)\tilde{H} \in \mathcal{H}_1$.*

Thus a set of transfer matrices is affine if, whenever two distinct transfer matrices are in the set, so is the entire line passing through them.

Definition 6.2: *$\mathcal{H}_1 \subseteq \mathcal{H}$ is* convex *if for any H, $\tilde{H} \in \mathcal{H}_1$, and any $\lambda \in [0,1]$, $\lambda H + (1-\lambda)\tilde{H} \in \mathcal{H}_1$.*

Thus a set of transfer matrices is convex if, whenever two transfer matrices are in the set, so is the entire line segment between them.

These notions are extended to functionals as follows:

Definition 6.3: *A functional ϕ on $\mathcal{H}$ is* affine *if for any H, $\tilde{H} \in \mathcal{H}$, and any $\lambda \in \mathbf{R}$, $\phi(\lambda H + (1-\lambda)\tilde{H}) = \lambda\phi(H) + (1-\lambda)\phi(\tilde{H})$.*

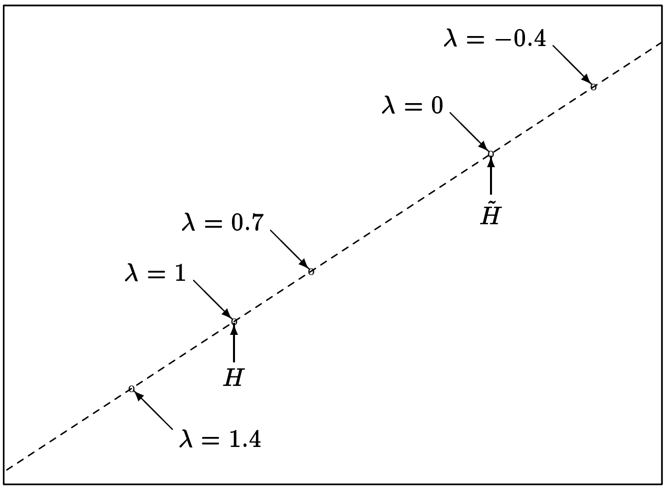


Figure 6.1 The line passing through H and $\tilde{H}$ consists of all affine combinations of H and $\tilde{H}$, *i.e.*, $\lambda H + (1 - \lambda)\tilde{H}$, $\lambda \in \mathbf{R}$. The line segment between H and $\tilde{H}$ consists of all convex combinations of H and $\tilde{H}$, *i.e.*, $\lambda H + (1 - \lambda)\tilde{H}$ for $0 \leq \lambda \leq 1$.

A functional is affine if the graph of its values along any line in $\mathcal{H}$ is a line in $\mathbf{R}^2$; an example is shown in figure 6.2.

Definition 6.4: *A functional ϕ on $\mathcal{H}$ is* convex *if for any H, $\tilde{H} \in \mathcal{H}$, and any $\lambda \in [0, 1]$, $\phi(\lambda H + (1 - \lambda)\tilde{H}) \leq \lambda\phi(H) + (1 - \lambda)\phi(\tilde{H})$.*

A functional is convex if the graph of its values along any line segment in $\mathcal{H}$ lies below the line segment joining its values at the ends of the line segment. This is shown in figure 6.3.

Under very mild conditions we can test convexity of a set or functional by just checking the case $\lambda = 1/2$. Specifically, a set $\mathcal{H}_1$ is convex if and only if whenever $H \in \mathcal{H}_1$ and $\tilde{H} \in \mathcal{H}_1$, the average $(H + \tilde{H})/2$ is also in $\mathcal{H}_1$. Similarly, a functional ϕ is convex if and only if, for every H and $\tilde{H}$ we have

$$\phi((H + \tilde{H})/2) \leq (\phi(H) + \phi(\tilde{H}))/2.$$

Since $(H + \tilde{H})/2$ can be interpreted as the midpoint of the line segment between H and $\tilde{H}$, this simple test is called the *midpoint rule.*

6.2.1 Some Important Properties

We collect here some useful facts about affine and convex sets and functionals.

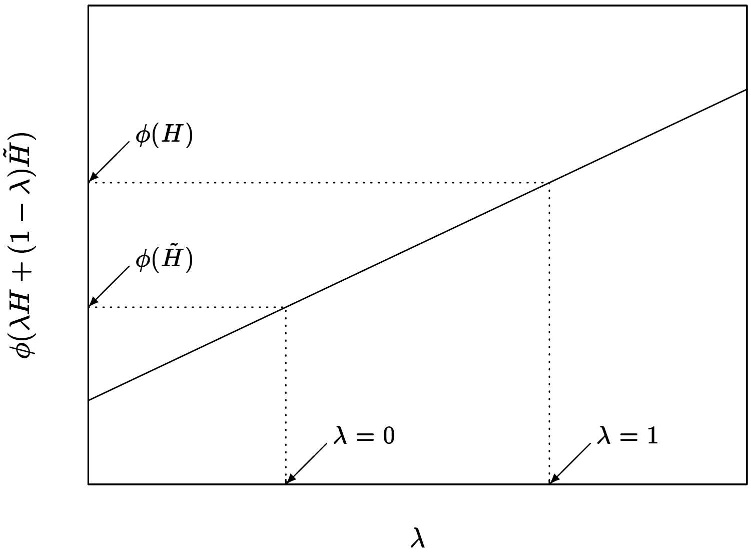


Figure 6.2 A functional ϕ is affine if for every pair of transfer matrices H and $\tilde{H}$ the graph of $\phi(\lambda H + (1-\lambda)\tilde{H})$ versus λ is a straight line passing through the points $(0,\ \phi(\tilde{H}))$, $(1,\ \phi(H))$.

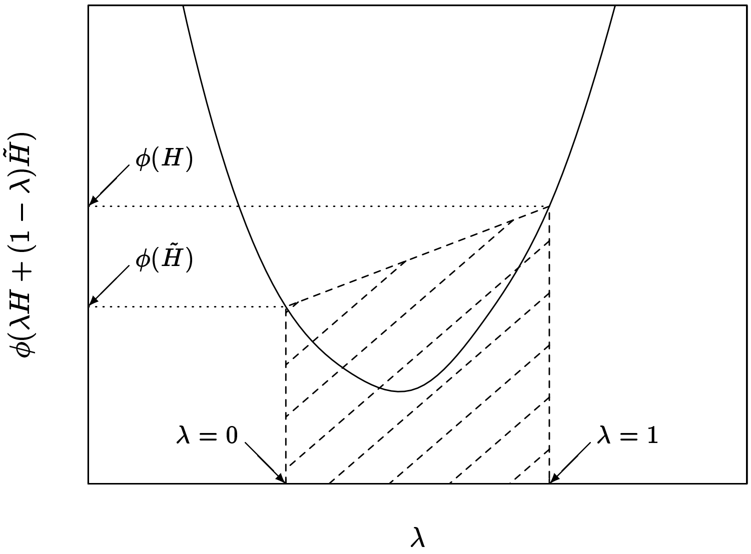


Figure 6.3 A functional ϕ is convex if for every pair of transfer matrices H and $\tilde{H}$ the graph of ϕ along the line $\lambda H + (1-\lambda)\tilde{H}$ lies on or below a straight line through the points $(0,\ \phi(\tilde{H}))$, $(1,\ \phi(H))$, *i.e.*, in the shaded region.

- *Affine implies convex.* If a set or functional is affine, then it is convex: being affine is a stronger condition than convex. If the functionals ϕ and $-\phi$ are both convex, then ϕ is affine.

- *Intersections.* Intersections of affine or convex sets are affine or convex, respectively.

- *Weighted-sum Functional.* If the functionals $\phi_1, \ldots, \phi_L$ are convex, and $\lambda_1 \geq 0, \ldots, \lambda_L \geq 0$, then the weighted-sum functional

  $$\phi_{\mathrm{wt\_sum}}(H) = \lambda_1 \phi_1(H) + \cdots + \lambda_L \phi_L(H)$$

  is convex (see section 3.6.1).

- *Weighted-max Functional.* If the functionals $\phi_1, \ldots, \phi_L$ are convex, and $\lambda_1 \geq 0, \ldots, \lambda_L \geq 0$, then the weighted-max functional

  $$\phi_{\mathrm{wt\_max}}(H) = \max\{\lambda_1 \phi_1(H), \ \ldots, \ \lambda_L \phi_L(H)\}$$

  is convex (see section 3.6.3).

The last two properties can be generalized to the integral of a family of convex functionals and the maximum of an infinite family of convex functionals. Suppose that for each $\alpha \in \mathcal{I}$ ($\mathcal{I}$ is an arbitrary index set), the functional ϕ_α is convex. Then the functional

$$\phi(H) = \sup\{\phi_\alpha(H) \mid \alpha \in \mathcal{I}\}$$

is convex.

We now describe some of the relations between sets and functionals that are convex or affine. A functional equality specification formed from an affine functional defines an affine set: if ϕ is affine and $\alpha \in \mathbf{R}$, then

$$\{H \mid \phi(H) = \alpha\}$$

is affine. Similarly, if ϕ is convex and $\alpha \in \mathbf{R}$, then the functional inequality specification

$$\{H \mid \phi(H) \leq \alpha\},$$

called a *sub-level set* of ϕ, is convex.

The converse is not true, however: there are functionals that are not convex, but every sub-level set is convex. Such functionals are our next topic.

6.2.2 Quasiconvex Functionals

Definition 6.5: *A functional ϕ on $\mathcal{H}$ is* quasiconvex *if for each $\alpha \in \mathbf{R}$, the functional inequality specification $\{H \mid \phi(H) \leq \alpha\}$ is convex.*

An equivalent definition of quasiconvexity, that has a form similar to the definition of convexity, is: whenever H, $\tilde{H} \in \mathcal{H}$ and $\lambda \in [0,1]$,

$$\phi(\lambda H + (1-\lambda)\tilde{H}) \leq \max\{\phi(H), \phi(\tilde{H})\}.$$

From definition 6.4 we can see that every convex functional is quasiconvex.

The values of a quasiconvex functional along a line in $\mathcal{H}$ is plotted in figure 6.4; note that this functional is *not* convex. A quasiconvex function of one variable is called *unimodal*, since, roughly speaking, it cannot have two separate regions where it is small.

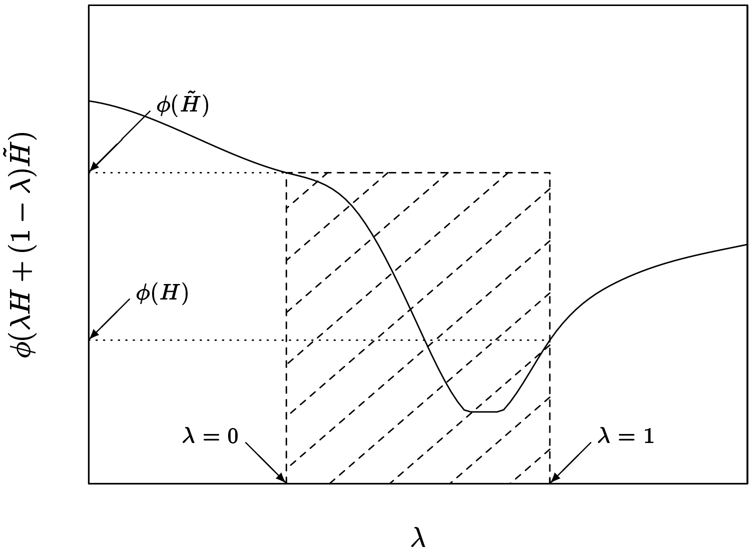


Figure 6.4 A functional ϕ is quasiconvex if for every pair of transfer matrices H and $\tilde{H}$ the graph of ϕ along the line $\lambda H + (1-\lambda)\tilde{H}$ lies on or below the larger of $\phi(H)$ and $\phi(\tilde{H})$, *i.e.*, in the shaded region.

A positive weighted maximum of quasiconvex functionals is quasiconvex, but a positive-weighted sum of quasiconvex functionals need not be quasiconvex.

There is a natural correspondence between quasiconvex functionals and nested families of convex specifications, *i.e.*, linearly ordered parametrized sets of specifications. Given a quasiconvex functional ϕ, we have the family of functional inequality specifications given by $\mathcal{H}^\alpha = \{H \mid \phi(H) \leq \alpha\}$. This family is linearly ordered: $\mathcal{H}^\alpha$ is stronger than $\mathcal{H}^\beta$ if $\alpha \leq \beta$.

Conversely, suppose we are given a family of convex specifications $\mathcal{H}^\alpha$, indexed by a parameter $\alpha \in \mathbf{R}$, such that $\mathcal{H}^\alpha$ is stronger than $\mathcal{H}^\beta$ if $\alpha \le \beta$ (so the family of specifications is linearly ordered). The functional

$$\phi_{\mathrm{family}}(H) \overset{\Delta}{=} \inf \{\alpha \mid H \in \mathcal{H}^\alpha\} \qquad (6.1)$$

($\phi_{\mathrm{family}}(H) \overset{\Delta}{=} \infty$ if $H \notin \mathcal{H}^\alpha$ for all α) simply assigns to a transfer matrix H the index corresponding to the tightest specification that it satisfies, as shown in figure 6.5. This functional ϕ_{family} is easily shown to be quasiconvex, and its sub-level sets are essentially the original specifications:

$$\mathcal{H}^\alpha \subseteq \{H \mid \phi_{\mathrm{family}}(H) \le \alpha\} \subseteq \mathcal{H}^{\alpha+\epsilon}$$

for any positive ϵ.

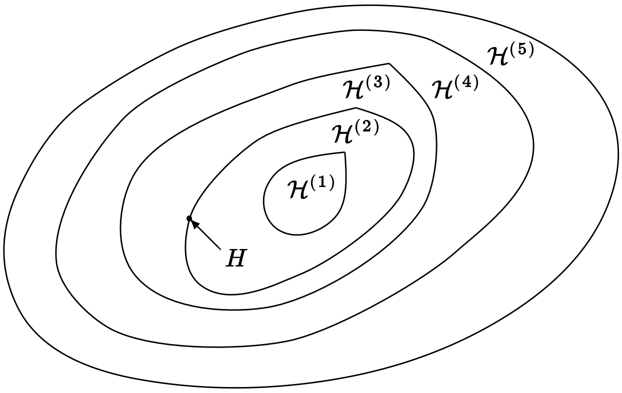


Figure 6.5 Five members of a nested family of convex sets are shown. Such nested families define a quasiconvex function by (6.1); for the given transfer matrix H, we have $\phi_{\mathrm{family}}(H) = 2$.

6.2.3 Linear Transformations

Convex subsets of $\mathcal{H}$ and convex functionals on $\mathcal{H}$ are often defined via linear transformations. Suppose that V is a vector space and $L : \mathcal{H} \to V$ is a linear function. If $\mathcal{V}$ is a convex (or affine) subset of V, then the subset of $\mathcal{H}$ defined by

$$\mathcal{H}_1 = \{H \mid L(H) \in \mathcal{V}\}$$

is convex (or affine). Similarly if ψ is a convex (or quasiconvex or affine) functional on V, then the functional ϕ on $\mathcal{H}$ defined by

$$\phi(H) = \psi(L(H))$$

is convex (or quasiconvex or affine, respectively).

These facts are easily established from the definitions above; we mention a few important examples.

Selecting a Submatrix or Entry of H

A simple but important example is the linear transformation that "selects" a submatrix or entry from H. More precisely, V is the vector space of $p \times q$ transfer matrices and L is given by

$$L(H) = E_z^T H E_w$$

where $E_z \in \mathbf{R}^{n_z \times p}$ and $E_w \in \mathbf{R}^{n_w \times q}$; the columns of E_z and E_w are unit vectors. Thus $L(H)$ is a submatrix of H (or an entry of H if $p = q = 1$); the unit vectors in E_z and E_w select the subsets of regulated variables and exogenous inputs, respectively.

If ψ is a functional on $p \times q$ transfer matrices, a functional ϕ on $\mathcal{H}$ is given by

$$\phi(H) = \psi(L(H)) = \psi(E_z^T H E_w). \tag{6.2}$$

Informally, ϕ results from applying ψ to a certain submatrix of H; a convex (or quasiconvex or affine) functional of an entry or submatrix of H yields a convex (or quasiconvex or affine) functional of H.

To avoid cumbersome notation, we will often describe functionals or specifications that take as argument only an entry or submatrix of H, relying on the reader to extend the functional to $\mathcal{H}$ via (6.2).

Time Domain Responses

Let V consist of scalar signals on $\mathbf{R}_+$, and let L be the transformation that maps a transfer matrix into the unit step response of its i, k entry:

$$L(H) = s$$

where, for $t \geq 0$,

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{H_{ik}(j\omega)}{j\omega} e^{j\omega t}\, d\omega.$$

Since L is linear, we see that a convex constraint on the step response of the i, k entry of a transfer matrix is a convex specification on H.

A similar situation occurs when L maps H into its response to any particular input signal w_{part}:

$$L(H) = H w_{\mathrm{part}}$$

where V is the set of all n_z-component vector signals. If $\mathcal{Z}_{\mathrm{spec}}$ is a convex subset of V, then the specification

$$\{H \mid H w_{\mathrm{part}} \in \mathcal{Z}_{\mathrm{spec}}\}$$

is a convex subset of $\mathcal{H}$.

6.3 Closed-Loop Convex Design Specifications

Many design specifications have the property that the set of closed-loop transfer matrices that satisfy the design specification is convex. We call such design specifications *closed-loop convex*:

Definition 6.6: *A design specification $\mathcal{D}$ is* closed-loop convex *if the set of closed-loop transfer matrices that satisfy $\mathcal{D}$ is convex.*

One of the themes of this book is that *many design specifications are closed-loop convex.*

6.3.1 Open Versus Closed-Loop Formulation

We noted in section 3.1 that it is possible to formulate design specifications in terms of the (open-loop) controller transfer matrix K, instead of the closed-loop transfer matrix H, as we have done. In such a formulation, a design specification is a predicate on candidate controllers, and the feasibility problem is to find a controller K that satisfies a set of design specifications. Such a formulation may seem more natural than ours, since the specifications refer directly to what we design—the controller K.

There is no *logical* difference between these two formulations, since in sensible problems there is a one-to-one correspondence between controllers and the closed-loop transfer matrices that they achieve. The difference appears when we consider *geometrical* concepts such as convexity: a closed-loop convex specification will generally *not* correspond to a convex set of controllers. In chapters 13–16, we will see that convexity of the specifications is the key to computationally tractable solution methods for the controller design problem. The same design problems, if expressed in terms of the controller K, have specifications that are not convex, and hence do not have this great computational advantage.

6.3.2 Norm-Bound Specifications

Many useful functionals are norms of an entry or submatrix of the transfer matrix H. A general form for a design specification is the *norm-bound* specification

$$\|H_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}}\| \le \alpha, \tag{6.3}$$

where $\|\cdot\|$ is some norm on transfer functions or transfer matrices (see chapter 5) and H_{xx} is a submatrix or entry of H (see section 6.2.3). We can interpret $H_{\mathrm{xx}}^{\mathrm{des}}$ as the desired transfer matrix, and the norm $\|\cdot\|$ used in (6.3) as our measure of the deviation of H_{xx} from the desired transfer matrix $H_{\mathrm{xx}}^{\mathrm{des}}$. We often have $H_{\mathrm{xx}}^{\mathrm{des}} = 0$, in which case (6.3) limits the size of H_{xx}.

We now show that the functional

$$\phi(H) = \|H_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}}\|$$

is convex. Let H and $\tilde{H}$ be any two transfer matrices, and let $0 \le \lambda \le 1$. Now, using the triangle inequality and homogeneity property for norms, together with the fact that λ and $1-\lambda$ are nonnegative, we see that

$$\begin{aligned}
\phi(\lambda H + (1-\lambda)\tilde{H}) &= \|\lambda H_{\mathrm{xx}} + (1-\lambda)\tilde{H}_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}}\| \\
&= \|\lambda(H_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}}) + (1-\lambda)(\tilde{H}_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}})\| \\
&\le \|\lambda(H_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}})\| + \|(1-\lambda)(\tilde{H}_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}})\| \\
&= \lambda\|H_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}}\| + (1-\lambda)\|\tilde{H}_{\mathrm{xx}} - H_{\mathrm{xx}}^{\mathrm{des}}\| \\
&= \lambda\phi(H) + (1-\lambda)\phi(\tilde{H}),
\end{aligned}$$

so the functional ϕ is convex. Since ϕ is convex, the norm-bound specification (6.3) is convex. In chapters 8–10 we will see that many specifications can be expressed in the form (6.3), and are therefore closed-loop convex. (We note that the entropy functional defined in section 5.3.5 is also convex, although it is not a norm.)

An important variation on the norm-bound specification (6.3) is the specification

$$\|H_{\mathrm{xx}}\| < \infty, \tag{6.4}$$

which requires that an entry or submatrix of H have a finite norm, as measured by $\|\cdot\|$. This specification is affine, since if H_{xx} and $\tilde{H}_{\mathrm{xx}}$ each satisfy (6.4) and $\lambda \in \mathbf{R}$, then we have

$$\|\lambda H_{\mathrm{xx}} + (1-\lambda)\tilde{H}_{\mathrm{xx}}\| \le |\lambda|\|H_{\mathrm{xx}}\| + |1-\lambda|\|\tilde{H}_{\mathrm{xx}}\| < \infty,$$

so that $\lambda H_{\mathrm{xx}} + (1-\lambda)\tilde{H}_{\mathrm{xx}}$ also satisfies (6.4).

If we use the $\mathbf{H}_\infty$ norm for $\|\cdot\|$, (6.4) is the specification that H_{xx} be stable, *i.e.*, the poles of H_{xx} have negative real parts. Similarly, if we use the a-shifted $\mathbf{H}_\infty$ norm as $\|\cdot\|$, (6.4) is the specification that the poles of H_{xx} have real parts less than $-a$. These specifications are therefore affine.

6.4 Some Examples

In chapters 7–10 we will encounter many specifications and functionals that are affine or convex; in most cases we will simply state that they are affine or convex without a detailed justification. In this section we consider a few typical specifications and functionals for our standard example plant (see section 2.4), and carefully establish that they are affine or convex.

6.4.1 An Affine Specification

Consider the specification that the closed-loop transfer function from the reference input r to y_{p} have unity gain at $\omega = 0$, so that a constant reference input results in

$y_{\mathrm{p}} = r$ in steady-state. The set of 2×3 transfer matrices that corresponds to this specification is

$$\mathcal{H}_{\mathrm{dc}} = \{ H \mid H_{13}(0) = 1 \}. \tag{6.5}$$

We will show that this set is affine. Suppose that H, $\tilde{H} \in \mathcal{H}_{\mathrm{dc}}$, so that $H_{13}(0) = \tilde{H}_{13}(0) = 1$, and $\lambda \in \mathbf{R}$. Then the transfer matrix $H_\lambda = \lambda H + (1-\lambda)\tilde{H}$ satisfies

$$H_{\lambda 13}(0) = \lambda H_{13}(0) + (1-\lambda)\tilde{H}_{13}(0) = \lambda + (1-\lambda) = 1,$$

and hence $H_\lambda \in \mathcal{H}_{\mathrm{dc}}$.

Alternatively, we note that the specification $\mathcal{H}_{\mathrm{dc}}$ can be written as the functional equality constraint

$$\mathcal{H}_{\mathrm{dc}} = \{ H \mid \phi_{\mathrm{dc}}(H) = 1 \},$$

where $\phi_{\mathrm{dc}}(H) \stackrel{\Delta}{=} H_{13}(0)$ is an affine functional.

6.4.2 Convex Specifications

Consider the specification $\mathcal{D}_{\mathrm{act\_eff}}$ introduced in section 3.1: "the RMS deviation of u due to the sensor and process noises is less than 0.1". The corresponding set of transfer matrices is

$$\mathcal{H}_{\mathrm{act\_eff}} = \{ H \mid \phi_{\mathrm{act\_eff}}(H) \le 0.1 \},$$

where we defined the functional

$$\phi_{\mathrm{act\_eff}}(H) \stackrel{\Delta}{=} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \left(|H_{21}(j\omega)|^2 S_{\mathrm{proc}}(\omega) + |H_{22}(j\omega)|^2 S_{\mathrm{sensor}}(\omega) \right) d\omega \right)^{1/2},$$

and S_{proc} and S_{sens} are the power spectral densities of the noises n_{proc} and n_{sensor}, respectively.

To show that $\phi_{\mathrm{act\_eff}}$ is a convex functional we first express it as a norm of the submatrix $[H_{21}\ H_{22}]$ of H:

$$\phi_{\mathrm{act\_eff}}(H) = \left\| \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T H \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W_{\mathrm{proc}} & 0 \\ 0 & W_{\mathrm{sensor}} \end{bmatrix} \right\|_2,$$

where $|W_{\mathrm{proc}}(j\omega)|^2 = S_{\mathrm{proc}}(\omega)$ and $|W_{\mathrm{sensor}}(j\omega)|^2 = S_{\mathrm{sensor}}(\omega)$. From the results of sections 6.2.3 and 6.3.2 we conclude that $\phi_{\mathrm{act\_eff}}$ is a convex functional, and therefore $\mathcal{H}_{\mathrm{act\_eff}}$ is a convex specification.

As another example, consider the specification $\mathcal{D}_{\mathrm{os}}$ introduced in section 3.1: "the step response overshoot from the command to y_{p} is less than 10%". We can see that the corresponding set of transfer matrices, $\mathcal{H}_{\mathrm{os}}$, is convex, using the argument in section 6.2.3 with $i = 1$, $k = 3$ and the convex subset of scalar signals

$$\mathcal{V} = \{ s : \mathbf{R}_+ \to \mathbf{R} \mid s(t) \le 1.1 \text{ for } t \ge 0 \}. \tag{6.6}$$

This is illustrated in figure 6.6.

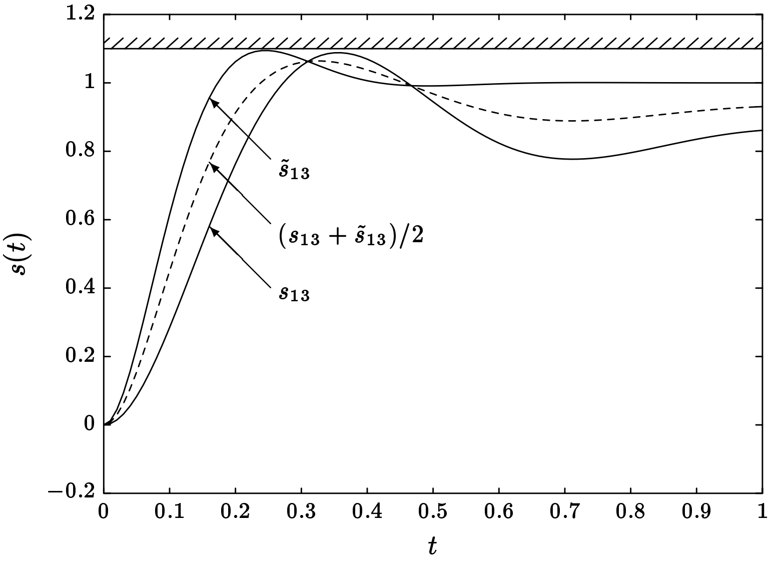


Figure 6.6 The set $\mathcal{V}$ in (6.6) is convex, since if the step responses s_{13} and $\tilde{s}_{13}$ do not exceed 1.1, then their average, $(s_{13}+\tilde{s}_{13})/2$, also does not exceed 1.1. By the argument in section 6.2.3, the specification $\mathcal{H}_{\rm os}$ is convex.

6.4.3 A Quasiconvex Functional

We will see in chapter 8 that several important functionals are quasiconvex, for example, those relating to settling time and bandwidth of a system. We describe one such example here.

Consider the *stability degree* of a transfer matrix, defined by

$$\phi_{\rm stab\_deg}(H) \stackrel{\Delta}{=} \max\{\Re p \mid p \text{ is a pole of } H\}.$$

The functional $\phi_{\rm stab\_deg}$ is quasiconvex since, for each α,

$$\{H \mid \phi_{\rm stab\_deg}(H) \le \alpha\} = \{H \mid \|H\|_{\infty,\beta} < \infty \text{ for } \beta < -\alpha\}$$

(recall that $\|\cdot\|_{\infty,\beta}$ is the β-shifted $\mathbf{H}_\infty$ norm described in section 5.2.7), and we saw above that the latter specification is affine. $\phi_{\rm stab\_deg}$ is *not* convex, however: for most values of λ, we have

$$\phi_{\rm stab\_deg}(\lambda H + (1-\lambda)\tilde{H}) = \max\left\{\phi_{\rm stab\_deg}(H),\ \phi_{\rm stab\_deg}(\tilde{H})\right\}.$$

6.5 Implications for Tradeoffs and Optimization

When the design specifications are convex, and more generally, when a family of design specifications is given by convex functional inequalities, we can say much more about the concepts introduced in chapter 3.

6.5.1 Performance Space Geometry

Suppose that in the multicriterion optimization problem described in section 3.5, the hard constraint $\mathcal{D}_{\text{hard}}$ and the objective functionals $\phi_1, \ldots, \phi_L$ are convex. Then the region of achievable specifications in performance space,

$$\mathcal{A} \triangleq \left\{ a \in \mathbf{R}^L \;\middle|\; \mathcal{D}_{\text{hard}} \wedge \mathcal{D}_{\phi_1}^{a_1} \wedge \ldots \wedge \mathcal{D}_{\phi_L}^{a_L} \text{ is achievable } \right\}, \tag{6.7}$$

is also convex.

To see this, suppose that a and $\tilde{a}$ each correspond to achievable specifications. Then there are transfer matrices H and $\tilde{H}$ that satisfy $\mathcal{D}_{\text{hard}}$ and also, for each k, $1 \leq k \leq L$,

$$\phi_k(H) \leq a_k, \qquad \phi_k(\tilde{H}) \leq \tilde{a}_k.$$

Now suppose that $0 \leq \lambda \leq 1$. The transfer matrix $\lambda H + (1-\lambda)\tilde{H}$ satisfies the hard constraint $\mathcal{D}_{\text{hard}}$, since $\mathcal{D}_{\text{hard}}$ is convex, and also, for each k, $1 \leq k \leq L$,

$$\phi_k(\lambda H + (1-\lambda)\tilde{H}) \leq \lambda a_k + (1-\lambda)\tilde{a}_k,$$

since the functional ϕ_k is convex. But this means that the specification $\lambda a + (1-\lambda)\tilde{a}$ corresponds to an achievable specification, which verifies that $\mathcal{A}$ is convex.

An important consequence of the convexity of the achievable region in performance space is that every Pareto optimal specification is optimal for the classical optimization problem with weighted-sum objective, for some nonnegative weights (*c.f.* figure 3.8). Thus the specifications considered in multicriterion optimization or classical optimization with the weighted-sum or weighted-max objectives are exactly the same as the Pareto optimal specifications.

In the next section we discuss another important consequence of the convexity of $\mathcal{A}$ (which in fact is equivalent to the observation above).

6.6 Convexity and Duality

The dual function ψ defined in section 3.6.2 is always *concave*, meaning that $-\psi$ is convex, even if the objective functionals and the hard constraint are not convex. To see this, we note that for each transfer matrix H that satisfies $\mathcal{D}_{\text{hard}}$, the function ψ_H of λ defined by

$$\psi_H(\lambda) = -\lambda_1\phi_1(H) - \cdots - \lambda_L\phi_L(H)$$

is convex (indeed, linear). Since $-\psi$ can be expressed as the maximum of this family of convex functions, *i.e.*,

$$-\psi(\lambda) = \max\left\{\psi_H(\lambda) \;\middle|\; H \text{ satisfies } \mathcal{D}_{\text{hard}}\right\}, \tag{6.8}$$

it is also a convex function of λ (see section 6.2.1).

The dual function can be used to determine specifications that are not achievable, and therefore represent a limit of performance (recall the discussion in section 1.4.1). Whenever $\lambda \geq 0$, meaning $\lambda \in \mathbf{R}_+^L$, and $a \in \mathbf{R}^L$, we have

$$\psi(\lambda) > a^T\lambda \Longrightarrow \mathcal{D}_{\rm hard} \wedge \mathcal{D}_{\phi_1}^{a_1} \wedge \ldots \wedge \mathcal{D}_{\phi_L}^{a_L} \text{ is unachievable .} \tag{6.9}$$

To establish (6.9), suppose that a corresponds to an achievable specification, *i.e.*, $\mathcal{D}_{\rm hard} \wedge \mathcal{D}_{\phi_1}^{a_1} \wedge \ldots \wedge \mathcal{D}_{\phi_L}^{a_L}$ is achievable. Then there is some transfer matrix $\tilde{H}$ that satisfies $\mathcal{D}_{\rm hard}$ and $\phi_i(\tilde{H}) \leq a_i$ for each i. Therefore whenever $\lambda \geq 0$,

$$\begin{aligned}\psi(\lambda) &= \min\left\{\lambda_1\phi_1(H) + \cdots + \lambda_L\phi_L(H) \;\middle|\; H \text{ satisfies } \mathcal{D}_{\rm hard}\right\}\\ &\leq \lambda_1\phi_1(\tilde{H}) + \cdots + \lambda_L\phi_L(\tilde{H})\\ &\leq \lambda_1 a_1 + \cdots + \lambda_L a_L\\ &= a^T\lambda.\end{aligned}$$

The implication (6.9) follows.

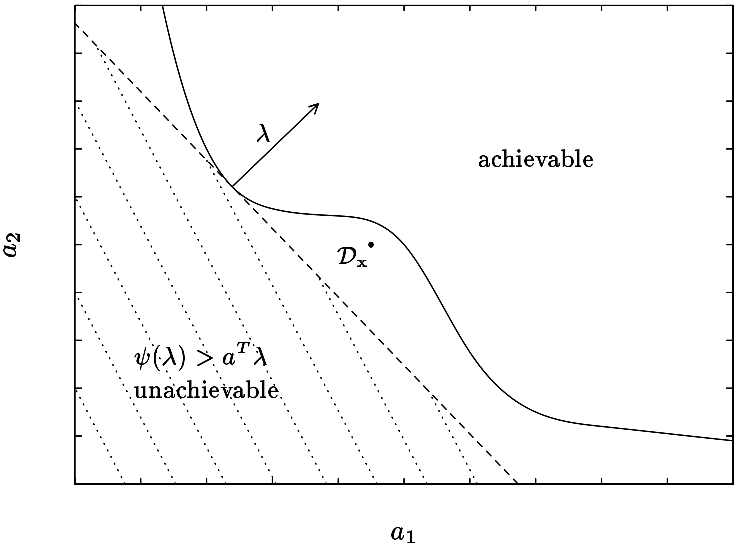


Figure 6.7 The shaded region corresponds to specifications that satisfy $\psi(\lambda) > a^T\lambda$, and hence by (6.9) are unachievable. The specification $\mathcal{D}_{\rm x}$ is unachievable, but cannot be proven unachievable by (6.9), for any choice of λ. See also figure 3.8.

The specifications that (6.9) establishes are unachievable have the simple geometric interpretation shown in figure 6.7, which suggests that when the region of achievable specifications is convex, (6.9) rules out *all* unachievable specifications as λ varies over all nonnegative weight vectors. This intuition is correct, except for a technical condition. More precisely, when $\mathcal{D}_{\rm hard}$ and each objective functional ϕ_i is convex, and the region of achievable specifications in performance space is closed, then we have:

$$\begin{array}{l}\text{the specification corresponding to } a \text{ is achievable}\\ \iff \\ \text{there is no } \lambda \geq 0 \text{ with } \psi(\lambda) > a^T\lambda.\end{array} \tag{6.10}$$

The equivalence (6.10) is called the *convex duality principle*, or a *theorem of alternatives*, since it states that exactly one of two alternative assertions must hold. The geometrical interpretation of the duality principle can be seen from figure 6.8: (6.10) asserts that every specification is either achievable (*i.e.*, lies in the shaded region to the upper right), or can be shown unachievable using (6.9) (*i.e.*, lies in the shaded region to the lower left) for some choice of λ.

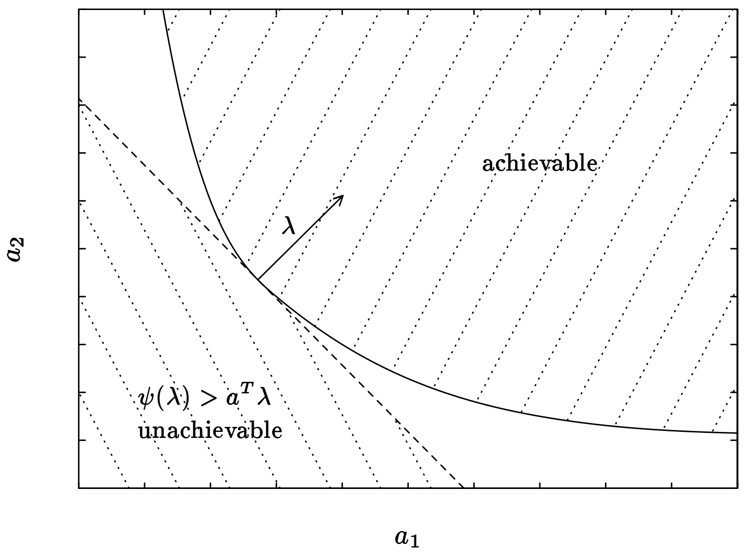


Figure 6.8 Unlike figure 6.7, in this figure the region of achievable specifications is convex, so, by varying λ, (6.9) can be made to rule out every specification that is unachievable.

The convex duality principle (6.10) is often expressed in terms of pairs of constrained optimization problems, *e.g.*, minimizing the functional ϕ_1 subject to the hard constraint and functional inequality specifications for the remaining functionals $\phi_2, \ldots, \phi_L$. We define

$$\alpha_{\text{pri}} \stackrel{\Delta}{=} \min \left\{ \phi_1(H) \;\middle|\; H \text{ satisfies } \mathcal{D}_{\text{hard}} \wedge \mathcal{D}_{\phi_2}^{a_2} \wedge \ldots \wedge \mathcal{D}_{\phi_L}^{a_L} \right\}, \tag{6.11}$$

$$\alpha_{\text{dual}} \stackrel{\Delta}{=} \max \left\{ \psi(\lambda) - a_2\lambda_2 - \cdots - a_L\lambda_L \;\middle|\; \lambda \geq 0,\ \lambda_1 = 1 \right\}. \tag{6.12}$$

The optimization problem on the right-hand side of (6.11) is called the *primal* problem; the right-hand side of (6.12) is called a corresponding *dual* problem. The convex duality principle can then be stated as follows: if the hard constraint and the objective functionals are all convex, then we have $\alpha_{\rm pri} = \alpha_{\rm dual}$. (Here we do not need the technical condition, provided we interpret the min and max as an infimum and supremum, respectively, and use the convention that the minimum of an infeasible problem is ∞.)

For the case when there are no convexity assumptions on the hard constraint or the objectives, then we still have $\alpha_{\rm pri} \geq \alpha_{\rm dual}$, but strict inequality can hold, in which case the difference is called the *duality gap* for the problem.

Notes and References

See the Notes and References for chapters 13 and 14 for general books covering the notion of convexity.

Closed-Loop Convex Specifications

The idea of a closed-loop formulation of controller design specifications has a long history; see section 16.3. In the early work, however, convexity is not mentioned explicitly.

The explicit observation that many design specifications are closed-loop convex can be found in Salcudean's thesis [SAL86], Boyd et al. [BBB88], Polak and Salcudean [PS89], and Boyd, Barratt, and Norman [BBN90].

Optimization

Many of the Notes and References from chapter 3 consider in detail the special case of convex specifications and functionals, and so are relevant. See also the Notes and References from chapters 13 and 14.

Duality

The fact that the nonnegatively weighted-sum objectives yield all Pareto optimal specifications is shown in detail in Da Cunha and Polak [CP67], and in chapter 6 of Clarke [CLA83]. The results on convex duality are standard, and can be found in complete detail in, *e.g.*, Barbu and Precupanu [BP78] or Luc [LUC89].

A Stochastic Interpretation of Closed-Loop Convex Functionals

Jensen's inequality states that if ϕ is a convex functional, $H \in \mathcal{H}$, and H_{stoch} is any zero-mean $\mathcal{H}$-valued random variable, then

$$\phi(H) \le \mathbf{E}\,\phi(H + H_{\mathrm{stoch}}), \tag{6.13}$$

i.e., zero-mean random fluctuations in a transfer matrix increase, on average, the value of a convex functional. In fact, Jensen's inequality characterizes convex functionals: if (6.13) holds for all (deterministic) H and all zero-mean H_{stoch}, then ϕ is convex.

Part III

DESIGN SPECIFICATIONS

Chapter 7

Realizability and Closed-Loop Stability

In this chapter we consider the design specifications of *realizability* and *internal (closed-loop) stability*. The central result is that the set of closed-loop transfer matrices realizable with controllers that stabilize the plant is *affine* and readily described. This description is referred to as the *parametrization of closed-loop transfer matrices achieved by stabilizing controllers.*

7.1 Realizability

An important constraint on the transfer matrix $H \in \mathcal{H}$ is that *it should be the closed-loop transfer matrix achieved by some controller* K, in other words, H should have the form $P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}$ for some K. We will refer to this constraint as *realizability*:

$$\mathcal{H}_{\mathrm{rlzbl}} \stackrel{\Delta}{=} \left\{ H \;\middle|\; H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw} \text{ for some } K \right\} . \tag{7.1}$$

We can think of $\mathcal{H}_{\mathrm{rlzbl}}$ as expressing the *dependencies* among the various closed-loop transfer functions that are entries of H.

As an example, consider the classical SASS 1-DOF control system described in section 2.3.2. The closed-loop transfer matrix is given by

$$\begin{aligned} H &= \begin{bmatrix} \dfrac{P_0}{1+P_0K} & -\dfrac{P_0K}{1+P_0K} & \dfrac{P_0K}{1+P_0K} \\ -\dfrac{P_0K}{1+P_0K} & -\dfrac{K}{1+P_0K} & \dfrac{K}{1+P_0K} \end{bmatrix} \\ &= \begin{bmatrix} P_0(1-T) & -T & T \\ -T & -T/P_0 & T/P_0 \end{bmatrix}, \end{aligned}$$

where T is the classical closed-loop I/O transfer function. Hence if H is the closed-loop transfer matrix realized by some controller K, *i.e.*, $H \in \mathcal{H}_{\rm rlzbl}$, then we must have

$$\begin{aligned} H_{12} &= H_{21}, &(7.2)\\ P_0 H_{22} &= H_{21}, &(7.3)\\ H_{11} - P_0 H_{21} &= P_0, &(7.4)\\ H_{13} &= -H_{12}, &(7.5)\\ H_{23} &= -H_{22}. &(7.6)\end{aligned}$$

It is not hard to show that the five explicit specifications on H given in (7.2–7.6) are not just implied by realizability; they are equivalent to it. Roughly speaking, we have only *one* transfer function that we can design, K, whereas the closed-loop transfer matrix H contains *six* transfer functions. The *five* constraints (7.2–7.6) among these six transfer functions make up the missing degrees of freedom.

The specifications (7.2–7.6) are *affine*, so at least for the classical SASS 1-DOF controller, realizability is an affine specification. In fact, we will see that in the general case, $\mathcal{H}_{\rm rlzbl}$ is affine. Thus, realizability is a closed-loop convex specification.

To establish that $\mathcal{H}_{\rm rlzbl}$ is affine in the general case, we use a simple trick that replaces the inverse appearing in (7.1) with a simpler expression. Given any $n_u \times n_y$ transfer matrix K, we define the $n_u \times n_y$ transfer matrix R by

$$R = K(I - P_{yu}K)^{-1}. \tag{7.7}$$

This correspondence is one-to-one: given any $n_u \times n_y$ transfer matrix R, the $n_u \times n_y$ transfer matrix K given by

$$K = (I + RP_{yu})^{-1}R \tag{7.8}$$

makes sense and satisfies (7.7).

Hence we can express the realizability specification as

$$\mathcal{H}_{\rm rlzbl} = \{H \mid H = P_{zw} + P_{zu}RP_{yw} \ \text{ for some } \ n_u \times n_y \ R\} \,. \tag{7.9}$$

This form of $\mathcal{H}_{\rm rlzbl}$ can be given a simple interpretation, which is shown in figure 7.1. The transfer matrix R can be thought of as the "controller" that would realize the closed-loop transfer matrix H *if there were no feedback through our plant*, *i.e.*, $P_{yu} = 0$ (see figure 7.1(b)). Our trick above is the observation that we can reconstruct the controller K that has the same effect on the true plant as the controller R that operates on the plant with P_{yu} set to zero. Variations on this simple trick will appear again later in this chapter.

From (7.9) we can establish that $\mathcal{H}_{\rm rlzbl}$ is affine. Suppose that H, $\tilde{H} \in \mathcal{H}_{\rm rlzbl}$. Then there are two $n_u \times n_y$ transfer matrices R and $\tilde{R}$ such that

$$\begin{aligned} H &= P_{zw} + P_{zu}RP_{yw},\\ \tilde{H} &= P_{zw} + P_{zu}\tilde{R}P_{yw}.\end{aligned}$$

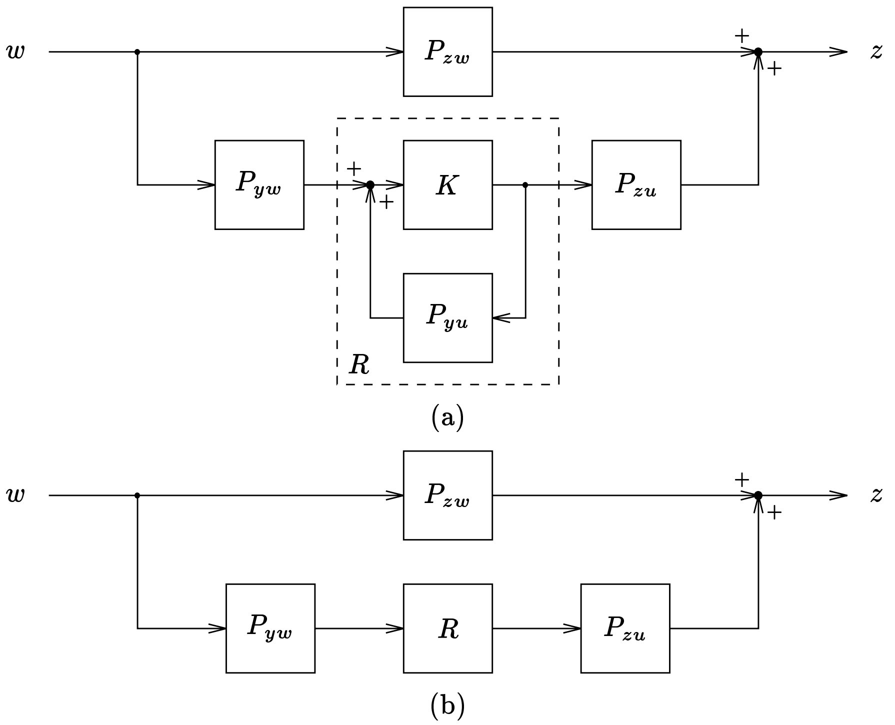


Figure 7.1 The closed-loop transfer matrix H can be realized by the feedback system (a) for some K if and only if it can be realized by the system (b) for some transfer matrix R. In (b) there is no feedback.

Let $\lambda \in \mathbf{R}$. We must show that the transfer matrix $H_\lambda = \lambda H + (1-\lambda)\tilde{H}$ is also realizable as the closed-loop transfer matrix of our plant with some controller. We note that

$$H_\lambda = P_{zw} + P_{zu} R_\lambda P_{yw},$$

where

$$R_\lambda = \lambda R + (1-\lambda)\tilde{R}.$$

This shows that $H_\lambda \in \mathcal{H}_{\rm rlzbl}$.

We can find the controller K_λ that realizes the closed-loop transfer matrix H_λ using the formula (7.8) with R_λ. If K and $\tilde{K}$ are controllers that yield the closed-loop transfer matrices H and $\tilde{H}$, respectively, the controller that realizes the closed-loop transfer matrix H_λ is *not* $\lambda K + (1-\lambda)\tilde{K}$; it is

$$K_\lambda = (A + \lambda B)^{-1}(C + \lambda D) \qquad (7.10)$$

where

$$A = I + \tilde{K}(I - P_{yu}\tilde{K})^{-1}P_{yu},$$

$$\begin{aligned}
B &= K(I - P_{yu}K)^{-1}P_{yu} - \tilde{K}(I - P_{yu}\tilde{K})^{-1}P_{yu},\\
C &= \tilde{K}(I - P_{yu}\tilde{K})^{-1},\\
D &= K(I - P_{yu}K)^{-1} - \tilde{K}(I - P_{yu}\tilde{K})^{-1}.
\end{aligned}$$

The special form of K_λ given in (7.10) is called a *bilinear* or *linear fractional* dependence on λ. We will encounter this form again in section 7.2.6.

7.1.1 An Example

We consider the standard plant example of section 2.4. The step responses from the reference input r to y_{p} and u for each of the controllers $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$ of section 2.4 are shown in figures 7.2(a) and (b). Since $\mathcal{H}_{\mathrm{rlzbl}}$ is affine, we conclude that every transfer matrix on the line passing through $H^{(\mathrm{a})}$ and $H^{(\mathrm{b})}$,

$$\lambda H^{(\mathrm{a})} + (1 - \lambda)H^{(\mathrm{b})}, \qquad \lambda \in \mathbf{R},$$

can be realized as the closed-loop transfer matrix of our standard plant with some controller. Figures 7.2(c) and (d) show the step responses from r to y_{p} and u of five of the transfer matrices on this line.

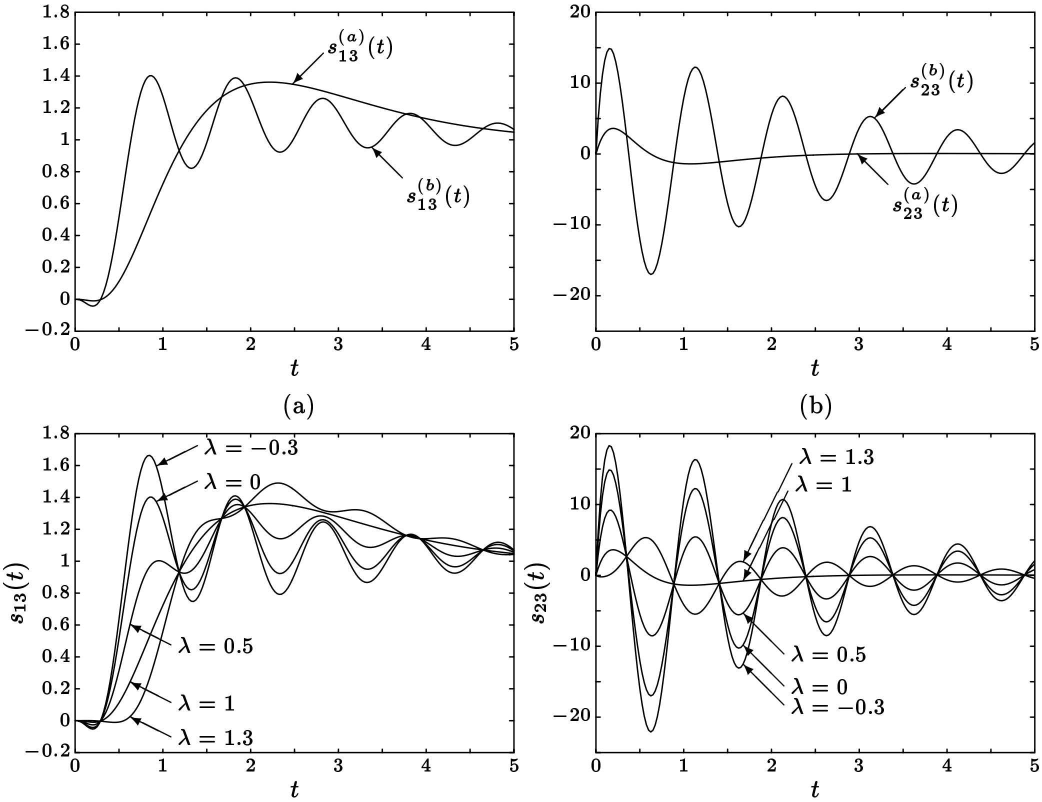
The average of the two closed-loop transfer matrices ($\lambda = 0.5$) is realized by the controller $K_{0.5}$, which can be found from (7.10). Even though both $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$ are 3rd order, $K_{0.5}$ turns out to be 9th order. From this fact we draw two conclusions. First, the specification that H be the transfer matrix achieved by a controller K of order no greater then n,

$$\mathcal{H}_{\mathrm{rlzbl},n} \triangleq \left\{ H \;\middle|\; \begin{array}{l} H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw} \\ \text{for some } K \text{ with order}(K) \le n \end{array} \right\},$$

is not in general convex (whereas the specification $\mathcal{H}_{\mathrm{rlzbl}}$, which puts no limit on the order of K, is convex). Our second observation is that the controller $K_{0.5}$, which yields a closed-loop transfer matrix that is the average of the closed-loop transfer matrices achieved by the controllers $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$, would not have been found by varying the parameters (*e.g.*, numerator and denominator coefficients) in $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$.

7.2 Internal Stability

We remind the reader that a transfer function is *proper* if it has at least as many poles as finite zeros, or equivalently, if it has a state-space realization; a transfer function is *stable* if it is proper and all its poles have negative real part; finally, a transfer matrix is stable if all of its entries are stable. We noted in section 6.4.1 that these are *affine* constraints on a transfer function or transfer matrix.



(a) (b)

(c) (d)

Figure 7.2 (a) shows the closed-loop step responses from r to $y_{\rm p}$ for the standard example with the two controllers $K^{(\rm a)}$ and $K^{(\rm b)}$. (b) shows the step responses from r to u. In (c) and (d) the step responses corresponding to five different values of λ are shown. Each of these step responses is achieved by some controller.

7.2.1 A Motivating Example

Consider our standard example SASS 1-DOF control system described in section 2.4, with the controller

$$K(s) = \frac{36 + 33s}{10 - s}.$$

This controller yields the closed-loop I/O transfer function

$$T(s) = \frac{33s + 36}{s^3 + 10s^2 + 33s + 36} = \frac{33s + 36}{(s+3)^2(s+4)},$$

which is a stable lowpass filter. Thus, we will have $y_{\rm p} \approx r$ provided the reference signal r does not change too rapidly; the controller K yields good tracking of slowly varying reference signals.

If realizability and good tracking of slowly varying reference signals are our *only* specifications, then K is a good controller. The potential problem with this controller can be seen by examining the whole closed-loop transfer matrix,

$$H = \begin{bmatrix} \frac{10-s}{(s+3)^2(s+4)} & -\frac{33s+36}{(s+3)^2(s+4)} & \frac{33s+36}{(s+3)^2(s+4)} \\ -\frac{33s+36}{(s+3)^2(s+4)} & -\frac{s^2(33s+36)(10+s)}{(s+3)^2(s+4)(10-s)} & \frac{s^2(33s+36)(10+s)}{(s+3)^2(s+4)(10-s)} \end{bmatrix}.$$

The entries H_{22} and H_{23}, which are the closed-loop transfer functions from the sensor noise and reference input to u, are *unstable*: for example, a reference input with a very small peak can cause the actuator signal to have a very large peak, a situation that is probably undesirable.

So with the controller K, the I/O transfer function T is quite benign, even desirable, but the closed-loop system will probably have a very large actuator signal. For a classical design approach, in which the requirement that the actuator signal not be too large is not explicitly stated (*i.e.*, it is side information), this example provides a "paradox": the I/O transfer function is acceptable, but the controller is not a reasonable design.

This example shows the importance of considering *all* of the closed-loop transfer functions of interest in a control system, *i.e.*, H, and not just the I/O transfer function T. In our framework, there is no paradox to explain: the controller K can be seen to be unacceptable by examining the whole closed-loop transfer matrix H, and not just T.

This phenomenon of a controller yielding a stable I/O transfer function, but with other closed-loop transfer functions unstable, is called *internal instability.* The qualifier *internal* stresses that the problem with the design cannot be seen by examining the I/O transfer function alone.

Various arguments have been made to explain the "paradox" of this example, *i.e.*, why our controller K is not an acceptable design. They include:

1. The unstable plant zero at $s = 10$ is canceled by the controller pole at $s = 10$. Such unstable pole-zero cancellations between the plant and controller cannot be allowed, because a slight perturbation of the plant zero, *e.g.*, to $s = 9.99$, will cause the I/O transfer function to become unstable.

2. A state-space description of the closed-loop system will be unstable (it will have an eigenvalue of 10), so for most initial conditions, the state will grow larger and larger as time progresses. The unstable mode is unobservable from y_{p}, which is why it does not appear as a pole in the I/O transfer function.

These are valid arguments, but in fact, they correspond to different "new", previously unstated, specifications on our system. In addition to the specifications of realizability and stability of the I/O transfer function, (1) is a sensitivity specification, and (2) requires that other signals (the components of the state vector) should not grow too large when the initial conditions are nonzero. Since these specifications are probably necessary in any real control system, they should be explicitly included in the specifications.

7.2.2 The Desoer-Chan Definition

Desoer and Chan gave a definition of internal stability of a closed-loop system that rules out the problem with our example in section 7.2.1 and other similar pathologies. The definition is:

Definition 7.1: *The closed-loop system with plant P and controller K is* internally stable *if the four transfer matrices*

$$H_{u\nu_1} = K(I - P_{yu}K)^{-1}P_{yu}, \tag{7.11}$$

$$H_{u\nu_2} = K(I - P_{yu}K)^{-1}, \tag{7.12}$$

$$H_{y\nu_1} = (I - P_{yu}K)^{-1}P_{yu}, \tag{7.13}$$

$$H_{y\nu_2} = (I - P_{yu}K)^{-1}, \tag{7.14}$$

are stable. In this case we say the controller K stabilizes *the plant P.*

These transfer matrices can be interpreted as follows. Suppose that ν_1 and ν_2 are an input-referred process noise and a sensor noise, respectively, as shown in figure 7.3. Then the four transfer matrices (7.11–7.14) are the closed-loop transfer matrices from these noises to u and y. So, roughly speaking, internal stability requires that a small process or sensor noise does not result in a very large actuator or sensor signal.

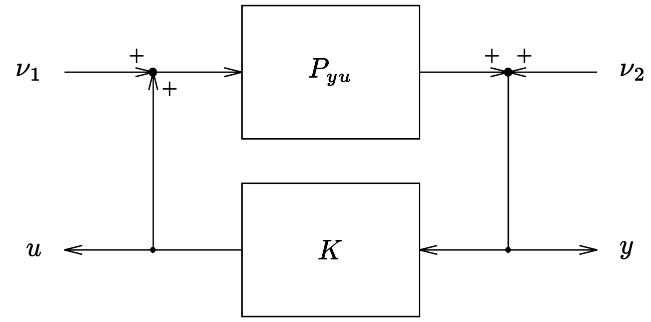


Figure 7.3 Sensor and actuator noises used in the formal definition of internal stability. K stabilizes P if the transfer matrices from ν_1 and ν_2 to u and y are all stable.

The specification of internal stability can be made in our framework as follows. We must include sensor and actuator-referred process noises in the exogenous input signal w, and we must include u and y in the regulated variables vector z, in which case the four transfer matrices (7.11–7.14) appear as submatrices of the closed-loop transfer matrix H. Internal stability is then expressed as the specification that these entries of H be stable, which we mentioned in chapter 6 (section 6.4.1) is an affine specification.

It seems clear that any *sensible* set of design specifications should limit the effect of sensor and process noise on u and y. Indeed, a sensible set of specifications will constrain these four transfer matrices more tightly than merely requiring stability; some norm of these transfer matrices will be constrained. So sensible sets of specifications will generally be *strictly tighter* than internal stability; internal stability will be a redundant specification. We will see examples of this in chapter 12: for the LQG and $\mathbf{H}_\infty$ problems, finiteness of the objective will *imply* internal stability, provided the objectives satisfy certain sensibility requirements. See the Notes and References at the end of this chapter.

7.2.3 Closed-loop Affineness of Internal Stability

We now consider the specification that H is the closed-loop transfer matrix achieved by a controller that stabilizes the plant:

$$\mathcal{H}_{\mathrm{stable}} = \left\{ H \;\middle|\; \begin{array}{l} H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw} \\ \text{for some } K \text{ that stabilizes } P \end{array} \right\}. \tag{7.15}$$

Of course, this is a stronger specification than realizability.

Like $\mathcal{H}_{\mathrm{rlzbl}}$, $\mathcal{H}_{\mathrm{stable}}$ is also affine: if K and $\tilde{K}$ each stabilize P, then for each $\lambda \in \mathbf{R}$, the controller K_λ given by (7.10) also stabilizes P. In the example of section 7.1.1, the controllers $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$ each stabilize the plant. Hence the five controllers that realize the five step responses shown in figure 7.2 also stabilize the plant.

We can establish that $\mathcal{H}_{\mathrm{stable}}$ is affine by direct calculation. Suppose that controllers K and $\tilde{K}$ each stabilize the plant, yielding closed-loop transfer matrices H and $\tilde{H}$, respectively. We substitute the controller K_λ given in (7.10) into the four transfer matrices (7.11–7.14), and after some algebra we find that

$$\begin{aligned}
K_\lambda(I - P_{yu}K_\lambda)^{-1}P_{yu} &= \lambda K(I - P_{yu}K)^{-1}P_{yu} + (1-\lambda)\tilde{K}(I - P_{yu}\tilde{K})^{-1}P_{yu}, \\
K_\lambda(I - P_{yu}K_\lambda)^{-1} &= \lambda K(I - P_{yu}K)^{-1} + (1-\lambda)\tilde{K}(I - P_{yu}\tilde{K})^{-1}, \\
(I - P_{yu}K_\lambda)^{-1}P_{yu} &= \lambda(I - P_{yu}K)^{-1}P_{yu} + (1-\lambda)(I - P_{yu}\tilde{K})^{-1}P_{yu}, \\
(I - P_{yu}K_\lambda)^{-1} &= \lambda(I - P_{yu}K)^{-1} + (1-\lambda)(I - P_{yu}\tilde{K})^{-1}.
\end{aligned}$$

Thus, the four transfer matrices (7.11–7.14) achieved by K_λ are affine combinations of those achieved by K and $\tilde{K}$. Since the right-hand sides of these equations are all stable, the left-hand sides are stable, and therefore K_λ stabilizes P.

We can use the same device that we used to simplify our description of $\mathcal{H}_{\mathrm{rlzbl}}$. The four transfer matrices (7.11–7.14) can be expressed in terms of the transfer matrix R given in (7.7):

$$K(I - P_{yu}K)^{-1}P_{yu} = RP_{yu}, \tag{7.16}$$

$$K(I - P_{yu}K)^{-1} = R, \tag{7.17}$$

$$(I - P_{yu}K)^{-1}P_{yu} = (I + P_{yu}R)P_{yu}, \tag{7.18}$$

$$(I - P_{yu}K)^{-1} = I + P_{yu}R. \tag{7.19}$$

Hence we have

$$\mathcal{H}_{\mathrm{stable}} = \left\{ P_{zw} + P_{zu}RP_{yw} \;\middle|\; \begin{array}{l} RP_{yu}, \\ R, \\ I + P_{yu}R, \\ (I + P_{yu}R)P_{yu} \end{array} \quad \text{are stable} \right\}. \tag{7.20}$$

We will find this description useful.

7.2.4 Internal Stability for a Stable Plant

If the plant P is stable, then in particular P_{yu} is stable. It follows that if R is stable, then so are RP_{yu}, $I + P_{yu}R$, and $(I + P_{yu}R)P_{yu}$. Hence we have

$$\mathcal{H}_{\mathrm{stable}} = \{P_{zw} + P_{zu}RP_{yw} \;\mid R \text{ stable}\}. \tag{7.21}$$

This is just our description of $\mathcal{H}_{\mathrm{rlzbl}}$, with the additional constraint that R be stable.

Given any stable R, the controller that stabilizes P and yields a closed-loop transfer matrix $H = P_{zw} + P_{zu}RP_{yw}$ is

$$K = (I + RP_{yu})^{-1}R. \tag{7.22}$$

Conversely, every controller that stabilizes P can be expressed by (7.22) for some stable R.

7.2.5 Internal Stability via Interpolation Conditions

For the classical SASS 1-DOF control system (see section 2.3.2), the specification of internal stability *can* be expressed in terms of the I/O transfer function T, although the specification is not as simple as stability of T alone (recall our motivating example). We have already noted that a closed-loop transfer matrix H that is realizable has the form

$$\begin{aligned} H &= \begin{bmatrix} P_0(1-T) & -T & T \\ -T & -T/P_0 & T/P_0 \end{bmatrix} \\ &= \begin{bmatrix} P_0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + T \begin{bmatrix} -P_0 & -1 & 1 \\ -1 & -1/P_0 & 1/P_0 \end{bmatrix}, \end{aligned} \tag{7.23}$$

where T is the I/O transfer function. We will describe $\mathcal{H}_{\mathrm{stable}}$ as the set of transfer matrices of the form (7.23), where T satisfies some additional conditions.

Let $p_1, \ldots, p_n$ be the unstable poles of P_0 (*i.e.*, the poles of P_0 that have nonnegative real part) and let $z_1, \ldots, z_m$ be the unstable zeros of P_0 (*i.e.*, the zeros of P_0 that have nonnegative real part); we will assume for simplicity that they are distinct and have multiplicity one. Let r denote the relative degree of P_0, *i.e.*, the difference between the degrees of its numerator and its denominator. Then a transfer matrix H is achievable with a stabilizing controller if and only if it has the form (7.23), where T satisfies:

1. T is stable,
2. $T(p_1) = \cdots = T(p_n) = 1$,
3. $T(z_1) = \cdots = T(z_m) = 0$, and
4. the relative degree of T is at least r.

These conditions are known as the *interpolation conditions* (on T; they can also be expressed in terms of S or other closed-loop transfer functions). The interpolation conditions can be easily understood in classical control terms. Condition 2 reflects the fact that the loop gain P_0K is infinite at the unstable plant poles, and so we have perfect tracking ($T = 1$) at these frequencies. Conditions 3 and 4 reflect the fact that there is no transmission through P_0 at a frequency where P_0 has a zero, and thus $T = 0$ at such a frequency. (Internal stability prohibits an unstable pole or zero of P_0 from being canceled by a zero or pole of K.)

The interpolation conditions are also readily understood in terms of our description of $\mathcal{H}_{\text{stable}}$ given in (7.20). Substituting the plant transfer matrix for the 1-DOF control system (2.10) into (7.20) and using $R = T/P_0$ we get:

$$\mathcal{H}_{\text{stable}} = \{H \text{ of form (7.23)} \mid T,\ T/P_0,\ (1-T)P_0 \text{ are stable}\}. \tag{7.24}$$

Assuming T is stable, T/P_0 will be stable if T vanishes at $z_1, \ldots, z_m$ and in addition T has relative degree at least that of P_0; in other words, T/P_0 is stable if conditions 1, 3, and 4 of the interpolation conditions hold. Similarly, $(1-T)P_0$ will be stable if T is stable and $1 - T$ vanishes at $p_1, \ldots, p_n$ (*i.e.*, conditions 1 and 2 of the interpolation conditions hold).

The interpolation conditions are the earliest description of $\mathcal{H}_{\text{stable}}$, and date back at least to 1955 (see the Notes and References at the end of this chapter for details).

7.2.6 General Free Parameter Representation

In the general case there is a *free parameter* description of the set of closed-loop transfer matrices achievable with stabilizing controllers:

$$\mathcal{H}_{\text{stable}} = \{T_1 + T_2QT_3 \mid Q \text{ is a stable } n_u \times n_y \text{ transfer matrix}\}, \tag{7.25}$$

where T_1, T_2, and T_3 are certain stable transfer matrices that depend on the plant. Q is referred to as the *parameter* in (7.25), not in the sense of a real number that is to be designed (*e.g.*, the integrator time constant in a PI controller), but rather in the sense that it is the free parameter in the description (7.25). We saw a special case of this form already in the example of the stable plant—in that case, P_{zw}, P_{zu} and P_{yw} are possible choices for T_1, T_2 and T_3, respectively.

The controller that stabilizes the plant and yields closed-loop transfer matrix $H = T_1 + T_2QT_3$ has the linear fractional form

$$K_Q = (A + BQ)^{-1}(C + DQ) \tag{7.26}$$

where A, B, C, D are certain stable transfer matrices related to T_1, T_2, and T_3. Thus the dependence of K_Q on Q is bilinear (*c.f.* equation (7.10)).

It is not hard to understand the basic idea behind the free parameter representation (7.25) of the set of achievable closed-loop transfer matrices (7.20), although a complete derivation is fairly involved (see the Notes and References at the end of this chapter).

We consider the subspace of $n_u \times n_y$ transfer matrices given by

$$\mathcal{S} = \{S \mid S,\ P_{yu}S,\ SP_{yu},\ P_{yu}SP_{yu} \text{ are stable}\}.$$

The basic idea is that an $S \in \mathcal{S}$ must have the appropriate zeros that cancel the unstable poles of P_{yu}. These zeros can be arranged by multiplying a stable transfer matrix on the left and right by appropriate stable transfer matrices D and $\tilde{D}$:

$$\mathcal{S} = \left\{ DQ\tilde{D} \;\middle|\; Q \text{ is stable} \right\}.$$

D and $\tilde{D}$ are not unique, but any suitable choice has the property that if Q is stable, then each of $DQ\tilde{D}$, $P_{yu}DQ\tilde{D}$, $DQ\tilde{D}P_{yu}$, and $P_{yu}DQ\tilde{D}P_{yu}$ are stable. We shall not derive the form of D and $\tilde{D}$.

By comparing (7.20) and (7.25) we see that one possible choice for T_2 and T_3 in (7.25) is

$$\begin{aligned} T_2 &= P_{zu}D \\ T_3 &= \tilde{D}P_{yw}. \end{aligned}$$

T_1 can be taken to be *any* closed-loop transfer matrix achieved by some stabilizing controller. The references cited at the end of this chapter contain the complete details.

7.3 Modified Controller Paradigm

The descriptions of $\mathcal{H}_{\text{stable}}$ given in the previous sections can be given an interpretation in terms of *modifying* a given nominal controller that stabilizes the plant.

Given one controller $K_{\rm nom}$ that stabilizes the plant, we can construct a large *family* of controllers that stabilize the plant, just as the formula (7.10) shows how to construct a one-parameter family of controllers that stabilize the plant.

The construction proceeds as follows:

- We *modify* or *augment* the nominal controller $K_{\rm nom}$ so that it produces an auxiliary output signal e (of the same size as y) and accepts an auxiliary input signal v (of the same size as u) as shown in figure 7.4. This augmentation is done in such a way that the closed-loop transfer matrix from v to e is zero while the open-loop controller transfer matrix from y to u remains $K_{\rm nom}$.
- We connect a stable $n_u \times n_y$ transfer matrix Q from e to v as shown in figure 7.5, and collect $K_{\rm nom}$ and Q together to form a new controller, K.

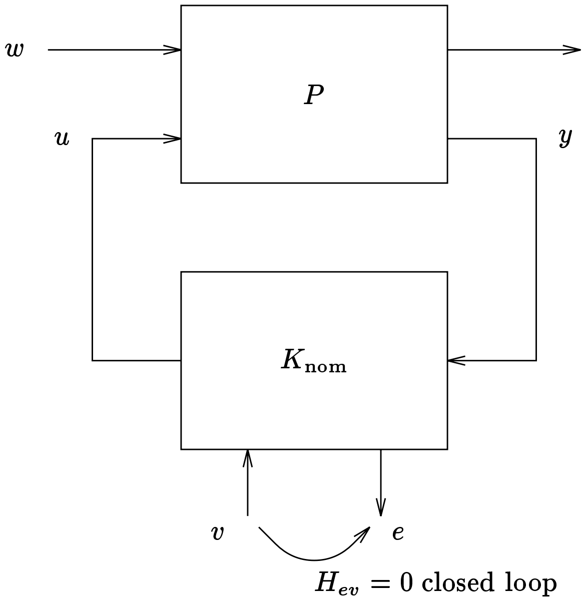


Figure 7.4 The nominal controller $K_{\rm nom}$ is augmented to produce a signal e and accept a signal v. The closed-loop transfer function from v to e is 0.

The intuition is that K should also stabilize P, since the Q system we added to $K_{\rm nom}$ is stable and "sees no feedback", and thus cannot destabilize our system. However, Q *can* change the closed-loop transfer matrix H. To see how Q affects the closed-loop transfer matrix H, we define the following transfer matrices in figure 7.4:

- U_1 is the closed-loop transfer matrix from w to z,
- U_2 is the closed-loop transfer matrix from v to z,
- U_3 is the closed-loop transfer matrix from w to e.

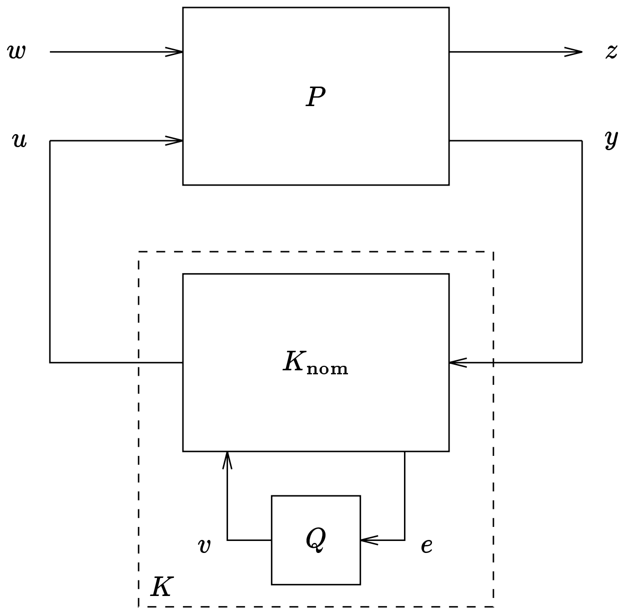


Figure 7.5 Modification of nominal controller K_{nom} with a stable transfer matrix Q.

Since the transfer matrix from v to e in figure 7.4 is zero, we can redraw figure 7.5 as figure 7.6. Figure 7.6 can then be redrawn as figure 7.7, which makes it clear that the closed-loop transfer matrix H resulting from our modified controller K is simply

$$H = U_1 + U_2 Q U_3, \tag{7.27}$$

which must be stable because Q, U_1, U_2 and U_3 are all stable.

It can be seen from (7.27) that as Q varies over all stable transfer matrices, H sweeps out the following affine set of closed-loop transfer matrices:

$$\mathcal{H}_{\mathrm{mcp}} = \{U_1 + U_2 Q U_3 \mid Q \text{ stable}\}.$$

Of course, $\mathcal{H}_{\mathrm{mcp}} \subseteq \mathcal{H}_{\mathrm{stable}}$. This means that a (possibly incomplete) family of stabilizing controllers can be generated from the (augmented) nominal controller using this *modified controller paradigm.*

If the augmentation of the nominal controller is done properly, then the modified controller paradigm yields *every* controller that stabilizes the plant P, in other words, $\mathcal{H}_{\mathrm{mcp}} = \mathcal{H}_{\mathrm{stable}}$. In this case, U_1, U_2, and U_3 can be used as T_1, T_2, and T_3 in the free parameter representation of $\mathcal{H}_{\mathrm{stable}}$ given in equation (7.25).

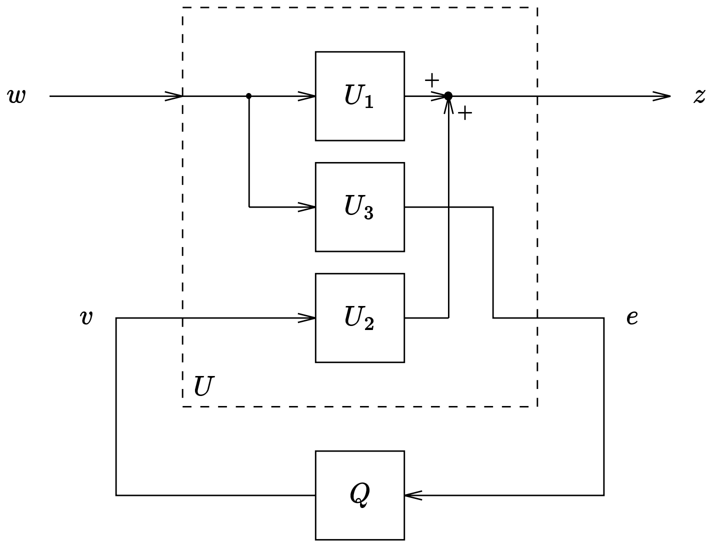


Figure 7.6 Figure 7.5 redrawn.

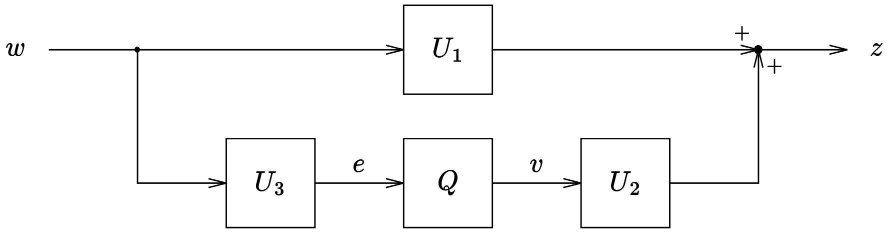


Figure 7.7 Figure 7.6 redrawn.

7.3.1 Modified Controller Paradigm for a Stable Plant

As an example of the modified controller paradigm, we consider the special case of a stable plant (see section 7.2.4). Since the plant is stable, the nominal controller $K_{\mathrm{nom}} = 0$ stabilizes the plant.

How do we modify the zero controller to produce e and accept v? One obvious method is to add v into u, and let e be the difference between y and $P_{yu}u$, which ensures that the closed-loop transfer matrix from v to e is zero, as required by the modified controller paradigm. This is shown in figure 7.8.

From figure 7.8 we see that

$$\begin{aligned} U_1 &= P_{zw}, \\ U_2 &= P_{zu}, \\ U_3 &= P_{yw}. \end{aligned}$$

To apply the second step of the modified controller paradigm, we connect a stable

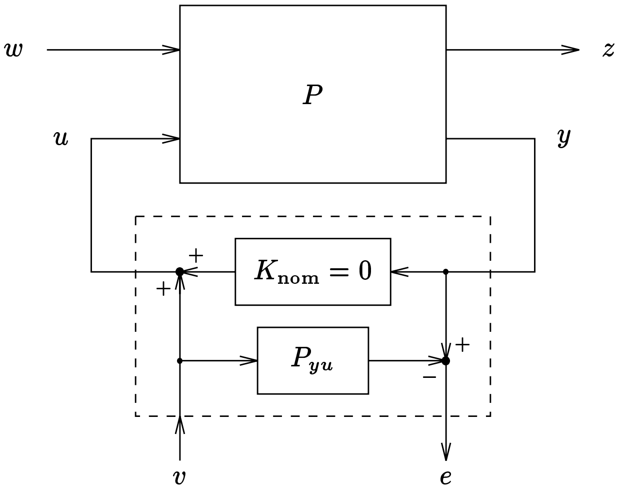


Figure 7.8 One method of extracting e and injecting v when the plant is stable.

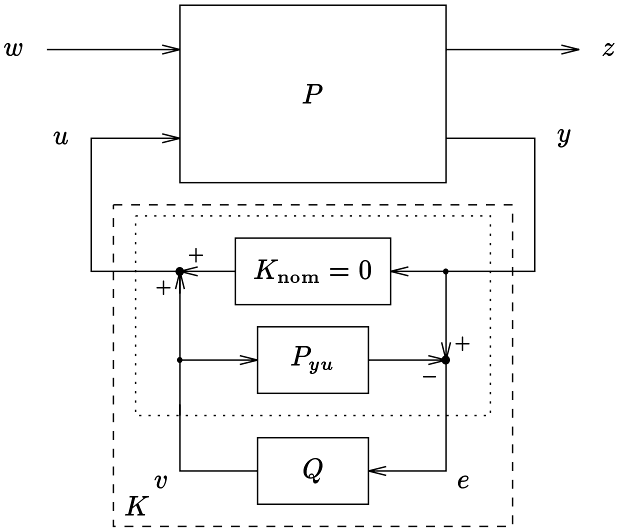


Figure 7.9 The modified controller paradigm, for a stable plant, using the augmented controller shown in figure 7.8.

Q as shown in figure 7.9. so that the closed-loop transfer matrix is

$$H = U_1 + U_2 Q U_3.$$

Thus the set of closed-loop transfer matrices achievable by the modified controller shown in figure 7.9 is

$$\mathcal{H}_{\rm mcp} = \{P_{zw} + P_{zu} Q P_{yw} \ \mid Q \text{ stable}\}\ .$$

The expression here for $\mathcal{H}_{\rm mcp}$ is the same as the expression for $\mathcal{H}_{\rm stable}$ in equation (7.21) in section 7.2.4. So in this case the modified controller paradigm generates all stabilizing controllers: any stabilizing controller K for a stable plant P can be implemented with a suitable stable Q as shown in figure 7.9.

The reader can also verify that the connection of Q with the augmented nominal controller yields $K = (I + QP_{yu})^{-1}Q$—exactly the same formula as (7.22) with Q substituted for R.

7.4 A State-Space Parametrization

A general method of applying the modified controller paradigm starts with a nominal controller that is an *estimated-state feedback.* The estimated-state-feedback controller is given by

$$u = -K_{\rm sfb}\hat{x}, \tag{7.28}$$

where $K_{\rm sfb}$ is some appropriate matrix (the *state-feedback gain*) and $\hat{x}$ is an estimate of the component of x due to u, governed by the observer equation

$$\dot{\hat{x}} = A_P\hat{x} + B_u u + L_{\rm est}(y - C_y\hat{x}), \tag{7.29}$$

where $L_{\rm est}$ is some appropriate matrix (the *estimator gain*). The transfer matrix of this controller is thus

$$K_{\rm nom}(s) = -K_{\rm sfb}(sI - A_P + B_u K_{\rm sfb} + L_{\rm est}C_y)^{-1}L_{\rm est}.$$

$K_{\rm nom}$ will stabilize P provided $K_{\rm sfb}$ and $L_{\rm est}$ are chosen such that $A_P - B_u K_{\rm sfb}$ and $A_P - L_{\rm est}C_y$ are stable, which we assume in the sequel.

To augment this estimated-state-feedback nominal controller, we inject v into u, *before the observer tap*, meaning that (7.28) is replaced by

$$u = -K_{\rm sfb}\hat{x} + v, \tag{7.30}$$

and therefore the signal v does not induce any observer error. For the signal e we take the *output prediction error*:

$$e = y - C_y\hat{x}. \tag{7.31}$$

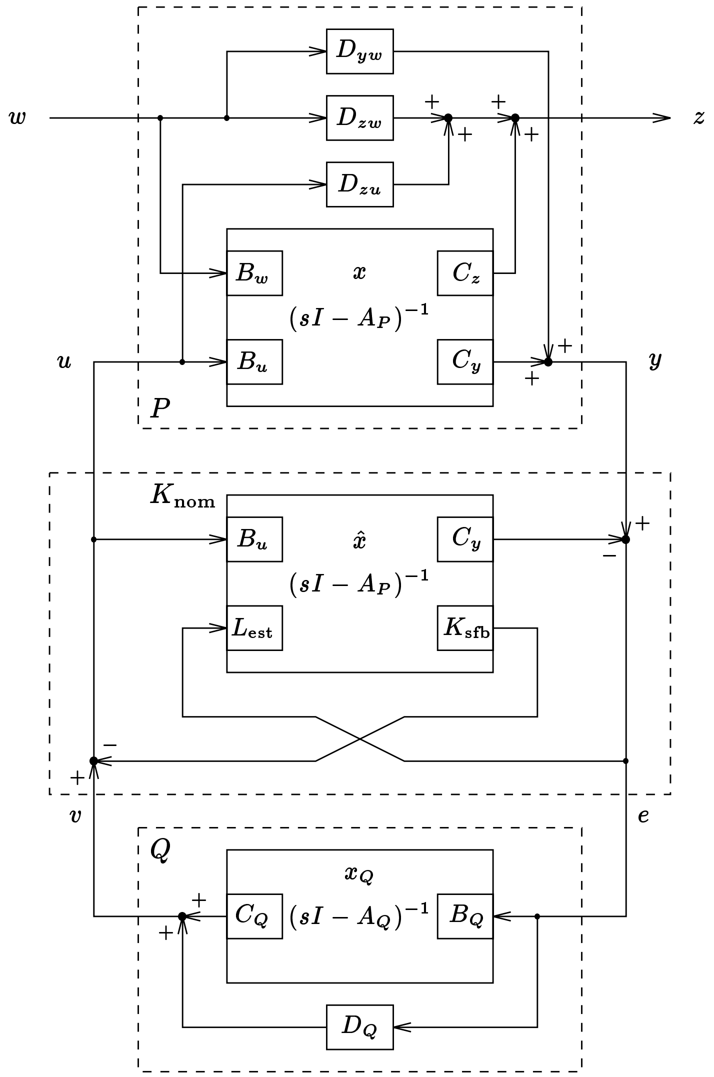


Figure 7.10 The modified controller paradigm as applied to a nominal estimated-state-feedback controller K_{nom}. v is added to the actuator signal u before the observer tap, and e is the output prediction error. With the stable Q realization added, the modified controller is called an observer-based controller.

This is shown in figure 7.10.

The requirement that the closed-loop transfer matrix from v to e be zero is satisfied because the observer state error, $x - \hat{x}$, is uncontrollable from v, and therefore the transfer matrix from v to $x - \hat{x}$ is zero. The transfer matrix from v to e is C_y times this last transfer matrix, and so is zero.

Applying the modified controller paradigm to the estimated-state-feedback controller yields the *observer-based controller* shown in figure 7.10. The observer-based controller is just an estimated-state-feedback controller, with the output prediction error processed through a stable transfer matrix Q and added to the actuator signal before the observer tap.

In fact, this augmentation is such that the modified controller paradigm yields every controller that stabilizes the plant. Every stabilizing controller can be realized (likely nonminimally) as an observer-based controller for some choice of stable transfer matrix Q.

From the observer-based controller we can form simple state-space equations for the parametrization of all controllers that stabilize the plant, and all closed-loop transfer matrices achieved by controllers that stabilize the plant.

The state-space equations for the augmented nominal controller are, from (7.29–7.31),

$$\dot{\hat{x}} = (A_P - B_u K_{\text{sfb}} - L_{\text{est}} C_y)\hat{x} + L_{\text{est}} y + B_u v \tag{7.32}$$

$$u = -K_{\text{sfb}}\hat{x} + v \tag{7.33}$$

$$e = y - C_y \hat{x}. \tag{7.34}$$

The state-space equations for the closed-loop system with the augmented controller are then found by eliminating u and y from (7.32–7.34) and the plant equations (2.19–2.21) of section 2.5:

$$\begin{aligned}
\dot{x} &= A_P x - B_u K_{\text{sfb}}\hat{x} + B_w w + B_u v \\
\dot{\hat{x}} &= L_{\text{est}} C_y x + (A_P - B_u K_{\text{sfb}} - L_{\text{est}} C_y)\hat{x} + L_{\text{est}} D_{yw} w + B_u v \\
z &= C_z x - D_{zu} K_{\text{sfb}}\hat{x} + D_{zw} w + D_{zu} v \\
e &= C_y x - C_y \hat{x} + D_{yw} w.
\end{aligned}$$

The transfer matrices T_1, T_2, and T_3 can therefore be realized as

$$\begin{bmatrix} T_1(s) & T_2(s) \\ T_3(s) & 0 \end{bmatrix} = C_T(sI - A_T)^{-1} B_T + D_T, \tag{7.35}$$

where

$$\begin{aligned}
A_T &= \begin{bmatrix} A_P & -B_u K_{\text{sfb}} \\ L_{\text{est}} C_y & A_P - B_u K_{\text{sfb}} - L_{\text{est}} C_y \end{bmatrix} \\
B_T &= \begin{bmatrix} B_w & B_u \\ L_{\text{est}} D_{yw} & B_u \end{bmatrix}
\end{aligned}$$

$$C_T = \begin{bmatrix} C_z & -D_{zu}K_{\mathrm{sfb}} \\ C_y & -C_y \end{bmatrix}$$

$$D_T = \begin{bmatrix} D_{zw} & D_{zu} \\ D_{yw} & 0 \end{bmatrix}.$$

If Q has state-space realization

$$\dot{x}_Q = A_Q x_Q + B_Q e \tag{7.36}$$

$$v = C_Q x_Q + D_Q e, \tag{7.37}$$

then a state-space realization of the observer-based controller can be found by eliminating e and v from the augmented controller equations (7.32–7.34) and the Q realization (7.36–7.37):

$$\begin{aligned}\dot{\hat{x}} = (A_P - B_u K_{\mathrm{sfb}} - L_{\mathrm{est}} C_y - B_u D_Q C_y)\hat{x} \\ + B_u C_Q x_Q + (L_{\mathrm{est}} + B_u D_Q) y\end{aligned} \tag{7.38}$$

$$\dot{x}_Q = -B_Q C_y \hat{x} + A_Q x_Q + B_Q y \tag{7.39}$$

$$u = -(K_{\mathrm{sfb}} + D_Q C_y)\hat{x} + C_Q x_Q + D_Q y \tag{7.40}$$

so that

$$K(s) = C_K (sI - A_K)^{-1} B_K + D_K, \tag{7.41}$$

where

$$A_K = \begin{bmatrix} A_P - B_u K_{\mathrm{sfb}} - L_{\mathrm{est}} C_y - B_u D_Q C_y & B_u C_Q \\ -B_Q C_y & A_Q \end{bmatrix}$$

$$B_K = \begin{bmatrix} L_{\mathrm{est}} + B_u D_Q \\ B_Q \end{bmatrix}$$

$$C_K = \begin{bmatrix} -K_{\mathrm{sfb}} - D_Q C_y & C_Q \end{bmatrix}$$

$$D_K = D_Q.$$

Some algebra verifies that the closed-loop transfer matrix H given by (2.27) of section 2.5 does indeed equal $T_1 + T_2 Q T_3$.

7.5 Some Generalizations of Closed-Loop Stability

So far, our discussion in this chapter has been built around the notion of a stable transfer function, *i.e.*, a transfer function for which each pole p satisfies $\Re p < 0$. We saw in chapter 5 that stability is equivalent to several other important properties of a transfer function, *e.g.*, finiteness of its peak or RMS gain. In fact, the material in this chapter can be adapted to various generalized notions of stability. The references discuss these ideas in a general setting; we will describe a specific example in more detail.

Instead of stability, we consider the requirement that each pole of a transfer function should satisfy $\Re p \leq -0.1$ and $|\Im p| \leq -\Re p$. We will call such transfer functions $\mathcal{G}$-stable ($\mathcal{G}$ stands for *generalized*). In classical terminology, $\mathcal{G}$-stability guarantees a stability degree and a minimum damping ratio for a transfer function, as illustrated in figure 7.11.

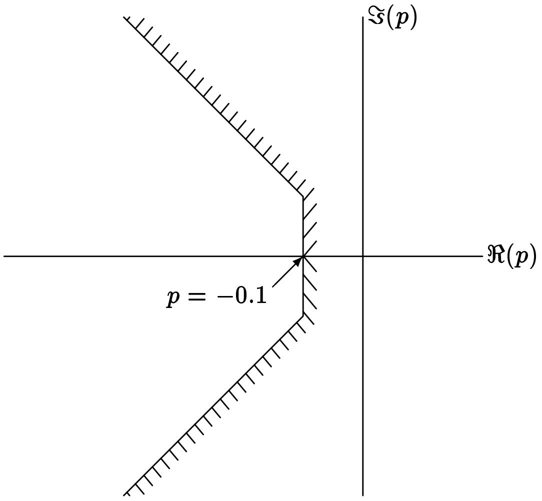


Figure 7.11 A transfer function is $\mathcal{G}$-stable if its poles lie in the region to the left. In classical control terminology, such transfer functions have a stability degree of at least 0.1 and a damping ratio of at least $1/\sqrt{2}$. All of the results in this chapter can be adapted to this generalized notion of stability.

We say that a controller K $\mathcal{G}$-stabilizes the plant P if every entry of the four transfer matrices (7.11–7.14) is $\mathcal{G}$-stable (*c.f.* definition 7.1). It is not hard to show that $\mathcal{H}_{\mathcal{G}\text{-stable}}$, the specification that the closed-loop transfer matrix is achievable by a $\mathcal{G}$-stabilizing controller, is affine; in fact, we have

$$\mathcal{H}_{\mathcal{G}\text{-stable}} = \left\{ P_{zw} + P_{zu}RP_{yw} \;\middle|\; \begin{array}{l} RP_{yu}, \\ R, \\ I + P_{yu}R, \\ (I + P_{yu}R)P_{yu} \end{array} \quad \text{are } \mathcal{G}\text{-stable} \right\},$$

which is just (7.20), with "$\mathcal{G}$-stable" substituted for "stable".

For the SASS 1-DOF control system, the specification $\mathcal{H}_{\mathcal{G}\text{-stable}}$ can be expressed in terms of the interpolation conditions described in section 7.2.5, with the following modifications: condition (1) becomes "T is $\mathcal{G}$-stable", and the list of poles and zeros of P_0 must be expanded to include any poles and zeros that are $\mathcal{G}$-unstable, *i.e.*, lie in the right-hand region in figure 7.11.

There is a free parameter representation of $\mathcal{H}_{\mathcal{G}\text{-stable}}$:

$$\mathcal{H}_{\mathcal{G}\text{-stable}} = \{T_1 + T_2 Q T_3 \ \mid Q \text{ is a } \mathcal{G}\text{-stable } n_u \times n_y \text{ transfer matrix}\},$$

which is (7.25) with "$\mathcal{G}$-stable" substituted for "stable". This free parameter representation can be developed from state-space equations exactly as in section 7.4, provided the state-feedback and estimator gains are chosen such that $A_P - B_u K_{\text{sfb}}$ and $A_P - L_{\text{est}} C_y$ are $\mathcal{G}$-stable, *i.e.*, their eigenvalues lie in the left-hand region of figure 7.11.

Notes and References

Realizability

Freudenberg and Looze [FL88] refer to some of the realizability constraints (*e.g.*, $S + T = 1$ in the classical 1-DOF control system) as *algebraic constraints* on closed-loop transfer functions. In contrast, they refer to constraints imposed by stability of T and the interpolation conditions as *analytic.*

Internal Stability

Arguments forbidding unstable pole-zero cancellations can be found in any text on classical control, *e.g.* [OGA90, P606–607]. For MAMS plants, finding a suitable definition of a zero is itself a difficult task, so this classical unstable cancellation rule was not easily extended. An extensive discussion of internal stability and state-space representations can be found in Kailath [KAI80, P175] or Callier and Desoer [CD82A].

Desoer and Chan's definition appears in [DC75]. Their definition has been widely used since, *e.g.*, in [FRA87, P15–17] and [VID85, P99–108].

Parametrization for Stable Plants

The parametrization given in (7.21) appears for example in the articles [ZAM81], [DC81A], [BD86], and chapter 8 of Callier and Desoer [CD82A]. In process control, the parametrization is called the *internal model principle*, since the controller K in figure 7.9 contains a model of P_{yu}; see Morari and Zafirou [MZ89, CH.3].

Parametrization via Interpolation Conditions

An early version of the interpolation conditions appears in Truxal's 1955 book [TRU55, P308–309]. There he states that if P_0 has an unstable zero, so should the closed-loop I/O transfer function T. He does not mention unstable plant poles, and his reasoning is not quite right (see [BBN90]).

The first essentially correct and explicit statement of the interpolation conditions appears in a 1956 paper by Bertram on discrete-time feedback control [BER56]. He states:

> In summary, for [the classical SASS 1-DOF control system], the following design restrictions must be considered.
>
> - Any zeros of the plant on or outside the unit circle in the z-plane must be contained in $[T(z)]$.
> - Any poles of the plant on or outside the unit circle in the z-plane must be contained in $[1 - T(z)]$.

He does not explicitly state that these conditions are not only necessary, but also sufficient for closed-loop stability; but it is implicit in his design procedure.

Another early exposition of the interpolation conditions can be found in chapter 7 of Ragazzini and Franklin's 1958 book [RF58, P157–158]. The equivalent interpolation conditions for continuous-time systems first appear in a 1958 paper by Bigelow [BIG58].

A recent paper that uses the interpolation conditions is Zames and Francis [ZF83]. Interpolation conditions for MAMS plants appear in [AS84].

Parametrization for MAMS Plants

The results on parametrization of achievable closed-loop transfer matrices in the multiple-actuator, multiple-sensor case depend on *factorizations* of transfer matrices. Early treatments use factorization of transfer matrices in terms of matrices of polynomials; see *e.g.*, [ROS70] and [WOL74]; extensive discussion appears in [KAI80]. The first parametrization of closed-loop transfer matrices that can be achieved with stabilizing controllers appears in Youla, Jabr, and Bongiorno's articles on Wiener-Hopf design [YJB76, YBJ76]. For discrete-time systems, the parametrization appears in the book by Kučera [KUC79].

A more recent version of the parametrization uses factorization in terms of stable transfer matrices, and appears first in Desoer, Liu, Murray and Saeks [DLM80]. The book by Vidyasagar [VID85, CH.3,5] contains a complete treatment of the parametrization of achievable closed-loop transfer matrices in terms of stable factorizations. A state-space parametrization can be found in Francis [FRA87, CH.4] or Vidyasagar [VID85].

Parametrization Using Observer-Based Controller

The observer-based controller parametrization was first pointed out by Doyle [DOY84]; it also appears in Anderson and Moore [AM90, §9.2], Maciejowski [MAC89, §6.4], and a recent article by Moore, Glover, and Telford [MGT90].

Why We Hear So Much About Stability

We mentioned in section 7.2.2 that any sensible set of design specifications will constrain the four critical transfer matrices more tightly than merely requiring stability. For example, the specifications may include specific finite limits on some norm of the four critical transfer matrices, such as

$$\|H_{u\nu_1}\|_\infty \le a_{11}, \quad \|H_{u\nu_2}\|_\infty \le a_{12}, \quad \|H_{y\nu_1}\|_\infty \le a_{21}, \quad \|H_{y\nu_2}\|_\infty \le a_{22}, \tag{7.42}$$

whereas internal stability only requires that these norms be finite:

$$\|H_{u\nu_1}\|_\infty < \infty, \quad \|H_{u\nu_2}\|_\infty < \infty, \quad \|H_{y\nu_1}\|_\infty < \infty, \quad \|H_{y\nu_2}\|_\infty < \infty. \tag{7.43}$$

In any particular problem, a design in which these transfer matrices are extremely large but stable is just as unacceptable as a design in which one or more of these transfer matrices is actually unstable. So in any particular problem, the "qualitative" (affine) specification of internal stability (7.43) will need to be replaced by a stronger "quantitative" (convex but not affine) specification such as (7.42).

We see so much discussion about the qualitative specification of internal stability for historical reasons. In Newton, Gould, and Kaiser [NGK57, P21] we find

> In the classical view, a feedback control problem could be identified almost always as a stability problem. To the early workers in the field, the problem of assuring stability was nearly always the foremost consideration. ... [A bad controller] caused the system to exhibit sustained oscillations of the output even though the input was quiescent. This phenomenon, often called hunting, so plagued the control engineer that even to the present time [1957] it has all but dwarfed the many other aspects of the feedback control problem.

In Black [BLA34] we find

It is far from a simple proposition to employ feedback in this way because of the very special control required of phase shifts in the amplifier and circuits, not only throughout the useful frequency band but also for a wide range of frequencies above and below this band. Unless these relations are maintained, singing will occur, usually at frequencies outside the useful range. Once having achieved a design, however, in which proper phase relations are secured, experience has demonstrated that the performance obtained is perfectly reliable.

Chapter 8

Performance Specifications

In this chapter we consider in detail *performance specifications*, which limit the response of the closed-loop system to the various commands and disturbances that may act on it. We show that many of these performance specifications are closed-loop convex.

We organize our discussion of performance specifications by their *meaning* or *purpose* (in the context of controller design), and not by the mathematical form of the constraints. In fact we shall see that performance specifications with different meanings, such as a limit on errors to commands, a minimum acceptable level of regulation, and a limit on actuator effort, can be expressed in similar forms as limits on the size of a particular submatrix of the closed-loop transfer matrix H. For this reason our discussion of these types of specifications will become briefer as the chapter progresses and the reader can refer back to chapter 5 or other specifications that have a similar form.

To facilitate this organization of performance specifications by meaning, we partition the exogenous input vector w as follows:

$$w = \left[\begin{array}{c} w_{\mathrm{c}} \\ w_{\mathrm{d}} \\ w_{\mathrm{etc}} \end{array}\right] \begin{array}{ll} \updownarrow n_{\mathrm{c}} & \text{commands} \\ \updownarrow n_{\mathrm{d}} & \text{disturbances} \\ & \text{other components of } w \end{array}$$

The n_{c} components of w_{c} are the command, reference, or set-point signals—the "input" in classical control terminology. The n_{d} components of w_{d} are the disturbance or noise signals. The vector signal w_{etc} contains all remaining exogenous inputs—some of these signals will be discussed in chapter 10.

We partition the regulated variables:

$$z = \left[\begin{array}{c} z_{\mathrm{c}} \\ z_{\mathrm{a}} \\ z_{\mathrm{o}} \\ z_{\mathrm{etc}} \end{array}\right] \begin{array}{ll} \updownarrow n_{\mathrm{c}} & \text{commanded variables} \\ \updownarrow n_{\mathrm{a}} & \text{actuator signals} \\ \updownarrow n_{\mathrm{o}} & \text{other critical signals} \\ & \text{other components of } z \end{array}$$

The $n_{\rm c}$ components of $z_{\rm c}$ are the regulated variables that the commands $w_{\rm c}$ are intended to control or regulate—the "output" in classical control terminology. The $n_{\rm a}$ components of $z_{\rm a}$ are the actuator signals, which we remind the reader must be included in z in any sensible formulation of the controller design problem. The $n_{\rm o}$ components of $z_{\rm o}$ are other critical signals such as sensor signals or state variables. The vector signal $z_{\rm etc}$ contains all remaining regulated variables.

We conformally partition the closed-loop transfer matrix H:

$$\begin{bmatrix} z_{\rm c} \\ z_{\rm a} \\ z_{\rm o} \\ z_{\rm etc} \end{bmatrix} = \begin{bmatrix} H_{\rm cc} & H_{\rm cd} & \star \\ H_{\rm ac} & H_{\rm ad} & \star \\ H_{\rm oc} & H_{\rm od} & \star \\ \star & \star & \star \end{bmatrix} \begin{bmatrix} w_{\rm c} \\ w_{\rm d} \\ w_{\rm etc} \end{bmatrix}.$$

The symbol $\star$ is used to denote a submatrix of H that is not used to formulate performance specifications (some of these submatrices will be used in chapter 10). In the next few sections we consider specifications on each of the other submatrices of H. We remind the reader that a convex or affine specification or functional on a submatrix of H corresponds to a convex or affine specification or functional on the entire matrix H (see section 6.2.3).

8.1 Input/Output Specifications

In this section we consider specifications on $H_{\rm cc}$, the closed-loop transfer matrix from the command or set-point inputs to the commanded variables, *i.e.* the variables the commands are intended to control. In classical control terminology, $H_{\rm cc}$ consists of the closed-loop I/O transfer functions. Of course, it is possible for a control system to not have any command inputs or commanded variables ($n_{\rm c} = 0$), *e.g.* the classical regulator.

The submatrix $H_{\rm cc}$ determines the response of the commanded variables $z_{\rm c}$ to the command inputs $w_{\rm c}$ *only*; $z_{\rm c}$ will in general also be affected by the disturbances ($w_{\rm d}$) and the other exogenous input signals ($w_{\rm etc}$) (these effects are considered in the next section). Thus, the signal $H_{\rm cc}w_{\rm c}$ is the *noise-free* response of the commanded variables. In this section, we will assume for convenience that $w_{\rm d} = 0$, $w_{\rm etc} = 0$, so that $z_{\rm c} = H_{\rm cc}w_{\rm c}$. In other words, throughout this section $z_{\rm c}$ will denote the noise-free response of the commanded variables.

8.1.1 Step Response Specifications

Specifications are sometimes expressed in terms of the step response of $H_{\rm cc}$, especially when there is only one command signal and one commanded variable ($n_{\rm c} = 1$). The step response gives a good indication of the response of the controlled variable to command inputs that are constant for long periods of time and occasionally change quickly to a new value (sometimes called the *set-point*). We first consider the case $n_{\rm c} = 1$. Let $s(t)$ denote the step response of the transfer function $H_{\rm cc}$.

Asymptotic Tracking

A common specification on $H_{\rm cc}$ is

$$\lim_{s\to\infty} s(t) = H_{\rm cc}(0) = 1,$$

which means that for $w_{\rm c}$ constant (and as mentioned above, $w_{\rm d} = 0$, $w_{\rm etc} = 0$), $z_{\rm c}(t)$ converges to $w_{\rm c}$ as $t \to \infty$, or equivalently, the closed-loop transfer function from the command to the commanded variable is one at $s = 0$. We showed in section 6.4.1 that the specification

$$\mathcal{H}_{\rm asympt\_trk} = \{H \mid H_{\rm cc}(0) = 1\}$$

is *affine*, since the functional

$$\phi(H) = H_{\rm cc}(0)$$

is affine.

A strengthened version of asymptotic tracking is *asymptotic tracking of order* k: $H_{\rm cc}(0) = 1$, $H_{\rm cc}^{(j)}(0) = 0$, $1 \le j \le k$. This specification is commonly encountered for $k = 1$ and $k = 2$, and referred to as “asymptotic tracking of ramps” or “zero steady-state velocity error” (for $k = 1$) and “zero steady-state acceleration error” (for $k = 2$). These higher order asymptotic tracking specifications are also affine.

Overshoot and Undershoot

We define two functionals of $H_{\rm cc}$: the *overshoot*,

$$\phi_{\rm os}(H_{\rm cc}) \stackrel{\Delta}{=} \sup_{t\ge 0} s(t) - 1,$$

and the *undershoot*,

$$\phi_{\rm us}(H_{\rm cc}) \stackrel{\Delta}{=} \sup_{t\ge 0} -s(t).$$

Figure 8.1 shows a typical step response and the values of these functionals.

These functionals are convex, so the specifications

$$\mathcal{H}_{\rm os} \stackrel{\Delta}{=} \{H \mid \phi_{\rm os}(H_{\rm cc}) \le \alpha\},$$
$$\mathcal{H}_{\rm us} \stackrel{\Delta}{=} \{H \mid \phi_{\rm us}(H_{\rm cc}) \le \alpha\}$$

are convex: for example, if each of two step responses does not exceed 10% overshoot ($\alpha = 0.1$) then neither does their average (see section 6.4.2 and figure 6.6).

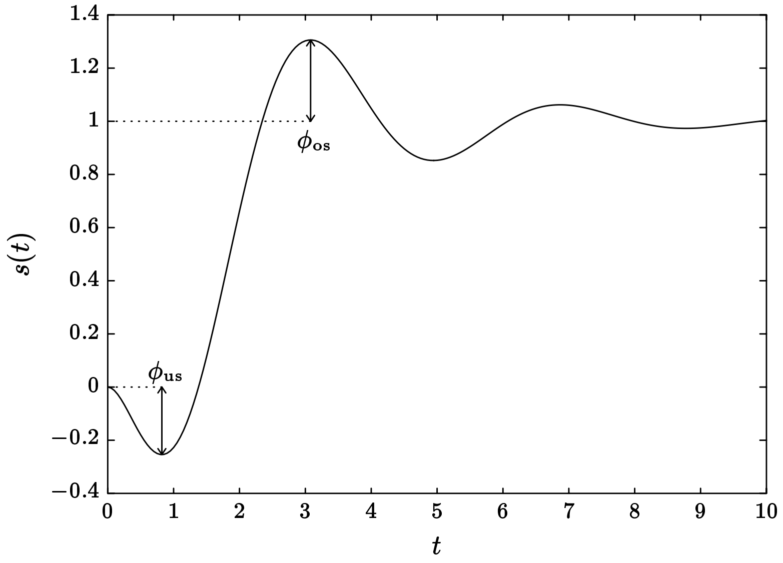


Figure 8.1 A typical step response s and its overshoot (ϕ_{os}) and undershoot (ϕ_{us}). The asymptotic tracking specification $\mathcal{H}_{\mathrm{asympt\_trk}}$ requires that the step response converge to one as $t \longrightarrow \infty$.

The functionals ϕ_{os} and ϕ_{us} are usually used together with the asymptotic tracking constraint $\mathcal{H}_{\mathrm{asympt\_trk}}$; otherwise we might use the *relative overshoot* and *relative undershoot* defined by

$$\phi_{\mathrm{ros}}(H_{\mathrm{cc}}) \triangleq \begin{cases} \sup_{t\geq 0} s(t)/H_{\mathrm{cc}}(0) - 1 & \text{if } H_{\mathrm{cc}}(0) > 0, \\ +\infty & \text{if } H_{\mathrm{cc}}(0) \leq 0, \end{cases}$$

$$\phi_{\mathrm{rus}}(H_{\mathrm{cc}}) \triangleq \begin{cases} \sup_{t\geq 0} -s(t)/H_{\mathrm{cc}}(0) & \text{if } H_{\mathrm{cc}}(0) > 0, \\ +\infty & \text{if } H_{\mathrm{cc}}(0) \leq 0. \end{cases}$$

It is less obvious that the specifications

$$\mathcal{H}_{\mathrm{ros}} \triangleq \{H \mid \phi_{\mathrm{ros}}(H_{\mathrm{cc}}) \leq \alpha\}, \tag{8.1}$$

$$\mathcal{H}_{\mathrm{rus}} \triangleq \{H \mid \phi_{\mathrm{rus}}(H_{\mathrm{cc}}) \leq \alpha\} \tag{8.2}$$

are convex. To see that the relative overshoot constraint $\mathcal{H}_{\mathrm{ros}}$ is convex, we rewrite it as

$$\mathcal{H}_{\mathrm{ros}} = \{H \mid H_{\mathrm{cc}}(0) > 0,\ s(t) - (1+\alpha)H_{\mathrm{cc}}(0) \leq 0 \text{ for all } t \geq 0\}.$$

If H, $\tilde{H} \in \mathcal{H}_{\mathrm{ros}}$ and $0 \leq \lambda \leq 1$, then $H_\lambda = \lambda H + (1-\lambda)\tilde{H}$ satisfies $H_{\lambda\mathrm{cc}}(0) > 0$, and for each $t \geq 0$ we have $s(t) - (1+\alpha)H_{\lambda\mathrm{cc}}(0) \leq 0$. Hence, $H_\lambda \in \mathcal{H}_{\mathrm{ros}}$.

Since the functional inequality specifications (8.1–8.2) are convex for each α, the relative overshoot and relative undershoot functionals are *quasiconvex*; they are not, however, convex. If one step response, $s(t)$, has a relative overshoot of 30%, and another step response $\tilde{s}(t)$ has a relative overshoot of 10%, then their average has a relative overshoot not exceeding 30%; but it may exceed 20%, the average of the two relative overshoots. An example of two such step responses is shown in figure 8.2.

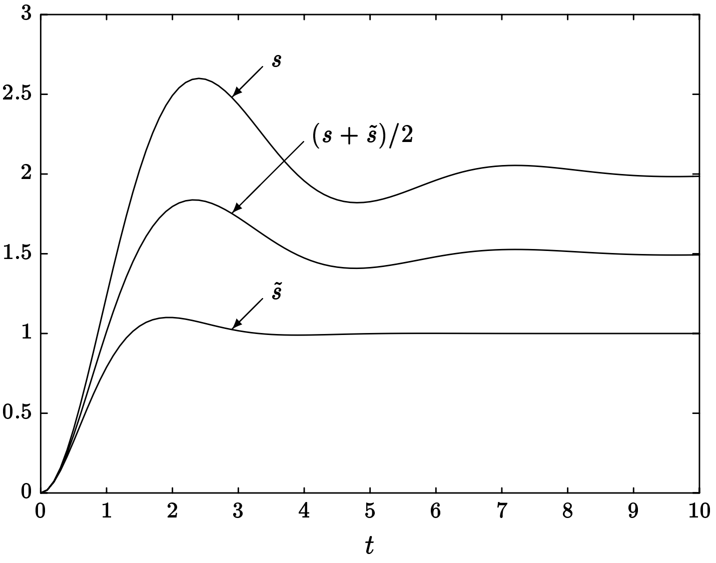


Figure 8.2 The relative overshoot of the step responses s and $\tilde{s}$ are 30% and 10% respectively. Their average, $(s+\tilde{s})/2$, has a relative overshoot of 23%. This example shows that relative overshoot is not a convex functional of H. It is, however, quasiconvex.

Rise Time and Settling Time

There are many definitions of *rise time* and *settling time* in use; we shall use

$$\phi_{\rm rise}(H_{\rm cc}) \stackrel{\Delta}{=} \inf\{T \mid s(t) > 0.8 \text{ for } t \geq T\},$$

$$\phi_{\rm settle}(H_{\rm cc}) \stackrel{\Delta}{=} \inf\{T \mid |s(t)-1| < 0.05 \text{ for } t \geq T\},$$

as illustrated in figure 8.3. The functional $\phi_{\rm rise}$ is usually used together with the asymptotic tracking specification $\mathcal{H}_{\rm asympt\_trk}$; we can also define relative or normalized rise time.

The functional inequality specifications

$$\mathcal{H}_{\rm rise} \stackrel{\Delta}{=} \{H \mid \phi_{\rm rise}(H_{\rm cc}) \leq T_{\rm max}\},$$

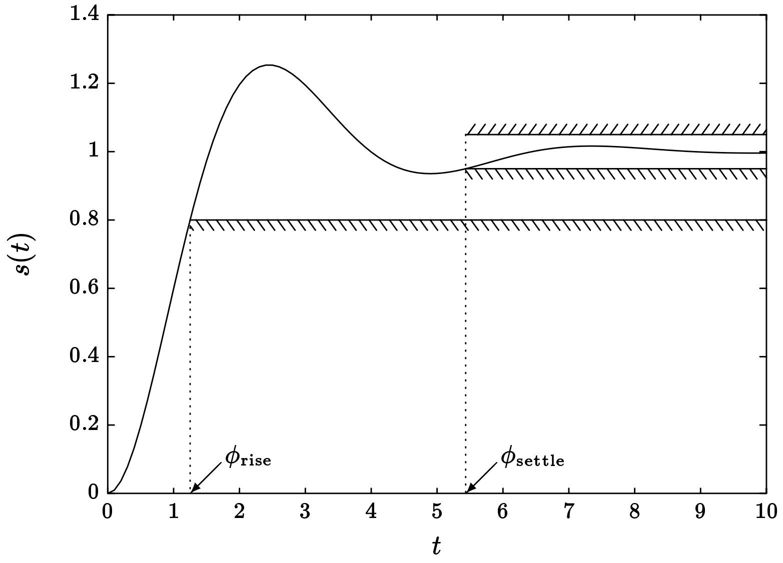


Figure 8.3 The value of the rise-time functional, $\phi_{\rm rise}$, is the earliest time after which the step response always exceeds 0.8. The value of the settling-time functional, $\phi_{\rm settle}$, is the earliest time after which the step response is always within 5% of 1.0.

$$\mathcal{H}_{\rm settle} \overset{\Delta}{=} \{H \mid \phi_{\rm settle}(H_{\rm cc}) \le T_{\rm max}\}$$

are convex: if two step responses each settle to with 5% within some time limit $T_{\rm max}$, then so does their average. Thus, the rise-time and settling-time functionals $\phi_{\rm rise}$ and $\phi_{\rm settle}$ are quasiconvex, *i.e.*,

$$\phi_{\rm rise}(\lambda H_{\rm cc} + (1-\lambda)\tilde{H}_{cc}) \le \max\{\phi_{\rm rise}(H_{\rm cc}),\ \phi_{\rm rise}(\tilde{H}_{cc})\}$$

for all $0 \le \lambda \le 1$, but we do not generally have

$$\phi_{\rm rise}(\lambda H_{\rm cc} + (1-\lambda)\tilde{H}_{cc}) \le \lambda\phi_{\rm rise}(H_{\rm cc}) + (1-\lambda)\phi_{\rm rise}(\tilde{H}_{cc}),$$

so they are not convex. Figure 8.4 demonstrates two step responses for which this inequality, with $\lambda = 0.5$, is violated.

We mention that there are other definitions of rise-time functionals that are not quasiconvex. One example of such a definition is the time for the step response to rise from the signal level 0.1 (10%) to 0.9 (90%):

$$\phi_{10-90}(H_{\rm cc}) \overset{\Delta}{=} \inf\{t \mid s(t) \ge 0.9\} - \inf\{t \mid s(t) \ge 0.1\}.$$

While this may be a useful functional of $H_{\rm cc}$ in some contexts, we doubt the utility of the (nonconvex) specification $\phi_{10-90}(H_{\rm cc}) \le T_{\rm max}$, which can be satisfied by a step response with a long initial delay or a step response with very large high frequency oscillations.

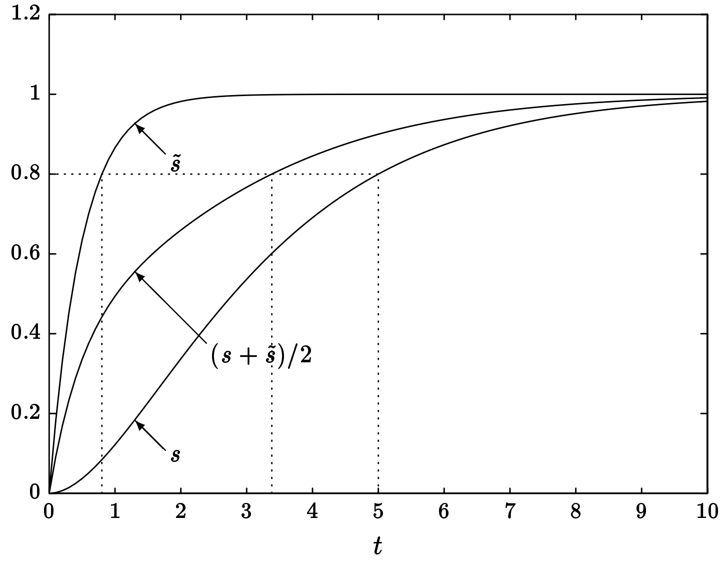


Figure 8.4 The rise times of the step responses s and $\tilde{s}$ are 5.0 and 0.8 seconds respectively. Their average, $(s+\tilde{s})/2$, has a rise time of $3.38 > 5.8/2$ seconds. This example shows that rise time is not a convex functional of H, although it is quasiconvex.

General Step Response Envelope Specifications

Many of the step response specifications considered so far are special cases of general *envelope* constraints on the step response:

$$\mathcal{H}_{\mathrm{env}} = \{H \mid s_{\min}(t) \leq s(t) \leq s_{\max}(t) \quad \text{for all } t \geq 0\}, \tag{8.3}$$

where $s_{\min}(t) \leq s_{\max}(t)$ for all $t \geq 0$. An example of an envelope constraint is shown in figure 8.5. The envelope constraint $\mathcal{H}_{\mathrm{env}}$ is convex, since if two step responses lie in the allowable region, then so does their average.

We mention one useful way that a general envelope constraint $\mathcal{H}_{\mathrm{env}}$ can be expressed as a functional inequality specification with a convex functional. We define the *maximum envelope violation*,

$$\phi_{\mathrm{max\_env\_viol}}(H_{\mathrm{cc}}) \stackrel{\Delta}{=} \sup_{t\geq 0} \max\{s(t) - s_{\max}(t),\ s_{\min}(t) - s(t),\ 0\}.$$

The envelope specification $\mathcal{H}_{\mathrm{env}}$ can be expressed as

$$\mathcal{H}_{\mathrm{env}} = \{H \mid \phi_{\mathrm{max\_env\_viol}}(H_{\mathrm{cc}}) \leq 0\}.$$

General Response-Time Functional

The quasiconvex functionals ϕ_{rise} and ϕ_{settle} are special cases of a simple, general paradigm for measuring the *response time* of a unit step response. Suppose that we

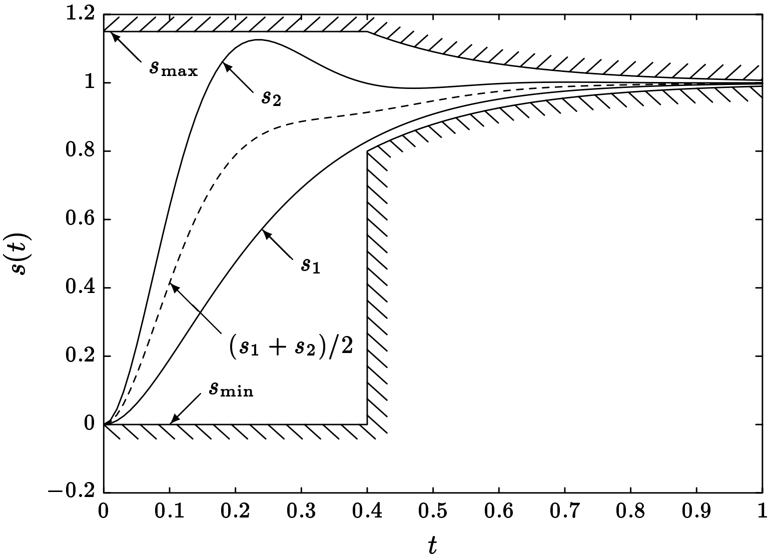


Figure 8.5 General envelope specification on a step response. The two step responses s_1 and s_2 satisfy this constraint; their average (shown in dashed line) also satisfies the envelope constraint.

have upper and lower bounds for a general envelope constraint, $s_{\max}(t)$ and $s_{\min}(t)$. Suppose that $s_{\max}(t)$ does not increase, and $s_{\min}(t)$ does not decrease, for increasing t. For each $T > 0$, we consider the *time-scaled* envelope specification

$$s_{\min}(t/T) \leq s(t) \leq s_{\max}(t/T) \quad \text{for all} \;\; t \geq 0. \tag{8.4}$$

(8.4) defines a nested family of convex specifications, parametrized by T. For $T = 1$ we have the original envelope specification with bounds $s_{\min}$ and $s_{\max}$; for $T > 1$ we have a weaker specification, and for $T < 1$ we have a stronger specification; roughly speaking, T is the normalized response time. We define the generalized response-time functional as

$$\phi_{\mathrm{grt}}(H_{\mathrm{cc}}) \overset{\Delta}{=} \inf \{T \mid s_{\min}(t/T) \leq s(t) \leq s_{\max}(t/T) \quad \text{for all} \;\; t \geq 0\}.$$

This construction is shown in figure 8.6. The comments at the end of section 6.2.2 show that ϕ_{grt} is quasiconvex.

Step Response Interaction

We now consider the case where there are multiple commands (and multiple commanded variables) so that H_{cc} is an $n_{\mathrm{c}} \times n_{\mathrm{c}}$ transfer matrix, where $n_{\mathrm{c}} > 1$. Its diagonal entries are the transfer functions from the command inputs to their associated commanded variables, which may be required to meet the various specifications

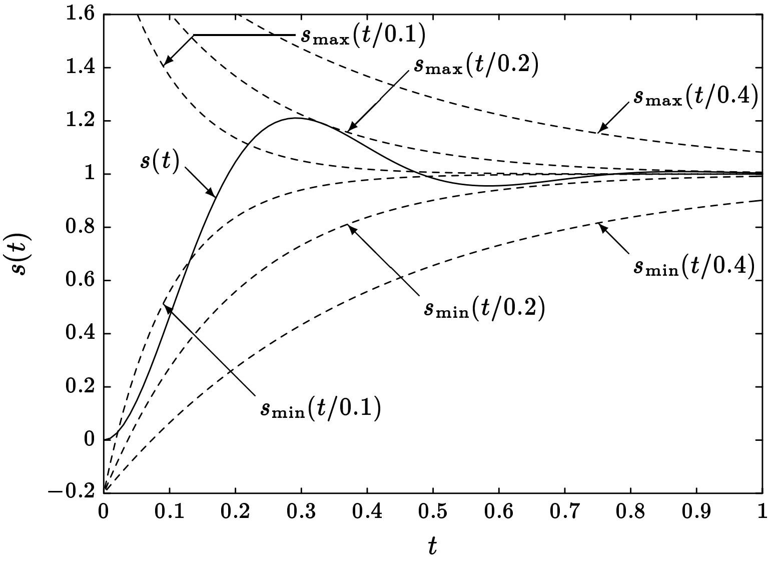


Figure 8.6 A step response is shown together with the envelopes $s_{\min}(t/T)$ and $s_{\max}(t/T)$ for three values of T, where $s_{\min}(t) = 1 - 1.2 \exp -t$ and $s_{\max}(t) = 1 + \exp -t$. For $T = 0.2$ the step response just lies inside the envelopes, so the value of ϕ_{grt} is 0.2.

discussed above, *e.g.*, limits on overshoot or rise time. The *off-diagonal* entries of H_{cc} are the transfer functions from the commands to *other* commanded variables, and are called the *command interaction* transfer functions. It is generally desirable that these transfer functions be small, so that each command input does not excessively disturb the other commanded variables.

Let $s(t)$ denote the step response matrix of H_{cc}. One mild constraint on command interaction is *asymptotic decoupling*:

$$\mathcal{H}_{\text{asympt\_dcpl}} = \left\{ H \;\middle|\; H_{\text{cc}}(0) = \lim_{t \to \infty} s(t) \text{ is diagonal} \right\}. \tag{8.5}$$

This specification ensures that if the commands are constant, then the effect on each commanded variable due to the other commands converges to zero: there is no steady-state interaction for constant commands.

A stronger specification that limits command interaction is an envelope constraint on each entry of $s(t)$,

$$\mathcal{H}_{\text{mimo\_env}} = \{ H \mid s_{\min}(t) \le s(t) \le s_{\max}(t) \quad \text{for all } t \ge 0 \}, \tag{8.6}$$

where $s_{\min}(t)$ and $s_{\max}(t)$ are matrices, and the inequalities in (8.6) are component by component. The envelope specification $\mathcal{H}_{\text{mimo\_env}}$ is convex.

An example of $\mathcal{H}_{\text{mimo\_env}}$ is shown in figure 8.7, along with a step matrix $s(t)$ that meets it. Of course, the responses shown in figure 8.7 are for steps applied to

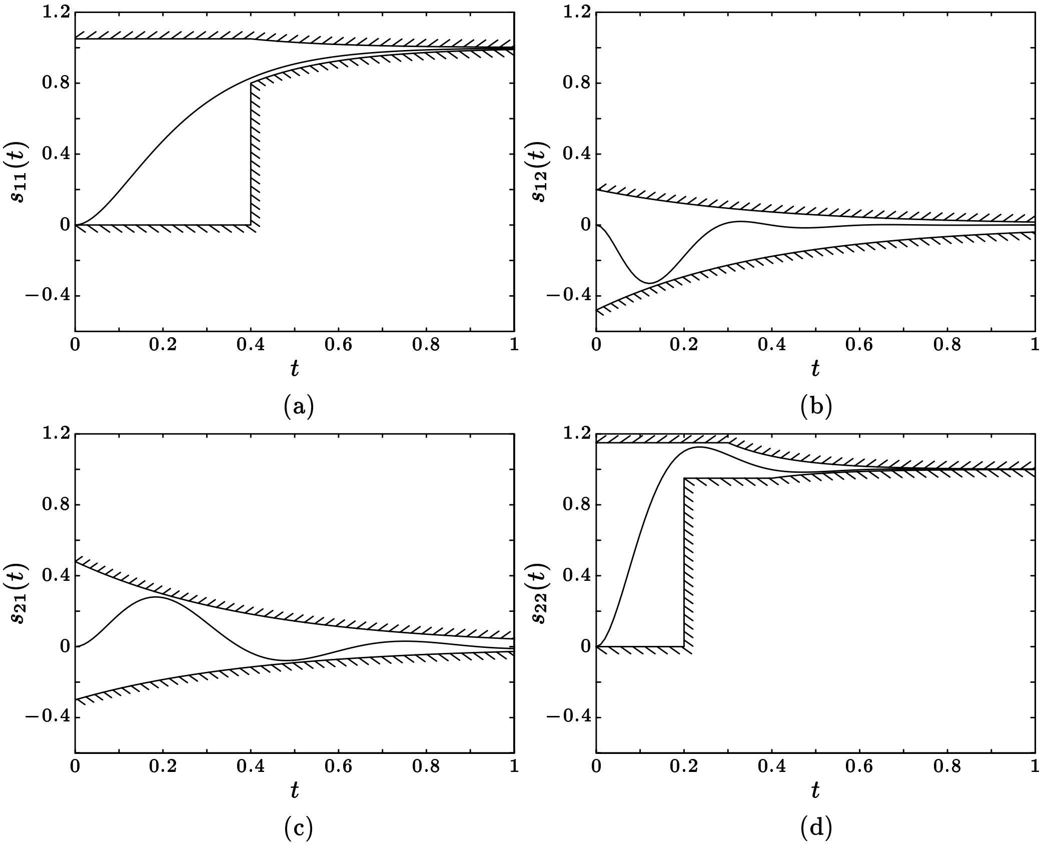


Figure 8.7 Design specifications requiring the decoupling of responses to step commands.

the inputs *one at a time*. Figure 8.8 shows the response of the commanded variables to a particular command signal that has a set-point change in w_1 at $t = 0.5$ and then a set-point change in w_2 at $t = 2.0$. The perturbation in z_2 right after $t = 0.5$ and the perturbation in z_1 right after $t = 2.0$ are due to command interaction. The specification $\mathcal{H}_{\rm mimo\_env}$ limits this perturbation, and guarantees that after the set-point change for z_2, for example, the effect on z_1 will fade away.

An extreme form of $\mathcal{H}_{\rm mimo\_env}$ is to require that the off-diagonal step responses be *zero*, or equivalently, that $H_{\rm cc}$ be diagonal. This is called *exact* or *complete decoupling*:

$$\mathcal{H}_{\rm dcpl} = \{H \mid H_{\rm cc} \text{ is diagonal }\}.$$

This specification forbids any command interaction at all, regardless of the command signals. $\mathcal{H}_{\rm dcpl}$ is affine.

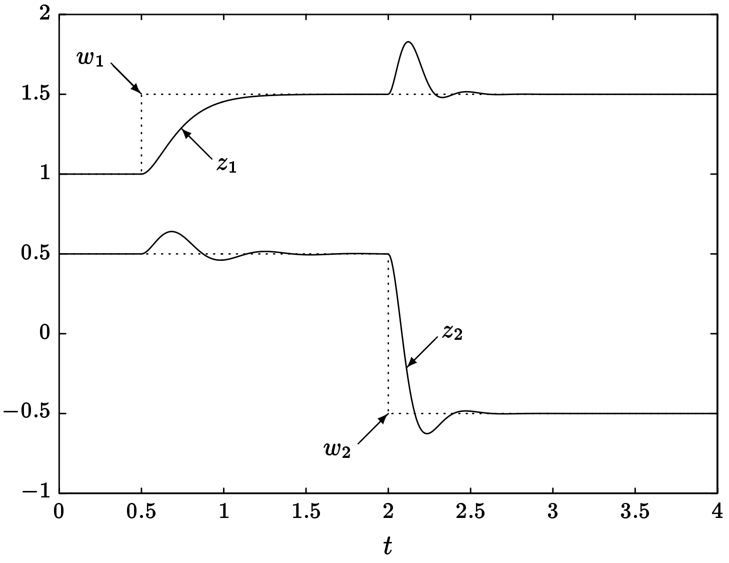


Figure 8.8 An example of command interaction in a two-input, two-output system. The individual step responses of the system are shown in figure 8.7. Output z_1 tracks a step change in the command signal w_1. However, z_1 is perturbed by a step change in the command signal w_2. Similarly, z_2 is perturbed by a step change in w_1.

Miscellaneous Step Response Specifications

Other specifications often expressed in terms of the step response of H_{cc} include *monotonic step response* and I/O *slew-rate limits.* The monotonic step response constraint is

$$\begin{aligned}\mathcal{H}_{\mathrm{mono\_sr}} &= \{H \mid s(t) \text{ is nondecreasing for all } t \geq 0\}\\ &= \{H \mid h(t) \geq 0 \quad \text{for all } t \geq 0\}\end{aligned}$$

where $h(t)$ is the impulse response of H_{cc}. Thus, $\mathcal{H}_{\mathrm{mono\_sr}}$ requires that the commanded variable move only in the direction of the new set-point command in response to a single abrupt change in set-point. $\mathcal{H}_{\mathrm{mono\_sr}}$ is a stricter specification than requiring that both the undershoot and overshoot be zero. $\mathcal{H}_{\mathrm{mono\_sr}}$ is convex.

A related specification is a slew-rate limit on the step response:

$$\mathcal{H}_{\mathrm{slew\_sr}} = \left\{ H \;\middle|\; \left|\frac{d}{dt}s(t)\right| = |h(t)| \leq M_{\mathrm{slew}} \;\text{ for all } t \geq 0 \right\}.$$

Response to Other Inputs

We close this section by noting that the response of z_{c} to *any* particular command signal, and not just a unit step, can be substituted for the step response in all of the specifications above. This *specific input tracking requirement* is convex.

For example, we may require that in response to the particular command signal $w_c = w_{pgm}$, shown in figure 8.9, the commanded variable z_c lies in the envelope shown. w_{pgm} might represent an often repeated temperature cycle in an industrial oven (the mnemonic abbreviates "program"). The specification

$$\mathcal{H}_{pgm\_trk} = \{ H \mid \|H_{cc} w_{pgm} - w_{pgm}\|_\infty \leq 30 \},$$

shown in figure 8.9, requires that the actual temperature, z_c, always be within 30°C of the commanded temperature, w_c.

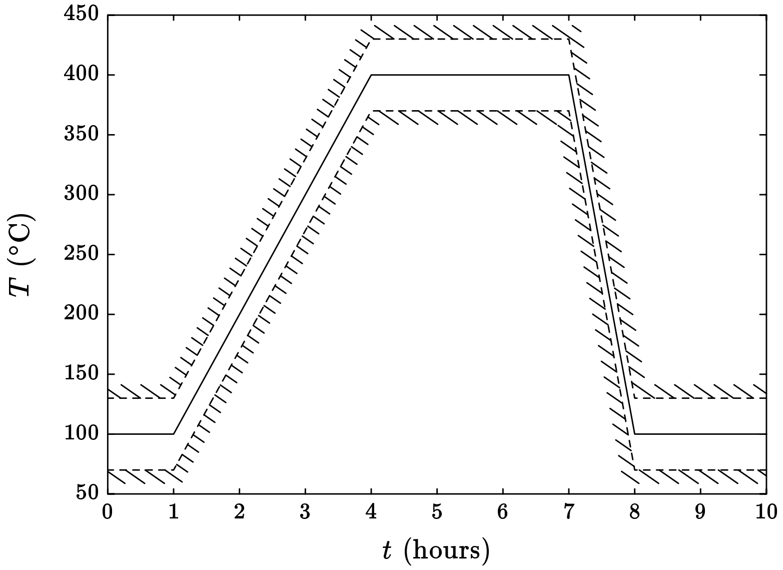


Figure 8.9 An example of a temperature command signal that might be used in a plastics process. Powder is slowly melted, and then sintered for 3 hours. It is then rapidly cooled through the melt point. The envelope constraints on the actual temperature require the temperature error to be less than 30°C.

8.1.2 Tracking Error Formulation

Step response specifications constrain the response of the system to specific commands: a step input at each command. By linearity and time-invariance, this constrains the response to commands that are constant for long periods and change abruptly to new values, which is sometimes a suitable model for the commands that will be encountered in practice. In many cases, however, the typical command signals are more diverse—they may change frequently in a way that is not completely predictable. This is often the case in a command-following system, where the goal is to have some system variables follow or track a continuously changing command signal.

In the tracking error formulation of I/O specifications, we define the *tracking error* as $e_{\mathrm{trk}} \overset{\Delta}{=} z_{\mathrm{c}} - w_{\mathrm{c}}$: the difference between the actual response of the commanded variables and the commands, as shown in figure 8.10.

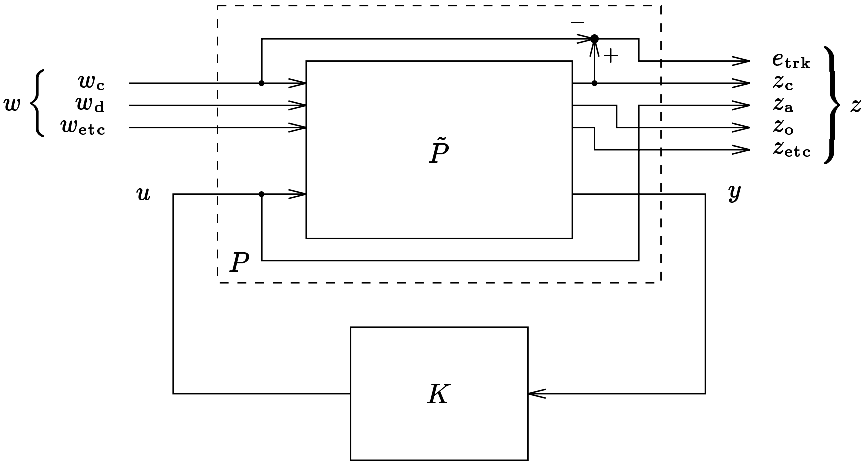


Figure 8.10 An architecture for expressing I/O specifications in terms of the tracking error $e_{\mathrm{trk}} = z_{\mathrm{c}} - w_{\mathrm{c}}$.

We will assume that e_{trk} is available as a part of z, and H_{trk} will denote the submatrix of H that is the closed-loop transfer matrix from the commands w_{c} to the tracking errors e_{trk}. A general tracking error specification has the form

$$\mathcal{H}_{\mathrm{trk}} \overset{\Delta}{=} \{H \mid \|H_{\mathrm{trk}}\|_{\mathrm{trk\_err}} \leq \alpha \}, \tag{8.7}$$

i.e., the closed-loop transfer matrix from commands to tracking error should be small, as measured with the norm $\|\cdot\|_{\mathrm{trk\_err}}$. Using any of the norms from chapter 5 allows a wide variety of I/O specifications to be formed from the general tracking error specification $\mathcal{H}_{\mathrm{trk}}$, all of which are convex. We will briefly list some of these specifications and their interpretations.

RMS Mistracking Limit

One simplified model of the command signal is that it is a stochastic process with a known power spectrum S_{cmd}. Of course this model is quite crude, and only intended to capture a few key features of the command signal, such as size and bandwidth: the command signal may in fact be generated by a human operator. If we accept this model, and take the RMS value as our measure of the size of the tracking error, the general tracking specification (8.7) can be expressed as the weighted $\mathbf{H}_2$ norm-bound

$$\mathcal{H}_{\rm rms\_trk} \stackrel{\Delta}{=} \{H \mid \|H_{\rm trk}W\|_2 \le \alpha \}, \tag{8.8}$$

where W is a spectral factor of $S_{\rm cmd}$: $S_{\rm cmd}(\omega) = W(j\omega)^* W(j\omega)$.

Worst Case RMS Mistracking Limit

We may reduce our a priori assumptions about the command signal even further, by assuming that we do not know the spectrum, but know only a maximum $W_{\rm cmd}$-weighted RMS value for the command signal $w_{\rm c}$, where $W_{\rm cmd}$ is some appropriate weight. If our measure of the size of the tracking error $e_{\rm trk}$ is the *worst case* $W_{\rm trk}$-weighted RMS value it might have, where $W_{\rm trk}$ is some appropriate weight, then the appropriate norm in the general tracking error specification (8.7) is the weighted $\mathbf{H}_\infty$ norm:

$$\mathcal{H}_{\rm hinf\_trk} \stackrel{\Delta}{=} \{H \mid \|W_{\rm trk}H_{\rm trk}W_{\rm cmd}\|_\infty \le \alpha \}. \tag{8.9}$$

For $n_{\rm c} = 1$ (meaning the weights are scalar, and the maximum singular value of the transfer function is simply its magnitude), this specification can also be cast in the more classical form:

$$\mathcal{H}_{\rm hinf\_trk} = \{H \mid |H_{\rm trk}(j\omega)| \le l_{\rm trk}(\omega),\ H_{\rm trk} \text{ is stable } \} \tag{8.10}$$

where

$$l_{\rm trk}(\omega) = \frac{\alpha}{|W_{\rm cmd}(j\omega)W_{\rm trk}(j\omega)|}.$$

The classical interpretation is that $l_{\rm trk}(\omega)$ is a frequency-dependent limit on the tracking error transfer function, and the specification (8.10) ensures that the "command to tracking error transfer function is small at those frequencies where the command has significant energy". An example is shown in figure 8.11.

Worst Case Peak Mistracking Limit

Another specific form that the general tracking error specification (8.7) can take is a worst case peak mistracking limit:

$$\mathcal{H}_{\rm pk\_trk} \stackrel{\Delta}{=} \{H \mid \|H_{\rm trk}\|_{\rm pk\_gn} \le \alpha\}. \tag{8.11}$$

This specification arises as follows. We use an unknown-but-bounded model of the command signals: we assume only

$$\|w_{\rm c}\|_\infty \le M. \tag{8.12}$$

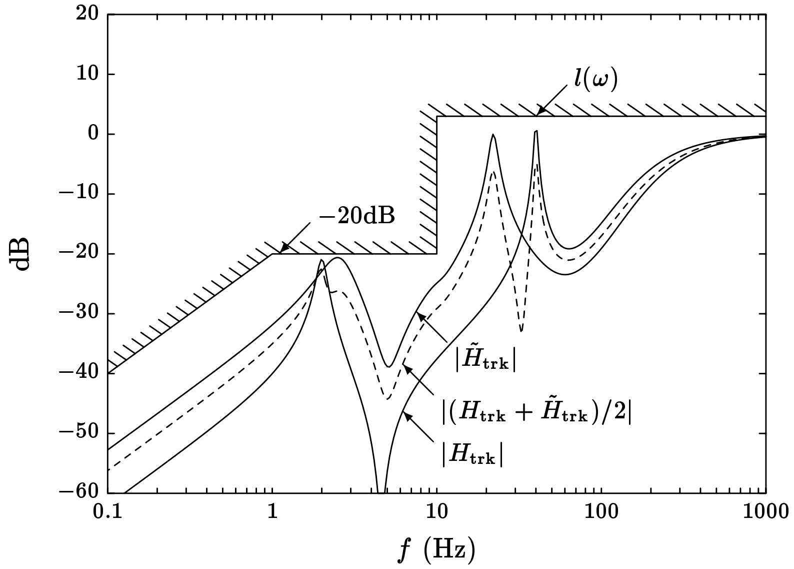


Figure 8.11 Upper bounds on frequency response magnitudes are convex. The bound $l(\omega)$ on the tracking error transfer function ensures that the tracking error transfer function is below -20dB at frequencies below 10Hz, and rolls off below 1Hz. Two transfer functions $H_{\rm trk}$ and $\tilde{H}_{\rm trk}$ that satisfy the specification (8.10) are shown, together with their average. Of course, the magnitude of $(H_{\rm trk} + \tilde{H}_{\rm trk})/2$ is not the average of the magnitudes of $H_{\rm trk}$ and $\tilde{H}_{\rm trk}$, although it is no larger than the average.

Our measure of the tracking error is the *worst case peak* (over all command signals consistent with (8.12)) of the tracking error $e_{\rm trk}$:

$$\|e_{\rm trk}\|_\infty \le M_{\rm trk} \text{ whenever } w_{\rm c} \text{ satisfies (8.12)} .$$

This constraint is precisely (8.11), with $\alpha = M_{\rm trk}/M$.

Since most plants are strictly proper, $\|H_{\rm trk}\|_{\rm pk\_gn}$ will usually be at least one. This can be seen from the block diagram in figure 8.10: a step change in the command input $w_{\rm c}$ will produce an immediate, equal sized change in the tracking error. After some time, the closed-loop system will drive the tracking error to a smaller value. For this reason, the specification (8.11) may not be useful.

A useful variation on this worst case peak tracking error limit is to assume more about the command signals, for example, to assume a maximum slew rate as well as a maximum peak for the command signal. In this case the appropriate norm in the general tracking error limit (8.7) would be the worst case norm $\|\cdot\|_{\rm wc}$, from section 5.2.4.

For example, consider the temperature response envelope shown in figure 8.9. Provided the system achieves asymptotic tracking of constant commands, so that

$H_{\rm trk}(0) = 0$, we can ignore the offset in the command signal. Since $\|w_{\rm c} - 250\|_\infty \le 150^\circ{\rm C}$ and $\|\dot{w}_c\|_\infty \le 300^\circ{\rm C/Hr}$, we can specify

$$\mathcal{H}_{\rm pk\_trk\_slew} \stackrel{\Delta}{=} \{H \mid \|H_{\rm trk}\|_{\rm wc} \le 30^\circ{\rm C}\}, \qquad (8.13)$$

where we take $M_{\rm ampl} = 150^\circ$C and $M_{\rm slew} = 300^\circ$C/Hr in the definition of the norm $\|\cdot\|_{\rm wc}$ (see section 5.2.4). The specification (8.13) is *tighter* than the envelope specification in figure 8.9: the specification (8.13) requires a peak tracking error of no more than 30°C for *any* command input that is between 100°C and 400°C, and slew limited by 300°C/Hr, while the specification in figure 8.9 requires the same peak tracking error for a *particular input* that is between 100°C and 400°C, and slew limited by 300°C/Hr.

8.1.3 Model Reference Formulation

An extension of the tracking error formulation consists of specifying a desired closed-loop I/O transfer matrix $H_{\rm ref\_des}$, called the *reference* or *model* transfer matrix, and the goal is to ensure that $H_{\rm cc} \approx H_{\rm ref\_des}$. Instead of forming the tracking error as $e_{\rm trk} = z_{\rm c} - w_{\rm c}$, we form the *model reference error* $e_{\rm mre} = z_{\rm c} - H_{\rm ref\_des}w_{\rm c}$: the difference between the actual response $z_{\rm c}$ and the desired response, $H_{\rm ref\_des}w_{\rm c}$. This is shown in figure 8.12. Note that the tracking error is just the model reference error when the model transfer matrix is the identity.

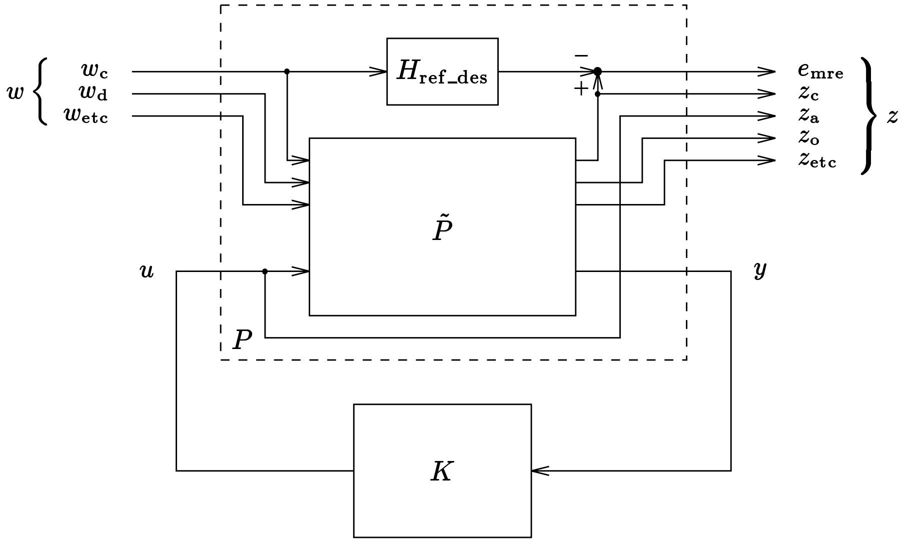


Figure 8.12 An architecture for expressing I/O specifications in terms of the error from a desired transfer matrix $H_{\rm ref\_des}$.

We will assume that the model reference error $e_{\rm mre}$ is contained in z. Let $H_{\rm mre}$ denote the submatrix of H that is the closed-loop transfer matrix from the command signal $w_{\rm c}$ to the model reference error $e_{\rm mre}$. In the model reference formulation of I/O specifications, we constrain $H_{\rm mre}$ to be small in some appropriate sense:

$$\mathcal{H}_{\rm mre} \stackrel{\Delta}{=} \{H \mid \|H_{\rm mre}\|_{\rm mre} \le \alpha \ \}. \tag{8.14}$$

The general model reference error specification (8.14) can take a wide variety of forms, depending on the norm used; we refer the reader to section 8.1.2 for a partial list, and chapter 5 for a general discussion.

8.2 Regulation Specifications

In this section we consider the effect on $z_{\rm c}$ of $w_{\rm d}$ only, just as in the previous sections we considered the effect on $z_{\rm c}$ of the command inputs only. The response of commanded variables to disturbances is determined by the closed-loop submatrix $H_{\rm cd}$; regulation specifications require that $H_{\rm cd}$ be "small". It is not surprising, then, that regulation specifications can usually be expressed in the form of norm-bound inequalities, *i.e.*,

$$\|H_{\rm cd}\|_{\rm reg} \le \alpha, \tag{8.15}$$

where $\|\cdot\|_{\rm reg}$ is some appropriate norm that depends, for example, on the model of the disturbances, how we measure the size of the undesired deviation of the commanded variables, and whether we limit the average or worst case deviation.

In the following sections we describe a few specific forms the general regulation specification (8.15) can take. Because these specifications have a form similar to I/O specifications such as limits on tracking error or model reference error, we will give a briefer description. For convenience we shall assume that $w_{\rm d}$ is a scalar disturbance and $w_{\rm c}$ is a single commanded variable, since the extension to vector disturbance signals and regulated variables is straightforward.

8.2.1 Rejection of Specific Disturbances

The simplest model for a disturbance is that it is constant, with some unknown value. The specification that this constant disturbance be asymptotically rejected at $z_{\rm c}$ is simply

$$\mathcal{H}_{\rm asympt\_rej} = \{H \mid H_{\rm cd}(0) = 0\}.$$

This specification has the same form as the asymptotic decoupling specification (8.5) (and is therefore closed-loop affine), but it has a very different meaning. The specification $\mathcal{H}_{\rm asympt\_rej}$ can be tightened by limiting the step response of $H_{\rm cd}$ to lie in a given envelope, as in the command response specifications discussed in section 8.1. For example, we may require that the effect of a unit step input at $w_{\rm d}$ on $z_{\rm c}$ should decay to no more than 0.05 within some given time $T_{\rm rej}$. Such a specification ensures that the closed-loop system will counteract the effects of a rapidly applied (or changed) constant disturbance on the commanded variable.

In most cases, however, disturbances cannot be so easily described. In the next few sections we discuss specifications that limit the effect of disturbances about which less is known.

8.2.2 RMS Regulation

A common model for a disturbance is a stochastic process with a known power spectral density $S_{\rm dist}$. The specification

$$\mathcal{H}_{\rm rms\_reg} \stackrel{\Delta}{=} \{H \mid \|H_{\rm cd}W\|_2 \le \alpha\}, \tag{8.16}$$

where W is a spectral factor of $S_{\rm dist}$, limits the RMS deviation of the commanded variable (due to the disturbance) to be less then α. This specification has exactly the same form as the RMS mistracking limit (8.8): a weighted $\mathbf{H}_2$ norm-bound.

The power spectral density of the disturbance is rarely known precisely; $S_{\rm dist}$ is usually meant to capture only a few key features of the disturbance, perhaps its RMS value and bandwidth. The power spectral density

$$S_{\rm dist}(\omega) = \frac{2a^2\omega_{\rm bw}}{\omega^2+\omega_{\rm bw}^2},$$

for example, might be used to model a disturbance with an RMS value a and a bandwidth $\omega_{\rm bw}$.

8.2.3 Classical Frequency Domain Regulation

We may not be willing to model the disturbance with a specific power spectral density. Instead, we may model $w_{\rm d}$ as having an unknown power spectral density, but some given maximum RMS value. A limit on the worst case RMS response of $z_{\rm c}$ can be expressed as the $\mathbf{H}_\infty$ norm-bound

$$\mathcal{H}_{\rm hinf\_reg} \stackrel{\Delta}{=} \{H \mid \|H_{\rm cd}\|_\infty \le \alpha\ \},$$

which limits the RMS gain of the closed-loop transfer function $H_{\rm cd}$. Often, this specification is modified by frequency domain weights, reflecting the fact that either a maximum possible weighted-RMS value for the disturbance is assumed, or a limit on some weighted-RMS value of the commanded variable must be maintained. Such a frequency-weighted $\mathbf{H}_\infty$ norm-bound can be cast in the more classical form:

$$\mathcal{H}_{\rm hinf\_reg} = \{H \mid |H_{\rm cd}(j\omega)| \le l_{\rm reg}(\omega),\ H_{\rm cd} \text{ is stable }\}. \tag{8.17}$$

The classical interpretation is that $l_{\rm reg}(\omega)$ is a frequency-dependent limit on the disturbance to commanded variable transfer function, and the specification (8.17) ensures that the "disturbance to commanded variable transfer function is small at those frequencies where the disturbance has significant energy".

8.2.4 Regulation Bandwidth

The classical frequency domain regulation specification (8.17) is often expressed as *minimum regulation bandwidth* for the closed-loop system. One typical definition of the regulation bandwidth of the closed-loop system is

$$\phi_{\rm bw}(H_{\rm cd}) \stackrel{\Delta}{=} \sup\left\{\Omega \;\mid\; |H_{\rm cd}(j\omega)| \leq 0.1 \text{ for all } \omega \leq \Omega\right\},$$

which is the largest frequency below which we can guarantee that the disturbance to commanded variable transfer function is no more than -20dB, as shown in figure 8.13.

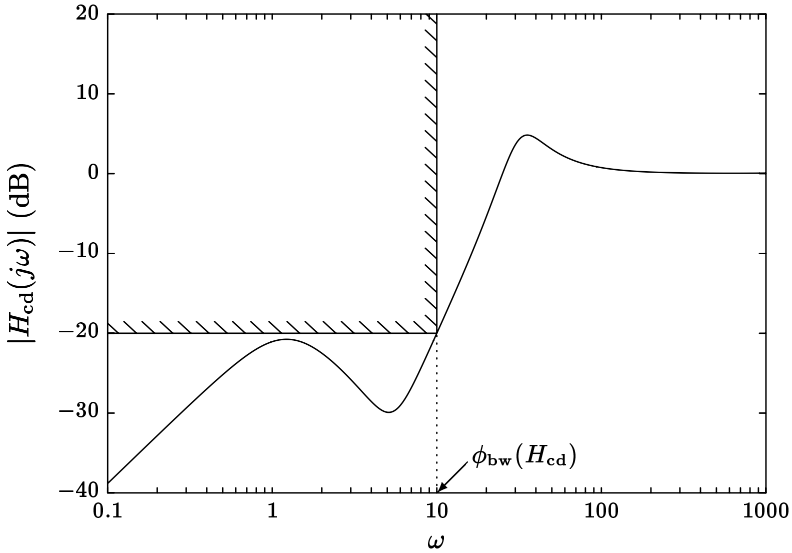


Figure 8.13 The value of the regulation bandwidth functional, $\phi_{\rm bw}$, is the largest frequency below which the disturbance to commanded variable transfer function, $H_{\rm cd}$, is no more than -20dB.

The minimum bandwidth specification

$$\mathcal{H}_{\rm min\_bw} \stackrel{\Delta}{=} \{H \;\mid\; \phi_{\rm bw}(H_{\rm cd}) \geq \Omega_{\rm min}\}$$

is convex, since it is a frequency-dependent bound on the magnitude of H, so the bandwidth functional $\phi_{\rm bw}$ is *quasiconcave*, meaning that $-\phi_{\rm bw}$ is quasiconvex. Alternatively, we note that the inverse of the regulation bandwidth, *i.e.*, $1/\phi_{\rm bw}$, is quasiconvex. The inverse bandwidth $1/\phi_{\rm bw}$ can be interpreted as a regulation response time.

A generalized definition of bandwidth, analogous to the generalized response time, is given by

$$\phi_{\rm gbw}(H_{\rm cd}) \stackrel{\Delta}{=} \sup\left\{\Omega \;\mid\; |H_{\rm cd}(j\omega)| \leq M(\omega/\Omega) \text{ for all } \omega\right\},$$

where $M(\omega)$ is a non-decreasing frequency-dependent magnitude bound.

8.2.5 Worst Case Peak Regulation

If we model the disturbance as unknown-but-bounded, say, $\|w_\mathrm{d}\|_\infty \leq M_\mathrm{d}$, and require that the worst case peak deviation of the commanded variable due to w_d is less than $M_\mathrm{max\_pk\_reg}$, *i.e.*,

$$\|H_\mathrm{cd} w_\mathrm{d}\|_\infty \leq M_\mathrm{max\_pk\_reg} \quad \text{whenever} \quad \|w_\mathrm{d}\|_\infty \leq M_\mathrm{d},$$

then we can specify the peak-gain bound

$$\mathcal{H}_\mathrm{pk\_dis} \stackrel{\Delta}{=} \{H \mid \|H_\mathrm{cd}\|_\mathrm{pk\_gn} \leq M_\mathrm{max\_pk\_reg}/M_\mathrm{d}\}.$$

8.3 Actuator Effort

In any control system the size of the actuator signals must be limited, *i.e.*,

$$\|u\|_\mathrm{act} \leq M_\mathrm{act}$$

for some appropriate norm $\|\cdot\|_\mathrm{act}$ and limit M_act. Reasons include:

- *Actuator heating.* Large actuator signals may cause excessive heating, which will damage or cause wear to the system. Such constraints can often be expressed in terms of an RMS norm of u, possibly with weights (see section 4.2.2).

- *Saturation or overload.* Exceeding absolute limits on actuator signals may damage an actuator, or cause the plant P to be a poor model of the system to be controlled. These specifications can be expressed in terms of a scaled or weighted peak norm of u.

- *Power, fuel, or resource use.* Large actuator signals may be associated with excessive power consumption or fuel or resource use. These specifications are often expressed in terms of a scaled or weighted average-absolute norm of u.

- *Mechanical or other wear.* Excessively rapid changes in the actuator signal may cause undesirable stresses or excessive wear. These constraints may be expressed in terms of slew rate, acceleration, or jerk norms of u (see section 4.2.8).

These limits on the size of u can be enforced by limiting in an appropriate way the size of H_ac and H_ad, the closed-loop transfer matrices from the command and disturbance signals to the actuator. For example, if the command signal is modeled as a stochastic process with a given power spectral density, then a weighted $\mathbf{H}_2$ norm-bound on H_ac will guarantee a maximum RMS actuator effort due to the command signal. If the command signal is modeled as unknown-but-bounded, and the peak of the actuator signal must be limited, then the actuator effort specification is a limit on the peak gain of H_{ac}. These specifications are analogous to many we have already encountered in this chapter.

We mention one simple but important distinction between a limit on the size of u and the associated limit on the size of H_{ac} (or H_{ad}). We will assume for simplicity that the command and actuator signals are scalar, and the disturbance is negligible. Suppose that we have the constraint

$$\|u\|_\infty \le 1 \tag{8.18}$$

on our actuator signal (perhaps, an amplifier driving a DC motor saturates), and our command signal is constant for long periods of time, and occasionally changes abruptly to a new set-point value between -1 and 1. We can ensure that (8.18) holds for all such command signals with the closed-loop convex design specification

$$2\|H_{\mathrm{ac}}\|_{\mathrm{pk\_step}} \le 1. \tag{8.19}$$

This specification ensures that even with the worst case full-scale set-point changes, from -1 to 1 and vice versa, the peak of the actuator signal will not exceed one. By linearity, the specification (8.19) ensures that a set-point change from -0.6 to 0.4 will yield an actuator signal with $\|u\|_\infty \le 0.6$. Roughly speaking, for such a set-point change we are only making use of 60% of our allowable actuator signal size; this may exact a cost in, say, the time required for the commanded variable to converge to within 0.01 of the final value 0.4.

This is illustrated in figure 8.14, which shows two command signals and the associated actuator signals in a control system that satisfies the specification (8.19). The command signal w in figure 8.14(a) is one of the worst case, full-scale set-point changes, and causes the actuator signal u, shown in figure 8.14(b), to nearly saturate. The command $\tilde{w}$ in figure 8.14(c), however, results in the actuator signal in figure 8.14(d), which uses only 48% of the allowable actuator capability.

8.4 Combined Effect of Disturbances and Commands

So far we have treated command inputs and disturbances separately; the specifications we have seen constrain the behavior of the closed-loop system when one, but not both, of these exogenous inputs acts. As a simple example, assume that the system has a single command input, a single disturbance, and a single actuator, so that $n_{\mathrm{c}} = n_{\mathrm{d}} = n_{\mathrm{a}} = 1$. Consider the two specifications

$$\begin{aligned} \mathcal{H}_{\mathrm{env}} &= \{ H \mid s_{\min}(t) \le s(t) \le s_{\max}(t) \quad \text{for all } t \ge 0 \}, \\ \mathcal{H}_{\mathrm{rms\_act}} &= \{ H \mid \|H_{\mathrm{ad}}\|_2 \le 1 \}. \end{aligned}$$

The first specification requires that the step response from w_{c} to z_{c} lie inside the envelope given by $s_{\min}$ and $s_{\max}$, as in figure 8.5. This means that the commanded

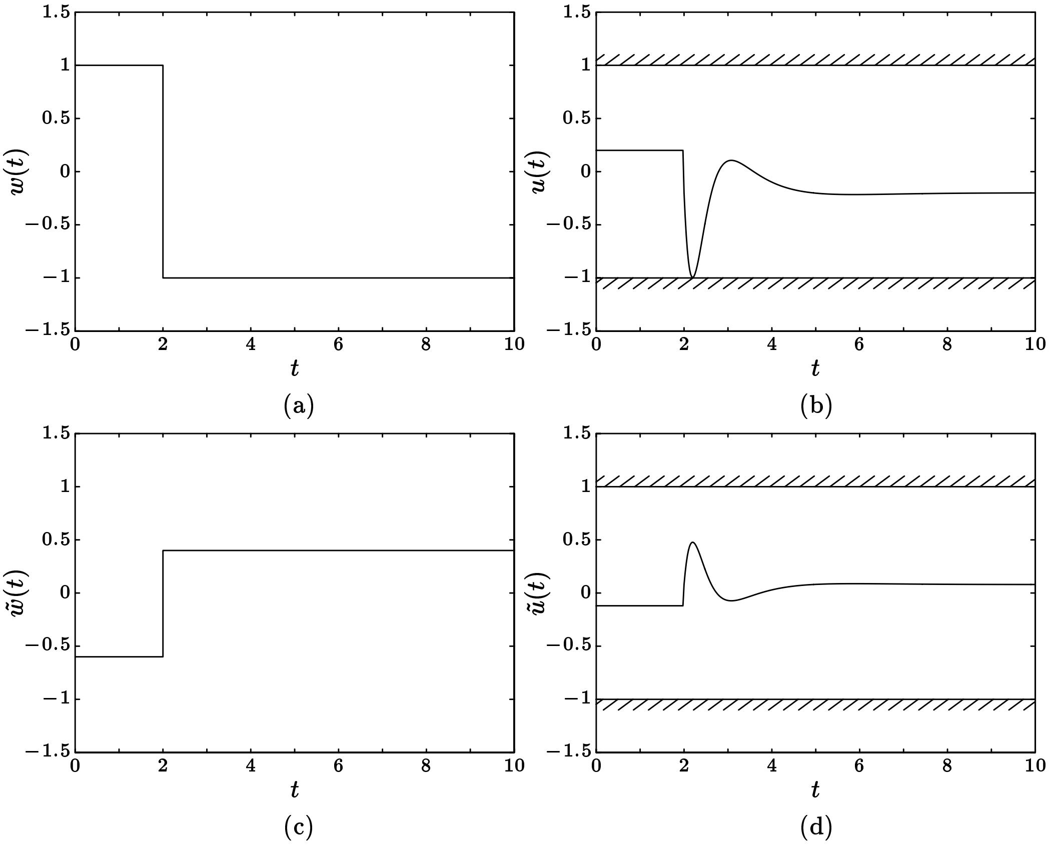


Figure 8.14 The specification (8.19) ensures that the actuator signal magnitude will not exceed one during set-point changes in the range −1 to 1. The input w in (a) shows a set-point change that drives the actuator signal, shown in (b), close to its limit. Because of linearity, smaller set-point changes will result in smaller actuator effort: the set-point change $\tilde{w}$ in (c) produces the actuator signal $\tilde{u}$ in (d), which only uses 48% of the available actuator effort.

variable $z_{\rm c}$ will lie in the given envelope when a step input is applied to $w_{\rm c}$ *and the disturbance input is zero.* The second specification requires that the RMS value of the actuator u be less than one when white noise is applied to $w_{\rm d}$ *and the command input is zero.* If a unit step command input is applied to $w_{\rm c}$ and a white noise is applied to $w_{\rm d}$ simultaneously, the response at $z_{\rm c}$ is simply the sum of the responses with the inputs acting separately. It is quite likely that this $z_{\rm c}$ would not lie in the specified envelope, because of the effect of the disturbance $w_{\rm d}$; similarly the RMS value of u would probably exceed one because of the constant component of the actuator signal that is due to the command.

This phenomenon is a basic consequence of linearity. These separate specifications often suffice in practice since each regulated variable may be mostly dependent on either the command or disturbances. For example, in a given system the disturbance may be small, so the component of the actuator signal due to the disturbance (*i.e.*, $H_{\rm ad}w_{\rm d}$) may be much smaller than the component due to the command (*i.e.*, $H_{\rm ac}w_{\rm c}$); therefore an actuator effort specification that limits the size of $H_{\rm ac}$ will probably acceptably limit the size of u, even though it "ignores" the effect of the disturbance.

We can also describe this phenomenon from a more general viewpoint. Each specification we have considered so far is a specification on some submatrix of H that does not contain all of its columns, and therefore considers the effects of only a subset of the exogenous inputs. In contrast, a specification on a submatrix of H that contains all of its columns will consider the effects of all of the exogenous inputs, acting simultaneously. For example, the RMS actuator effort specification $\mathcal{H}_{\rm rms\_act}$ involves only the submatrix $H_{\rm ad}$, and does not consider the effect of the command on the RMS value of the actuator signal. On the other hand, the specification

$$\mathcal{H}_{\rm rms\_act\_cmb} \stackrel{\Delta}{=} \left\{ H \;\middle|\; \sqrt{\|H_{\rm ad}\|_2^2 + H_{\rm ac}(0)^2} \le 1 \right\},$$

which is a specification on the bigger submatrix $[H_{\rm ac}\ H_{\rm ad}]$ of H, correctly guarantees that the RMS value of u will not exceed one when the command is a unit step and the disturbance is a white noise (and, we should add, $w_{\rm etc} = 0$).

This discussion suggests that a general actuator effort specification should really limit the size of the transfer matrix $[H_{\rm ac}\ H_{\rm ad}]$. Limiting the sizes of $H_{\rm ac}$ and $H_{\rm ad}$ separately will, of course, limit the size of $[H_{\rm ac}\ H_{\rm ad}]$; this corresponds to a *prior allocation of actuator effort between regulation and command-following tasks.*

In cases where different types of models for the commands and disturbances are used (or indeed, different types of models for different components of either), it can be difficult or cumbersome to formulate a sensible specification on the bigger submatrix of H. Returning to our example, let us form a specification on the response of $z_{\rm c}$ that considers both a unit step at $w_{\rm c}$ (a particular signal) and a white noise at $w_{\rm d}$ (a stochastic process). A possible form for such a specification might be

$$\mathcal{H}_{\mathrm{env\_cmb}} \stackrel{\Delta}{=} \{H \mid \mathbf{Prob}\,(s_{\min}(t) \le z_{\mathrm{c}}(t) \le s_{\max}(t) \text{ for } 0 \le t \le 10) \ge 0.90\},$$

where $z_{\mathrm{c}} = H_{\mathrm{cc}}w_{\mathrm{c}} + H_{\mathrm{cd}}w_{\mathrm{d}}$, w_{c} is a unit step, and w_{d} is a white noise. Roughly speaking, the specification $\mathcal{H}_{\mathrm{env\_cmb}}$ requires at least a 90% probability that the envelope constraint $\mathcal{H}_{\mathrm{env}}$ be satisfied over the first ten seconds. $\mathcal{H}_{\mathrm{env\_cmb}}$ limits the effect the (stochastic) disturbance can have during a command step input; however, we do not know whether $\mathcal{H}_{\mathrm{env\_cmb}}$ is convex.

Chapter 9

Differential Sensitivity Specifications

In this chapter we consider specifications that limit the differential sensitivity of the closed-loop transfer matrix with respect to changes in the plant. For certain important cases, these specifications are closed-loop convex. The most general specification that limits differential sensitivity of the closed-loop system is, however, *not* closed-loop convex.

In the previous chapter we considered various specifications that prescribe how the closed-loop system should perform. This included such important considerations as the response of the system to commands and disturbances that may affect the system. In this chapter and the next we focus on another extremely important consideration: how the system *would* perform if the plant were to *change*.

Many control engineers believe that the primary benefits of feedback are those considered in this chapter and the next—robustness or insensitivity of the closed-loop system to variations or perturbations in the plant. From another point of view, the performance of a control system is often limited not by its ability to meet the performance specifications of the previous chapter, but rather by its ability to meet the specifications to be studied in this chapter and the next, which limit the sensitivity or guarantee robustness of the system.

There are several general methods that measure how sensitive the closed-loop system is to changes in the plant:

- *Differential sensitivity*: the size of the derivative of H with respect to P.
- *Worst case perturbation*: the largest change in H that can be caused by a certain specific set of plant perturbations.
- *Margin*: the smallest change in the plant that can cause some specification to be violated.

In this chapter we consider the first method; the other two are discussed in the next chapter.

9.1 Bode's Log Sensitivities

9.1.1 First Order Fractional Sensitivity

H. Bode was the first to systematically study the effect of small changes in closed-loop transfer functions due to small changes in the plant. He considered the I/O transfer function T of the SASS 1-DOF control system (see section 2.3.2),

$$T = \frac{P_0 K}{1 + P_0 K}.$$

He noted that for any frequency s,

$$\frac{\partial T(s)}{\partial P_0(s)} \Bigg/ \frac{T(s)}{P_0(s)} = \frac{1}{1 + P_0(s)K(s)} = S(s), \tag{9.1}$$

which gave the name and symbol S to the classical sensitivity transfer function.

We thus have a basic rule-of-thumb for the 1-DOF control system,

$$\frac{\delta T(s)}{T(s)} \simeq S(s) \frac{\delta P_0(s)}{P_0(s)} \tag{9.2}$$

($\simeq$ means equal to first order), which we interpret as follows: the fractional or relative change in the I/O transfer function is, to first order, the sensitivity transfer function times the fractional change in P_0. For example, and roughly speaking, at a frequency ω with $|S(j\omega)| = 0.1$, a ten percent change in the complex number $P_0(j\omega)$ yields a change in $T(j\omega)$ of only (and approximately) one percent.

An important consequence of (9.2) is that a design specification that limits the first order fractional change in the I/O transfer function with respect to fractional changes in P_0 can be expressed as an equivalent *closed-loop convex* specification that limits the size of the sensitivity transfer function. For example, the specification

$$\left| \frac{\delta T(j\omega)}{T(j\omega)} \right| \lesssim 0.05 \left| \frac{\delta P_0(j\omega)}{P_0(j\omega)} \right| \qquad \text{for } \omega \le \omega_{\rm bw}, \tag{9.3}$$

($\lesssim$ means $\le$ holds to first order), which limits the first order fractional change in T to no more than 5% of the fractional change in P_0 for frequencies less than $\omega_{\rm bw}$, is equivalent to the design specification

$$|S(j\omega)| \le 0.05 \qquad \text{for } \omega \le \omega_{\rm bw}, \tag{9.4}$$

which is closed-loop convex.

The precise meaning of (9.3) is

$$\left| \lim_{\delta P_0(j\omega)\to 0} \frac{\delta T(j\omega)}{T(j\omega)} \Big/ \frac{\delta P_0(j\omega)}{P_0(j\omega)} \right| = |S(j\omega)| \leq 0.05 \qquad \text{for } \omega \leq \omega_{\text{bw}}. \tag{9.5}$$

In many cases, the limit in (9.5) is rapidly approached, *i.e.*, the first order approximation to the fractional change in T accurately predicts the actual fractional change in T due to a (non-differential) change in P_0. We will see two examples of this in sections 9.1.3 and 9.1.4.

9.1.2 Logarithmic Sensitivity

We can express (9.1) as

$$S(s) = \frac{\partial \log T(s)}{\partial \log P_0(s)}. \tag{9.6}$$

For this reason $S(s)$ is called the *logarithmic sensitivity* of T with respect to P_0.

We must be careful about what (9.6) means. By $\log T(s)$ we mean

$$\log T(s) = \log |T(s)| + j\angle T(s) \tag{9.7}$$

where $\angle T(s)$ is a phase angle, in radians, of $T(s)$. Whereas $\log |T(s)|$ is unambiguous and well-defined for all s for which $T(s) \neq 0$, the phase angle $\angle T(s)$ is ambiguous: it is only defined to within an integer multiple of 2π. On any simply-connected region in the complex plane on which $T(s) \neq 0$, it is possible to make a selection of particular phase angles in (9.7) at each s in such a way that $\angle T(s)$ is continuous on the region, and in fact $\log T(s)$ is analytic there. When this process of phase selection is applied along the imaginary axis, it is called *phase unwrapping.*

In particular, if $T(s_0) \neq 0$, then in some small disk around $T(s_0)$ in the complex plane, we can define $\log T(s)$ (*i.e.*, choose the phase angles) so that it is analytic there. Moreover any two such definitions of $\log T(s)$ will differ by a *constant* multiple of $2\pi j$, and therefore yield the same result in the partial derivative in (9.6), evaluated at s_0. A similar discussion applies to the expression $\log P_0(s)$: while it need not make sense as an unambiguous function of the complex variable s over the whole complex plane, the result in (9.6) will nevertheless be unambiguous.

The real part of (9.7), which is the natural log of the magnitude, has an uncommon unit in engineering, called *nepers.* One neper is the gain that corresponds to a phase angle of one radian, and is approximately 8.7dB. In more familiar units, one decibel corresponds to about 6.6 degrees of phase.

(9.2) can be expressed more completely as

$$\delta \log |T(s)| \simeq \Re S(s) \delta \log |P_0(s)| - \Im S(s) \delta \angle P_0(s) \tag{9.8}$$

$$\delta \angle T(s) \simeq \Im S(s) \delta \log |P_0(s)| + \Re S(s) \delta \angle P_0(s). \tag{9.9}$$

These formulas show the first order change in the magnitude and phase of the I/O transfer function caused by magnitude and phase variations in P_0.

9.1.3 Example: Gain Variation

Our first example concerns a change in gain, *i.e.*

$$\delta P_0(s) = \alpha P_0(s), \tag{9.10}$$

so that

$$\delta \log P_0(j\omega) \simeq \alpha.$$

Hence from (9.8–9.9) we have

$$\delta \log |T(j\omega)| \simeq \alpha \Re S(j\omega), \tag{9.11}$$

$$\delta \angle T(j\omega) \simeq \alpha \Im S(j\omega). \tag{9.12}$$

It follows, for example, that the closed-loop convex specification

$$\Re S(j\omega_0) = 0$$

guarantees that the magnitude of the I/O transfer function at the frequency ω_0 is *first order insensitive* to variations in α.

To give a specific example that compares the first order deviation in $|T(j\omega)|$ to the real deviation, we take the standard example plant and controller $K^{(\mathrm{a})}$ described in section 2.4, and consider the effect on $|T(j\omega)|$ of a gain perturbation of $\alpha = 25\%$, which is about 2dB. Figure 9.1 shows:

$$|T(j\omega)| = \left| \frac{P_0^{\mathrm{std}}(j\omega)K^{(\mathrm{a})}(j\omega)}{1 + P_0^{\mathrm{std}}(j\omega)K^{(\mathrm{a})}(j\omega)} \right|,$$

which is the nominal magnitude of the I/O transfer function;

$$|T^{\mathrm{pert}}(j\omega)| = \left| \frac{1.25 P_0^{\mathrm{std}}(j\omega)K^{(\mathrm{a})}(j\omega)}{1 + 1.25 P_0^{\mathrm{std}}(j\omega)K^{(\mathrm{a})}(j\omega)} \right|,$$

which is the actual magnitude with the 25% gain increase in P_0^{std}; and

$$|T^{\mathrm{approx}}(j\omega)| = |T(j\omega)| \exp\left(0.25 \Re \left(\frac{1}{1 + P_0^{\mathrm{std}}(j\omega)K^{(\mathrm{a})}(j\omega)} \right)\right), \tag{9.13}$$

which is the magnitude of the perturbed I/O transfer function predicted by the first order perturbation formula (9.11). For this example, the first order prediction gives a good approximation of the perturbed magnitude of the I/O transfer function, even for this 2dB gain change in P_0^{std}.

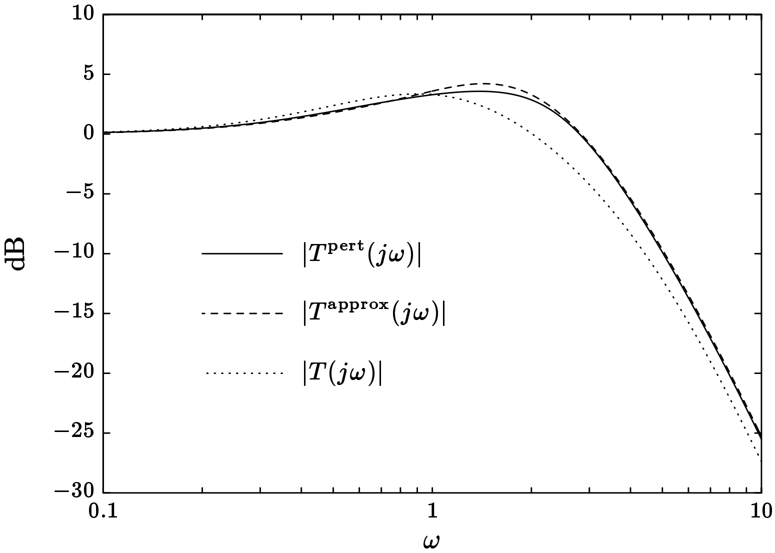


Figure 9.1 When $P_0^{\rm std}$ is replaced by $1.25P_0^{\rm std}$, the I/O transfer function's magnitude changes from $|T|$ to $|T^{\rm pert}|$. $|T^{\rm approx}|$ is a first order approximation of $|T^{\rm pert}|$ computed from (9.13). In this example the effect of a plant gain change as large as 25% is well approximated using the differential sensitivity.

9.1.4 Example: Phase Variation

In this example we study the effects on T of a phase variation in P_0, *i.e.*,

$$P_0(j\omega) + \delta P_0(j\omega) = e^{j\theta(\omega)} P_0(j\omega).$$

In this case we have, from (9.8–9.9),

$$\delta \log |T(j\omega)| \simeq -\theta(\omega) \Im S(j\omega), \tag{9.14}$$

$$\delta \angle T(s) \simeq \theta(\omega) \Re S(j\omega). \tag{9.15}$$

To guarantee that $|T(j\omega_0)|$ is, for example, first order insensitive to phase variations in $P_0(j\omega_0)$, we have the specification

$$\Im S(j\omega_0) = 0,$$

which is closed-loop affine.

We now consider a specific example that compares the actual effect of a phase variation in P_0 to the effect predicted by the first order perturbational analysis (9.14). As above, our plant is the standard example described in section 2.4, together with the same controller $K^{(\rm a)}$. The specific perturbed $P_0^{\rm std}$ is

$$P_0^{\rm std}(s) + \delta P_0^{\rm std}(s) = \frac{1}{s^2} \frac{5-s}{5+s},$$

so that the phase perturbation is

$$\theta(\omega) = 2\left(\tan^{-1}\frac{\omega}{10} - \tan^{-1}\frac{\omega}{5}\right),$$

which is plotted in figure 9.2. The maximum phase shift of −38.9° corresponds to about 6dB of gain variation (see the discussion in section 9.1.2).

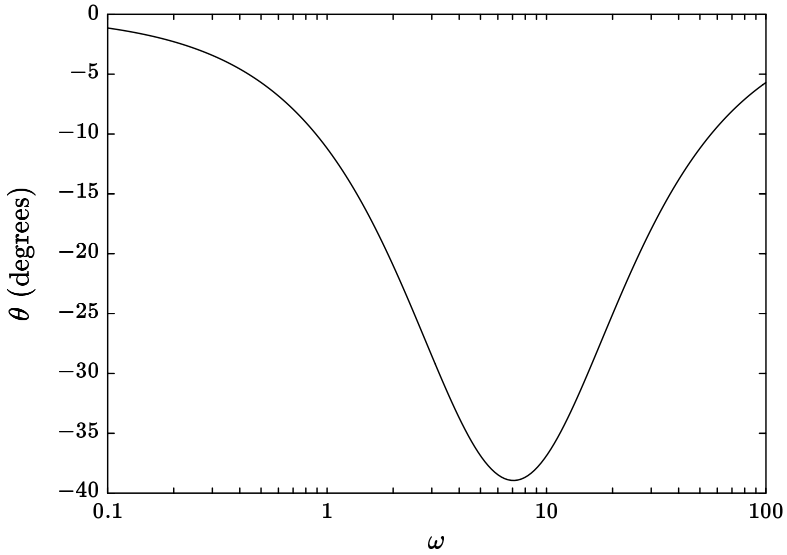


Figure 9.2 A specific phase shift in P_0^{std}.

Figure 9.3 shows the nominal magnitude of T, the actual perturbed magnitude caused by the phase shift in P_0^{std}, and the perturbed magnitude predicted by the first order analysis,

$$|T^{\mathrm{approx}}(j\omega)| = |T(j\omega)| \exp\left(-\theta(\omega)\Im\frac{1}{1+P_0^{\mathrm{std}}(j\omega)K^{(\mathrm{a})}(j\omega)}\right). \tag{9.16}$$

9.1.5 Other Log Sensitivities

We have seen that the logarithmic sensitivity of the I/O transfer function T is given by another closed-loop transfer function, the sensitivity S. Several other important closed-loop transfer functions have logarithmic sensitivities that are also closed-loop transfer functions. Table 9.1 lists some of these.

From the top line of this table we see that a specification such as

$$\left|\frac{\partial \log S(j\omega)}{\partial \log P_0(j\omega)}\right| \le 2 \qquad \text{for } \omega \le \omega_{\mathrm{bw}}, \tag{9.17}$$

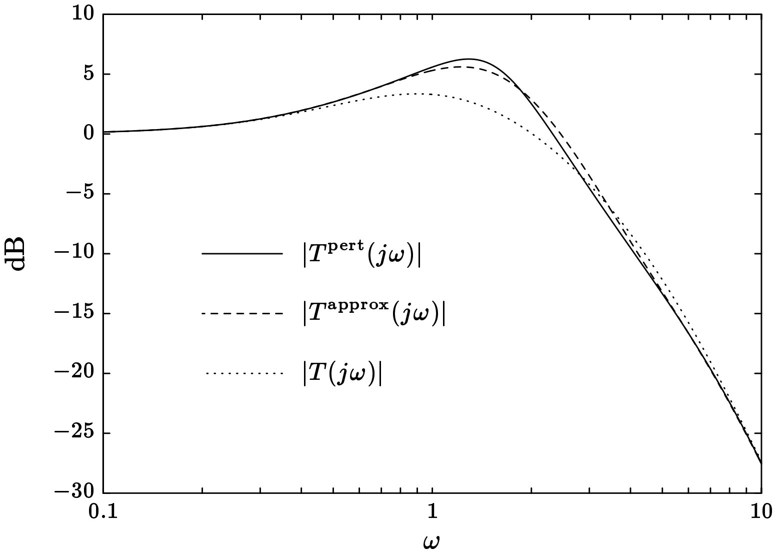


Figure 9.3 When the phase factor $(10 - s)/(10 + s)$ in $P_0^{\rm std}$ is replaced by $(5-s)/(5+s)$, the magnitude of the I/O transfer function changes from $|T|$ to $|T^{\rm pert}|$. $|T^{\rm approx}|$ is a first order approximation of $|T^{\rm pert}|$ computed from (9.16).

H	$\dfrac{\partial \log H}{\partial \log P_0}$
$\dfrac{1}{1+P_0K}$	$\dfrac{-P_0K}{1+P_0K}$
$\dfrac{K}{1+P_0K}$	$\dfrac{-P_0K}{1+P_0K}$
$\dfrac{P_0}{1+P_0K}$	$\dfrac{1}{1+P_0K}$
$\dfrac{P_0K}{1+P_0K}$	$\dfrac{1}{1+P_0K}$

Table 9.1 The logarithmic sensitivity of some important closed-loop transfer functions are also closed-loop transfer functions. In the general case, however, the logarithmic sensitivity of a closed-loop transfer function need not be another closed-loop transfer function.

is equivalent to the closed-loop convex specification

$$\left|\frac{-P_0(j\omega)K(j\omega)}{1+P_0(j\omega)K(j\omega)}\right| \le 2 \qquad \text{for } \omega \le \omega_{\text{bw}}. \tag{9.18}$$

In (9.17) we might interpret $-S$ as the command input to tracking error transfer function, so that the specification (9.17), and hence (9.18), limits the logarithmic sensitivity of the command input to tracking error transfer function with respect to changes in P_0.

9.2 MAMS Log Sensitivity

It is possible to generalize Bode's results to the MAMS case. We consider the MAMS 2-DOF control system (see section 2.3.4), with I/O transfer matrix

$$T = (I + P_0K_{\tilde{y}})^{-1}P_0K_r.$$

If the plant is perturbed so that P_0 becomes $P_0 + \delta P_0$, we have

$$T + \delta T = \left(I + (P_0 + \delta P_0)K_{\tilde{y}}\right)^{-1}(P_0 + \delta P_0)K_r$$

Retaining only first order terms in δP_0 we have

$$\begin{aligned}\delta T &\simeq (I + P_0K_{\tilde{y}})^{-1}\delta P_0K_r - (I + P_0K_{\tilde{y}})^{-1}\delta P_0K_{\tilde{y}}(I + P_0K_{\tilde{y}})^{-1}P_0K_r\\ &= (I + P_0K_{\tilde{y}})^{-1}\delta P_0(I + K_{\tilde{y}}P_0)^{-1}K_r\\ &= S\delta P_0(I + K_{\tilde{y}}P_0)^{-1}K_r, \end{aligned} \tag{9.19}$$

where $S = (I + P_0K_{\tilde{y}})^{-1}$ is the (output-referred) sensitivity matrix of the MAMS 2-DOF control system.

Now suppose that we can express the change in P_0 as

$$\delta P_0 = \delta P_0^{\text{frac}}P_0.$$

We can interpret δP_0^{frac} as an *output-referred* fractional perturbation of P_0, as shown in figure 9.4. Then from (9.19) we have

$$\delta T \simeq S\delta P_0^{\text{frac}}P_0(I + K_{\tilde{y}}P_0)^{-1}K_r = S\delta P_0^{\text{frac}}T,$$

so that

$$\delta T \simeq \delta T^{\text{frac}}T,$$

where

$$\delta T^{\text{frac}} \simeq S\delta P_0^{\text{frac}}. \tag{9.20}$$

This is analogous to (9.2): it states that the output-referred fractional change in the I/O transfer matrix T is, to first order, the sensitivity matrix S times the output-referred fractional change in P_0.

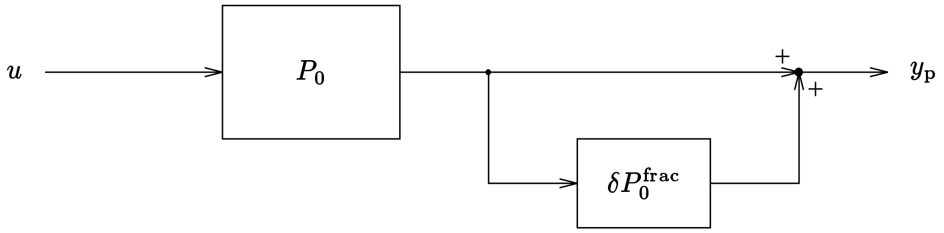


Figure 9.4 An output-referred fractional perturbation of the transfer matrix P_0. In the SASS case, $\delta P_0 \approx \delta \log P_0$.

The design specification,

$$\sigma_\mathrm{max}(\delta T^\mathrm{frac}(j\omega)) \lesssim 0.01 \quad \text{for } \omega \le \omega_\mathrm{bw}, \; \sigma_\mathrm{max}(\delta P_0^\mathrm{frac}(j\omega)) \le 0.20, \qquad (9.21)$$

which limits the first order fractional change in T to 1% over the bandwidth ω_bw, despite variations in P_0 of 20%, is therefore equivalent to the closed-loop convex specification

$$\sigma_\mathrm{max}(S(j\omega)) \le 0.05 \quad \text{for } \omega \le \omega_\mathrm{bw}. \qquad (9.22)$$

We remind the reader that the inequality in (9.21) holds only to first order in δP_0^frac: its precise meaning is

$$\lim_{\delta P_0 \to 0} \frac{\sigma_\mathrm{max}(\delta T^\mathrm{frac}(j\omega))}{\sigma_\mathrm{max}(\delta P_0^\mathrm{frac}(j\omega))} \le \frac{0.01}{0.20} \quad \text{for } \omega \le \omega_\mathrm{bw}.$$

9.2.1 Cruz and Perkins' Comparison Sensitivity

Cruz and Perkins gave another generalization of Bode's log sensitivity to the MAMS 2-DOF control system using the concept of *comparison sensitivity*, in which the perturbed closed-loop system is compared to an equivalent open-loop system.

The open-loop equivalent system consists of $P_0 + \delta P_0$ driven by the unperturbed actuator signal, as shown at the top of figure 9.5. For $\delta P_0 = 0$, the open-loop equivalent system is identical to the closed-loop system shown at the bottom of figure 9.5. For δP_0 nonzero, however, the two systems differ. By comparing the first order changes in these two systems, we can directly see the effect of the feedback on the perturbation δP_0.

The transfer matrix of the open-loop equivalent system is

$$T^\mathrm{ole} = (P_0 + \delta P_0)(I + K_{\tilde{y}} P_0)^{-1} K_r,$$

so that

$$\delta T^\mathrm{ole} = \delta P_0 (I + K_{\tilde{y}} P_0)^{-1} K_r.$$

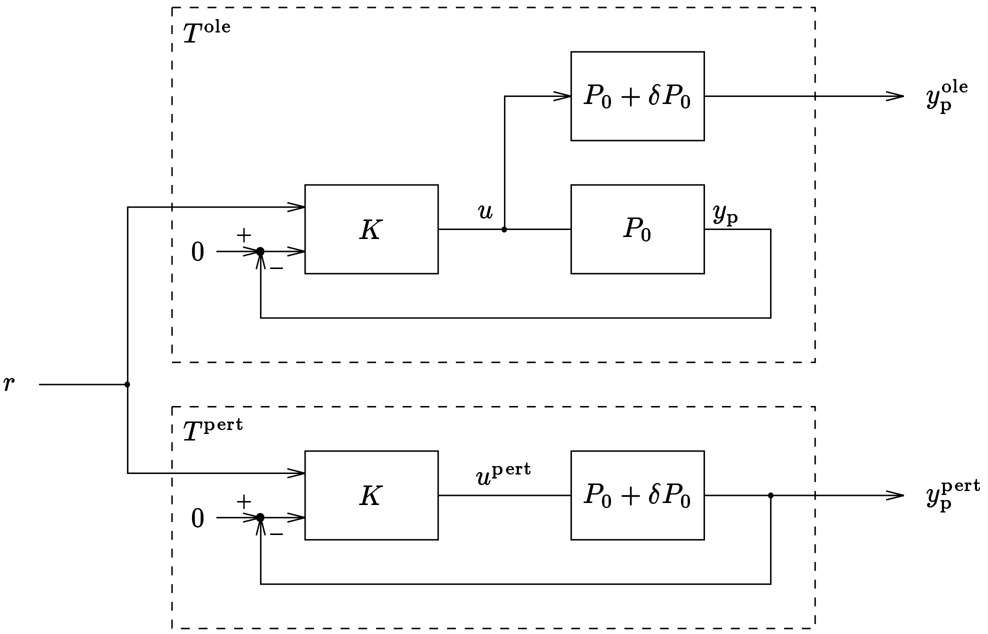


Figure 9.5 In the open-loop equivalent system, shown at top, the actuator signal u drives $P_0+\delta P_0$, so there is no feedback around the perturbation δP_0. The benefit of feedback can be seen by comparing the first order changes in the transfer matrices from r to $y_{\rm p}^{\rm ole}$ and $y_{\rm p}^{\rm pert}$, respectively (see (9.23)).

Comparing this to the first order change in the I/O transfer matrix, given in (9.19), we have

$$\delta T \simeq S \delta T^{\rm ole}. \tag{9.23}$$

This simple equation shows that the first order variation in the I/O transfer matrix is equal to the sensitivity transfer matrix times the first order variation in the open-loop equivalent system. It follows that the specification (9.22) can be interpreted as limiting the sensitivity of the I/O transfer matrix to be no more than 5% of the sensitivity of I/O transfer matrix of the open-loop equivalent system.

9.3 General Differential Sensitivity

The general expression for the first order change in the closed-loop transfer matrix H due to a change in the plant transfer matrix is

$$\begin{aligned} \delta H \simeq\ & \delta P_{zw} + \delta P_{zu} K (I - P_{yu}K)^{-1} P_{yw} + P_{zu} K (I - P_{yu} K)^{-1} \delta P_{yw} \\ & + P_{zu} K (I - P_{yu}K)^{-1} \delta P_{yu} K (I - P_{yu} K)^{-1} P_{yw}. \end{aligned} \tag{9.24}$$

The last term shows that $\partial H/\partial P_{yu}$ (which is a complicated object with four indices) has components that are given by the *product of two* closed-loop transfer functions. It is usually the case that design specifications that limit the size of $\partial H/\partial P_{yu}$ are *not* closed-loop convex, since, roughly speaking, a product can be made small by making either of its terms small.

9.3.1 An Example

Using the standard example plant and controller $K^{(\mathrm{a})}$, described in section 2.4, we consider the sensitivity of the I/O step response with respect to gain variations in P_0^{std}. Since $\delta P_0^{\mathrm{std}} = \alpha P_0^{\mathrm{std}}$, the sensitivity

$$s_\alpha(t) \triangleq \left.\frac{\partial s(t)}{\partial \alpha}\right|_{\alpha=0}$$

is simply the unit step response of the transfer function

$$\frac{P_0^{\mathrm{std}} K^{(\mathrm{a})}}{(1+P_0^{\mathrm{std}} K^{(\mathrm{a})})^2} = ST. \tag{9.25}$$

This transfer function is the product of two closed-loop transfer functions, which is consistent with our general comments above.

In figure 9.6 the actual effect of a 20% gain reduction in P_0^{std} on the step response is compared to the step response predicted by the first order perturbational analysis,

$$s^{\mathrm{approx}}(t) = s(t) - 0.2 s_\alpha(t),$$

with the controller $K^{(\mathrm{a})}$. The step response sensitivity with this controller is shown in figure 9.7. For plant gain changes between ±20%, the first order approximation to the step response falls in the shaded envelope $s(t) \pm 0.2 s_\alpha(t)$.

We now consider the specification

$$|s_\alpha(1)| \le 0.75, \tag{9.26}$$

which limits the sensitivity of the step response at time $t = 1$ to gain variations in P_0^{std}. We will show that this specification is not convex.

The controller $K^{(\mathrm{a})}$ yields a closed-loop transfer matrix $H^{(\mathrm{a})}$ with $s_\alpha^{(\mathrm{a})}(1) = 0.697$, so $H^{(\mathrm{a})}$ satisfies the specification (9.26). The controller $K^{(\mathrm{b})}$ yields a closed-loop transfer matrix $H^{(\mathrm{b})}$ with $s_\alpha^{(\mathrm{b})}(1) = 0.702$, so $H^{(\mathrm{b})}$ also satisfies the specification (9.26). However, the average of these two transfer matrices, $(H^{(\mathrm{a})} + H^{(\mathrm{b})})/2$, has a step response sensitivity at $t = 1$ of 0.786, so $(H^{(\mathrm{a})} + H^{(\mathrm{b})})/2$ does not satisfy the specification (9.26). Therefore the specification (9.26) is not convex.

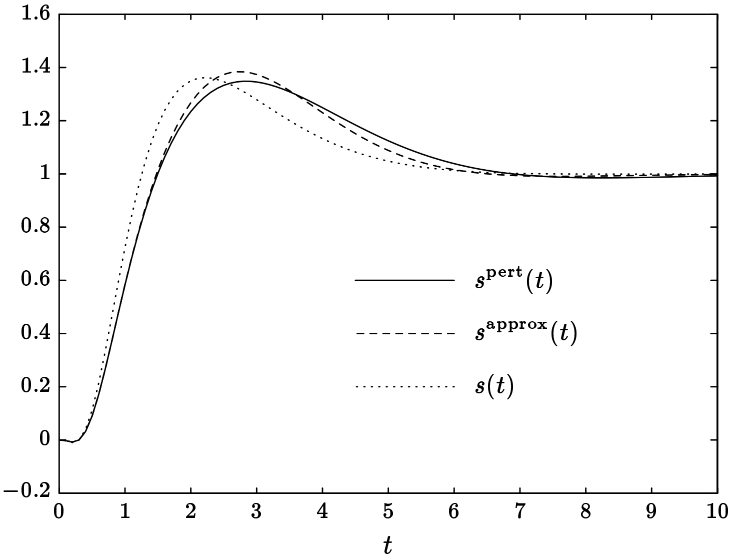


Figure 9.6 When P_0^{std} is replaced by $0.8P_0^{\mathrm{std}}$, the step response changes from s to s^{pert}. The first order approximation of s^{pert} is given by $s^{\mathrm{approx}}(t) = s(t) - 0.2s_\alpha(t)$.

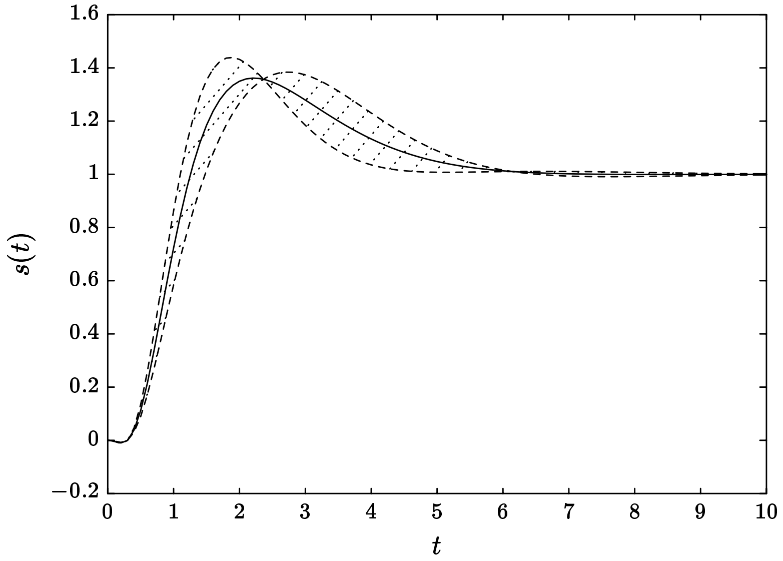


Figure 9.7 The sensitivity of the step response to plant gain changes is shown for the controller $K^{(\mathrm{a})}$. The first order approximation of the step response falls in the shaded envelope when P_0^{std} is replaced by $\alpha P_0^{\mathrm{std}}$, for $0.8 \leq \alpha \leq 1.2$.

9.3.2 Some Convex Approximations

In many cases there are useful convex approximations to specifications that limit general differential sensitivities of the closed-loop system.

Consider the specification

$$|s_\alpha(t)| \le 0.75 \qquad \text{for } t \ge 0, \tag{9.27}$$

which limits the sensitivity of the step response to gain variations in P_0. This specification is equivalent to

$$\|ST\|_{\mathrm{pk\_step}} \le 0.75,$$

which is not closed-loop convex. We will describe two convex approximations for the nonconvex specification (9.27).

Suppose

$$s_{\min}(t) \le s(t) \le s_{\max}(t) \quad \text{for } t \ge 0 \tag{9.28}$$

is a design specification (see figure 8.5). A weak approximation of the sensitivity specification (9.26) (along with the step response specification (9.28)) is that a typical (and therefore fixed) step response satisfies the specification:

$$\|ST_{\mathrm{typ}}\|_{\mathrm{pk\_step}} \le 0.75,$$

where T_{typ} is the transfer function that has unit step response

$$s_{\mathrm{typ}}(t) = \frac{s_{\min}(t) + s_{\max}(t)}{2}.$$

A stronger approximation of (9.27) (along with the step response specification (9.28)) requires that the sensitivity specification be met for *every* step response that satisfies (9.28):

$$\max\{\, \|Sv\|_\infty \mid s_{\min}(t) \le v(t) \le s_{\max}(t) \text{ for } t \ge 0\} \le 0.75.$$

This is an *inner* approximation of (9.27), meaning that it is tighter than (9.27).

Notes and References

Feedback and Sensitivity

The ability of feedback to make a system less sensitive to changes in the plant is discussed in essentially every book on feedback and control; see Mayr [MAY70] for a history of this idea. An early discussion (in the context of feedback amplifiers) can be found in Black [BLA34], in which we find:

> ... by building an amplifier whose gain is deliberately made, say 40dB higher than necessary, and then feeding the output back on the input in such a way as to throw away the excess gain, it has been found possible to effect extraordinary improvement in constancy of amplification ... By employing this feedback principle, amplifiers have been built and used whose gain varied less than 0.01dB with a change in plate voltage from 240V to 260V [whereas] for an amplifier of conventional design and comparable size this change would have been 0.7dB.

For a later discussion see Horowitz [HOR63, CH3]. A concise discussion appears in chapter 1, *On the Advantages of Feedback*, of Callier and Desoer [CD82A].

Differential Sensitivity

Bode [BOD45] was the first to systematically study the effect of small (differential) changes in closed-loop transfer functions due to small (differential) changes in the plant. On page 33 of [BOD45] we find (with our corresponding notation substituted),

> The variation in the final gain characteristic [T] in dB, per dB change in the gain of [P_0], is reduced in the ratio [S].

A recent exposition of differential sensitivity can be found in chapter 3 of Lunze [LUN89].

Comparison Sensitivity

The notion of comparison sensitivity was introduced by Cruz and Perkins in [CP64]; see also the book edited by Cruz [CRU73]. The idea of an open-loop equivalent system, however, is older. In [NGK57, §1.7], it is called the *equivalent cascade configuration* of the control system. Recent discussions of comparison sensitivity can be found in Callier and Desoer [CD82A, CH1] and Anderson and Moore [AM90, §5.3].

Sensitivity Specifications that Limit Control System Performance

The idea that sensitivity or robustness specifications can limit the achievable control system performance is explicitly expressed in, *e.g.*, Newton, Gould, and Kaiser [NGK57, P23]:

> Control systems often employ mechanical, hydraulic, or pneumatic elements which have less reproducible behavior than high quality electric circuit elements. This practical problem often causes the control designer to stop short of an optimum design because he knows full well that the parameters of the physical system may deviate considerably from the data on which he bases his design.

A more recent paper that raised this issue, in the context of regulators designed by state-space methods, is Doyle and Stein [DS81].

Chapter 10

Robustness Specifications via Gain Bounds

In this chapter we consider robustness specifications, which limit the worst case variation in the closed-loop system that can be caused by a specific set of plant variations. We describe a powerful method for formulating inner approximations of robustness specifications as norm-bounds on the nominal closed-loop transfer matrix. These specifications are closed-loop convex.

In the previous chapter we studied the differential sensitivity of the closed-loop system to variations in the plant. Differential sensitivity analysis often gives a good prediction of the changes that occur in the closed-loop system when the plant changes by a moderate (non-vanishing) amount, and hence, designs that satisfy differential sensitivity specifications are often robust to moderate changes in the plant. But differential sensitivity specifications cannot guarantee that the closed-loop system does not change dramatically (*e.g.*, become unstable) when the plant changes by a non-vanishing amount.

In this chapter we describe *robustness specifications*, which, like differential sensitivity specifications, limit the variation in the closed-loop system that can be caused by a change or perturbation in the plant. In this approach, however,

- the sizes of plant variations are explicitly described, *e.g.*, a particular gain varies ±1dB,
- robustness specifications limit the worst case change in the closed-loop system that can be caused by one of the possible plant perturbations.

By contrast, in the differential sensitivity approach,

- the sizes of plant variations are *not* explicitly described; they are vaguely described as “small”,

- differential sensitivity specifications limit the *first order* changes in the closed-loop system that can be caused by the plant perturbations.

Robustness specifications give guaranteed bounds on the performance deterioration, even for "large" plant variations, for which extrapolations from differential sensitivity specifications are dubious. Offsetting this advantage are some possible disadvantages of robustness specifications over differential sensitivity specifications:

- It may not be possible to model the actual variations in the plant in the precise way required by robustness specifications. For example, we may not know whether to expect a ±1dB or a ±0.5dB variation in a particular gain.
- It may not be desirable to limit the worst case variation in the closed-loop system, which results in a conservative design. A specification that limits the *typical* variations in the closed-loop system (however we may define typical) may better capture the designer's intention.

Robustness specifications are often not closed-loop convex, just as the most general specifications that limit differential sensitivity are not closed-loop convex. We will describe a general *small gain method* for formulating convex inner approximations of robustness specifications; the Notes and References for this chapter describe some of the attempts that have been made to make approximations of robustness specifications that are less conservative, but not convex. Since we will be describing convex *approximations* of robustness specifications, we should add the following item to the list of possible disadvantages:

- The small gain based convex inner approximations of robustness specifications can be poor approximations. Thus, designs based on these approximations can be conservative.

This topic is addressed in some of the references at the end of this chapter.

In the next section we give a precise and general definition of a robustness specification, which may appear abstract on first reading. In the remainder of this chapter we describe the framework for small gain methods, and then the small gain methods themselves. The framework and methods are demonstrated on some simple, specific examples that are based on our standard example SASS 1-DOF control system described in section 2.4. These examples continue throughout the chapter.

10.1 Robustness Specifications

10.1.1 Some Definitions

In this section we give a careful definition of a robustness specification; we defer until the next section examples of common robustness specifications. Roughly speaking, a robustness specification requires that some design specification $\mathcal{D}$ must hold, even if the plant P is replaced by any $P^{\rm pert}$ from a specified set $\mathcal{P}$ of possible perturbed plants.

Let us be more precise. Suppose that $\mathcal{P}$ is any set of $(n_w + n_u) \times (n_z + n_y)$ transfer matrices. We will refer to $\mathcal{P}$ as the *perturbed plant set*, and its elements as *perturbed plants.* Let $\mathcal{D}$ denote some design specification, *i.e.*, a boolean function on $n_z \times n_w$ transfer matrices, and let K denote any $n_u \times n_y$ transfer matrix.

Definition 10.1: *We say* $\mathcal{D}$ holds robustly for K and $\mathcal{P}$ *if for each* $P^{\rm pert} \in \mathcal{P}$, $\mathcal{D}$ *holds for the transfer matrix* $P_{zw}^{\rm pert} + P_{zu}^{\rm pert}K(I - P_{yu}^{\rm pert}K)^{-1}P_{yw}^{\rm pert}$.

In words, the design specification $\mathcal{D}$ holds robustly for K and $\mathcal{P}$ if the controller K connected to any of the perturbed plants $P^{\rm pert} \in \mathcal{P}$ yields a closed-loop system that satisfies $\mathcal{D}$. Definition 10.1 is not, by itself, a design specification: it is a property of a controller and a set of transfer matrices. Note also that definition 10.1 makes no mention of the plant P.

Once we have the concept of a design specification holding robustly for a given controller and perturbed plant set, we can define the notion of a robustness specification, which will involve the plant P.

Definition 10.2: *The robustness specification* $\mathcal{D}_{\rm rob}$ *formed from* $\mathcal{D}$, $\mathcal{P}$, *and* P *is given by:*

$\mathcal{D}_{\rm rob}$: $\mathcal{D}$ *holds robustly for* K *and* $\mathcal{P}$,

for every K *that satisfies*

$$H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}. \tag{10.1}$$

Thus a robustness specification is formed from a design specification $\mathcal{D}$, a perturbation plant set $\mathcal{P}$, and the plant P. The reader should note that $\mathcal{D}_{\rm rob}$ is indeed a design specification: it is a boolean function of H. We can interpret $\mathcal{D}_{\rm rob}$ as follows: if H satisfies $\mathcal{D}_{\rm rob}$ and K is any controller that yields the closed-loop transfer matrix H (when connected to P), the closed-loop transfer matrix that results from connecting K to any $P^{\rm pert} \in \mathcal{P}$ will all satisfy $\mathcal{D}$.

A sensible formulation of the plant will including signals such as sensor and actuator noises and the sensor and actuator signals (recall chapter 7). In this case the controller K is uniquely determined by a closed-loop transfer matrix H that is realizable, since $(I - P_{yu}K)^{-1}$ will appear as a submatrix of H, and we can determine K from this transfer matrix. In these cases we may substitute "the K" for "every K" in the definition 10.2.

In many cases, $P \in \mathcal{P}$, and $\mathcal{P}$ consists of transfer matrices that are "close" to P. In this context P is sometimes called the *nominal* plant. In this case the robustness specification $\mathcal{D}_{\rm rob}$ requires that even with the worst perturbed plant substituted for the nominal plant, the design specification $\mathcal{D}$ will continue to hold.

If the design specification $\mathcal{D}$ is $\mathcal{D}_{\rm stable}$, *i.e.*, closed-loop stability (see chapter 7), we call $\mathcal{D}_{\rm rob}$ the *robust stability* design specification associated with $\mathcal{P}$ and P.

Throughout this chapter, P is understood, so the robustness specification will be written

$$\mathcal{D}_{\rm rob}(\mathcal{P}, \mathcal{D}).$$

The robust stability specification associated with the perturbed plant set $\mathcal{P}$ will be denoted

$$\mathcal{D}_{\rm rob\_stab}(\mathcal{P}) \stackrel{\Delta}{=} \mathcal{D}_{\rm rob}(\mathcal{P}, \mathcal{D}_{\rm stable}).$$

10.1.2 Time-Varying and Nonlinear Perturbations

It is possible to extend the perturbed plant set to include time-varying or nonlinear systems, although this requires some care since many of our basic concepts and notation depend on our assumption 2.2 that the plant is LTI. Such an extension is useful for designing a controller K for a nonlinear or time-varying plant $P^{\rm nonlin}$. The controller K is often designed for a "nominal" LTI plant P that is in some sense "close" to $P^{\rm nonlin}$; $P^{\rm nonlin}$ is then considered to be a perturbation of P.

In this section we briefly and informally describe how we may modify our framework to include such nonlinear or time-varying perturbations. In this case the perturbed plant is a nonlinear or time-varying system with $n_w + n_u$ inputs and $n_z + n_y$ outputs. The perturbed closed-loop system, obtained by connecting the controller between the signals y and u of the perturbed plant, is now also nonlinear or time-varying, so the perturbed closed-loop system cannot be described by an $n_z \times n_w$ transfer matrix, as in (10.1). Instead, the closed-loop system is described by the nonlinear or time-varying *closed-loop operator* that maps w into the resulting z.

A design specification will simply be a predicate of the closed-loop system. The only predicate that we will consider is closed-loop stability, which, roughly speaking, means that z is bounded whenever w is bounded (the definition of closed-loop stability given in chapter 7 does not apply here, since it refers to the transfer matrix H). The reader can consult the references given at the end of this chapter for precise and detailed definitions of closed-loop stability of nonlinear or time-varying systems.

The robust stability specification $\mathcal{D}_{\rm rob\_stab}$ will mean that when the (LTI) controller K, which is designed on the basis of the (LTI) plant P, is connected to any of the nonlinear or time-varying perturbed plants in $\mathcal{P}$, the resulting (nonlinear or time-varying) closed-loop system is stable.

10.2 Examples of Robustness Specifications

In this section we consider some examples of robustness specifications, organized by their associated plant perturbation sets. Most of these robustness specifications are *not* convex, but later in this chapter we describe a general method of forming convex inner approximations of these specifications.

10.2.1 Finite Plant Perturbation Sets

A simple but important case occurs when $\mathcal{P}$ is a finite set:

$$\mathcal{P} = \{P_1, \ldots, P_N\}. \tag{10.2}$$

Neglected Dynamics

Recall from chapter 1 that P may be a simple (but not very accurate) model of the system to be controlled. Our perturbed plant set might then be $\mathcal{P} = \{P^{\mathrm{cmplx}}\}$, where P^{cmplx} is a complex, detailed, and accurate model of the system to be controlled. In this case, the robustness specification $\mathcal{D}_{\mathrm{rob}}$ guarantees that the controller we design using the simple model P, will, when connected to P^{cmplx}, satisfy the design specification $\mathcal{D}$.

As a specific example, suppose that our plant is our standard numerical example, the 1-DOF controller described in section 2.4, with plant

$$P = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c} P_0^{\mathrm{std}} & 0 & 0 & P_0^{\mathrm{std}} \\ 0 & 0 & 0 & 1 \\ \hline -P_0^{\mathrm{std}} & -1 & 1 & -P_0^{\mathrm{std}} \end{array}\right].$$

The more detailed model of the system to be controlled might take into account a high frequency resonance and roll-off in the system dynamics and some fast sensor dynamics, neither of which is included in the plant model P:

$$P^{\mathrm{cmplx}} = \left[\begin{array}{ccc|c} P_0^{\mathrm{cmplx}} & 0 & 0 & P_0^{\mathrm{cmplx}} \\ 0 & 0 & 0 & 1 \\ \hline -P_0^{\mathrm{cmplx}} H_{\mathrm{sens}} & -H_{\mathrm{sens}} & 1 & -P_0^{\mathrm{cmplx}} H_{\mathrm{sens}} \end{array}\right]$$

where

$$P_0^{\mathrm{cmplx}}(s) = \frac{P_0^{\mathrm{std}}(s)}{1 + 1.25(s/100) + (s/100)^2}, \qquad H_{\mathrm{sens}}(s) = \frac{1}{1 + s/80}.$$

This is shown in figure 10.1 below (*c.f.* figure 2.11).

For this example, the perturbed plant set is

$$\mathcal{P} = \{P^{\mathrm{cmplx}}\}. \tag{10.3}$$

The robust stability specification $\mathcal{D}_{\mathrm{rob\_stab}}$ that corresponds to (10.3) requires that the controller designed on the basis of the nominal plant P will also stabilize the complex model P^{cmplx} of the system to be controlled. Roughly speaking, $\mathcal{D}_{\mathrm{rob\_stab}}$ requires that the system cannot be made unstable by the high frequency resonance and roll-off in the system dynamics and the dynamics of the sensor, which are ignored in the model P.

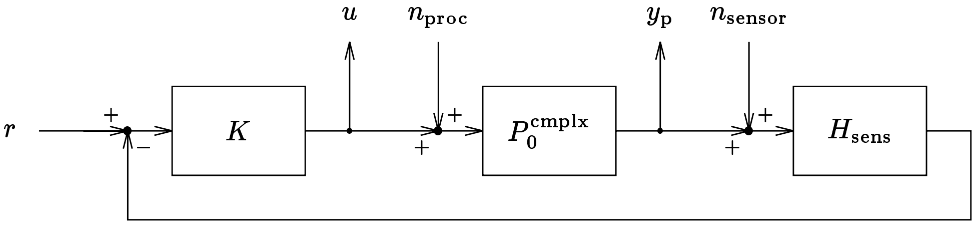


Figure 10.1 The controller K, which is designed on the basis of the model P_0^{std}, is connected to a more detailed model of the system to be controlled, P^{cmplx}. The model P^{cmplx} includes a high frequency resonance and roll-off in P_0^{cmplx}, and the sensor dynamics $(1+s/80)^{-1}$.

Failure Modes

The perturbed plants in (10.2) may represent different *failure modes* of the system to be controlled. For example, P_1 might be a model of the system to be controlled after an actuator has failed (*i.e.*, P_1 is P, but with the column associated with the failed actuator set to zero). In this case the specification of robust stability guarantees that the closed-loop system will remain stable, despite the failures modeled by $P_1, \ldots, P_N$.

10.2.2 Parametrized Plant Perturbations

In some cases the perturbed plant set $\mathcal{P}$ can be described by some parameters that vary over ranges:

$$\mathcal{P} = \left\{ P^{\mathrm{pert}}(\alpha) \;\middle|\; L_1 \le \alpha_1 \le U_1, \ldots, L_k \le \alpha_k \le U_k \right\}.$$

In this case we often have $P \in \mathcal{P}$; the corresponding parameter is called the *nominal parameter*:

$$P = P^{\mathrm{pert}}(\alpha^{\mathrm{nom}}).$$

Parametrized plant perturbation sets can be used to model several different types of plant variation:

- *Component tolerances.* A single controller K is to be designed for many plants, for example, a manufacturing run of the system to be controlled. The controller is designed on the basis of a nominal plant; the parameter variations represent the (slight, one hopes) differences in the individual manufactured systems. Designing a controller that robustly achieves the design specifications avoids the need and cost of tuning each manufactured control system (see section 1.1.5).

- *Component drift or aging.* A controller is designed for a system that is well modeled by P, but it is desired that the system should continue to work if or when the system to be controlled changes, due to aging or drift in its components. Designing a controller that robustly achieves the design specifications avoids the need and cost of periodically re-tuning the control system.

- *Externally induced changes.* The system to be controlled may be well modeled as an LTI system that depends on an external operating condition, which changes slowly compared to the system dynamics. Examples include temperature induced variations in a system, and the effects of varying aerodynamic pressure on aircraft dynamics. Designing a controller that robustly achieves the design specifications can avoid the need for a gain-scheduled or adaptive controller. (See the Notes and References in chapter 2.)

- *Model parameter uncertainty.* A parametrized perturbed plant set can model uncertainty in modeling the system to be controlled (see section 1.1.2). In a model developed from physical principles, the α_i may represent physical parameters such as lengths, masses, and heat conduction coefficients, and the bounds L_i and U_i are then minimum and maximum values that could be expected to occur. In a black box model derived from an identification procedure, the α_i could represent transfer function coefficients, and the L_i and U_i might represent the 90% confidence bands for the identified model.

Example: Gain Margins

Gain margin specifications are examples of classical robustness specifications that are associated with a parametrized plant perturbation set. We consider the classical 1-DOF controller, with perturbed plant set described informally by

$$P_0^{\mathrm{pert}}(s) = \alpha P_0(s), \qquad L \le \alpha \le U.$$

More precisely, we have the perturbed plant set

$$\mathcal{P} = \left\{ \left[\begin{array}{ccc|c} \alpha P_0 & 0 & 0 & \alpha P_0 \\ 0 & 0 & 0 & 1 \\ \hline -\alpha P_0 & -1 & 1 & -\alpha P_0 \end{array} \right] \;\middle|\; L \le \alpha \le U \right\}, \qquad \alpha^{\mathrm{nom}} = 1, \tag{10.4}$$

where $0 < L \le 1 \le U$.

The specification of robust stability with the perturbed plant set (10.4) is described in classical terms as a positive gain margin of $20\log_{10} U$dB and a negative gain margin of $20\log_{10} L$dB.

As an example we will use later, the robustness specification that requires gain margins of +4dB and −3.5dB is given by

$$\mathcal{D}_{+4,-3.5\mathrm{db\_gm}}: \qquad \mathcal{D}_{\mathrm{rob\_stab}}(\mathcal{P}) \quad \text{with } L = 0.668,\ U = 1.585, \tag{10.5}$$

with the plant perturbation set (10.4).

Example: Pole Variation

The parameter vector α may determine pole or zero locations in the plant. As a specific example, consider the perturbed plant set for the standard 1-DOF example of section 2.4, described informally by

$$P_0^{\mathrm{pert}}(s) = \frac{1}{s^2}\frac{\alpha - s}{\alpha + s}, \qquad 5 \le \alpha \le 15, \qquad \alpha^{\mathrm{nom}} = 10. \tag{10.6}$$

Some of these phase variations in P_0^{std} are shown in figure 10.2.

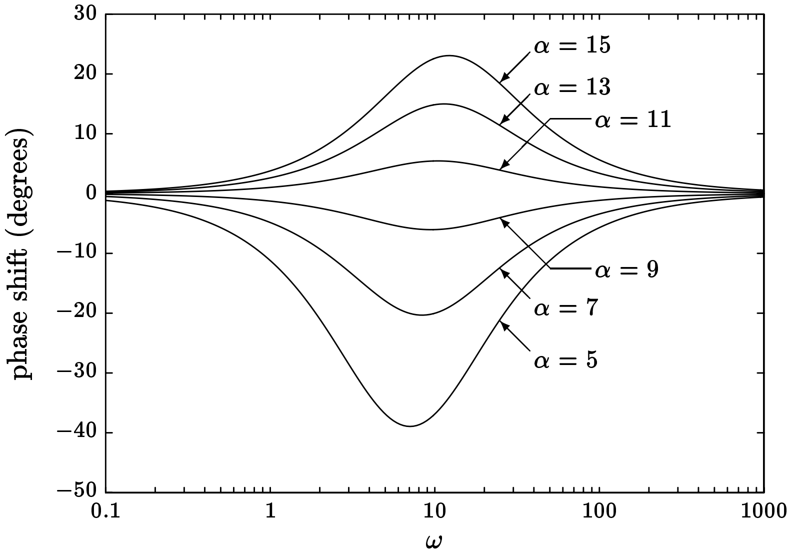


Figure 10.2 The perturbed plants described by (10.6) consist of a phase shift in P_0^{std}: shown here are the phase shifts for $\alpha - \alpha^{\mathrm{nom}} = \pm 1, \pm 3, \pm 5$. (*c.f.* figure 9.2, which shows a particular phase shift.)

The robust stability specification $\mathcal{D}_{\mathrm{rob\_stab}}$ is a strengthening of the stability specification $\mathcal{D}_{\mathrm{stable}}$: $\mathcal{D}_{\mathrm{rob\_stab}}$ requires that the controller stabilize not only the plant P, but also the perturbed plants (10.6). $\mathcal{D}_{\mathrm{rob\_stab}}$ can be thought of as a type of phase margin specification.

10.2.3 Unknown-but-Bounded Transfer Function Perturbations

It is often useful to model the uncertainty in the plant (as a model of the system to be controlled) as frequency-dependent errors in the frequency responses of its entries. Such plant perturbation sets can be used to account for:

- *Model uncertainty.* The plant transfer functions may inaccurately model the system to be controlled because of measurement or identification errors. For example, the transfer functions of the system to be controlled may have been measured at each frequency to an accuracy of 1%, or these measurements might be repeatable only to 1%.

- *High frequency parasitic dynamics.* A model of the system to be controlled may become less accurate at high frequencies because of unknown or unmodeled parasitic dynamics. Moreover these parasitic dynamics may change with time or other physical parameters, and so cannot be confidently modeled. In electrical systems, for example, we may have small stray capacitances and (self and mutual) inductances between conductors; these parasitic dynamics can change significantly when the electrical or magnetic environment of the system is changed.

  In state-space plant descriptions, the addition of high frequency parasitic dynamics is called a *singular perturbation*, because the perturbed plant has more states than the plant.

Since these plant perturbation sets cannot be described by the variation of a small number of real parameters, they are sometimes called *nonparametric plant perturbations.*

There are some subtle distinctions between intentionally neglected system dynamics that could in principle be modeled, and parasitic dynamics that cannot be confidently modeled. For example, a model of a mechanical system may be developed on the assumption that a drive train is rigid, an assumption that is good at low frequencies, but poor at high frequencies. If the high frequency dynamics of this drive train could be accurately modeled or consistently measured, then we could develop a more accurate (and more complex) model of the system to be controlled, as in section 10.2.1. However, it may be the case that these high frequency dynamics are very sensitive to minor physical variations in the system, such as might be induced by temperature changes, bearing wear, and so on. In this case the drive train dynamics could reasonably be modeled as an unknown transfer function that is close to one at low frequencies, and less close at high frequencies.

Example: Relative Uncertainty in P_0

We consider again our standard SASS 1-DOF control system example. Suppose we believe that the relative or fractional error in the transfer function P_0 (as a model of the system to be controlled) is about 20% at low frequencies (say, $\omega \leq 5$), and much larger at high frequencies (say, up to 400% for $\omega \geq 500$). We define the relative error as

$$P_0^{\mathrm{rel\_err}} \triangleq \frac{P_0^{\mathrm{pert}} - P_0}{P_0},$$

so that

$$P_0^{\mathrm{pert}} = (1 + P_0^{\mathrm{rel\_err}})P_0. \tag{10.7}$$

Our plant model uncertainty can be described by a frequency-dependent limit on the magnitude of $P_0^{\mathrm{rel\_err}}$, *e.g.*,

$$\|P_0^{\mathrm{rel\_err}}/W_{\mathrm{rel\_err}}\|_\infty \leq 1, \tag{10.8}$$

where

$$W_{\mathrm{rel\_err}}(s) = 0.2\frac{1+s/10}{1+s/200}. \tag{10.9}$$

We interpret $|W_{\mathrm{rel\_err}}(j\omega)|$, which is plotted in figure 10.3, as the maximum relative error in $P_0(j\omega)$. We say that $P_0^{\mathrm{rel\_err}}$ is an *unknown-but-bounded* transfer function. One interpretation is shown in figure 10.4.

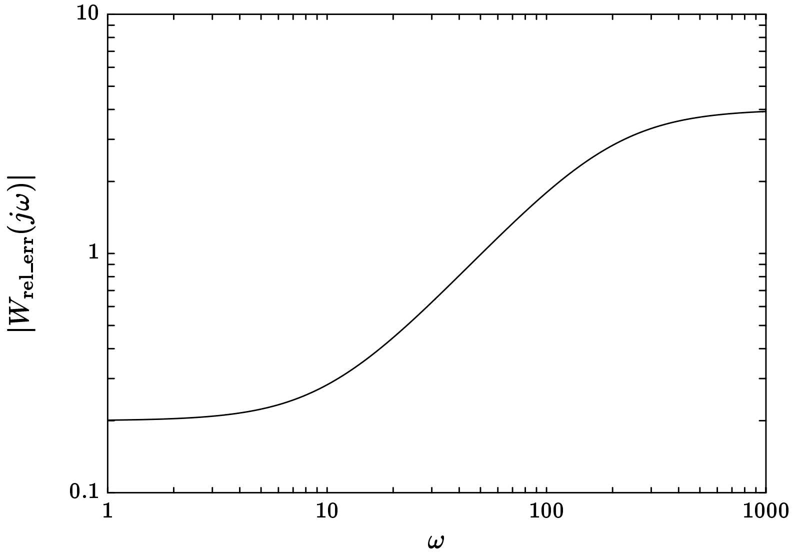


Figure 10.3 From (10.8), $|W_{\mathrm{rel\_err}}(j\omega)|$ is the maximum relative error in $P_0(j\omega)$. This represents a relative error of 20% at low frequencies and up to 400% at high frequencies.

The perturbed plant set for this example is thus

$$\mathcal{P} = \left\{ \left[\begin{array}{ccc|c} P_0^{\mathrm{pert}} & 0 & 0 & P_0^{\mathrm{pert}} \\ 0 & 0 & 0 & 1 \\ \hline -P_0^{\mathrm{pert}} & -1 & 1 & -P_0^{\mathrm{pert}} \end{array} \right] \;\middle|\; \left\|P_0^{\mathrm{rel\_err}}/W_{\mathrm{rel\_err}}\right\|_\infty \leq 1 \right\}, \tag{10.10}$$

where P_0^{pert} is given by (10.7).

We note for future reference two robustness specifications using the perturbed plant set (10.10). The first is robust stability, $\mathcal{D}_{\mathrm{rob\_stab}}$, and the second is the

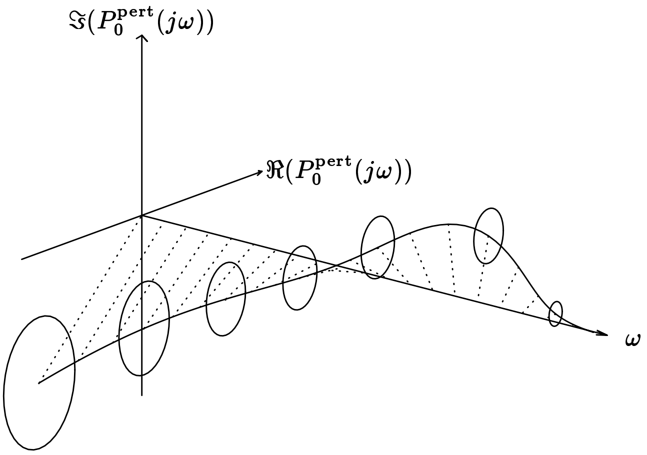

$\Im(P_0^{\mathrm{pert}}(j\omega))$
$\Re(P_0^{\mathrm{pert}}(j\omega))$
ω

Figure 10.4 The complex transfer function $P_0^{\mathrm{pert}}(j\omega)$ is shown versus frequency ω. Circles that are centered at the nominal plant transfer function, $P_0(j\omega)$, with radius $|W_{\mathrm{rel\_err}}(j\omega)P_0(j\omega)|$ are also shown. (10.7) and (10.8) require that the perturbed plant transfer function, P_0^{pert}, must lie within the region enclosed by these circles.

stronger specification that these plant perturbations never cause the RMS gain from the reference input to the actuator signal to exceed 75:

$$\mathcal{D}_{\mathrm{rob}}\left(\mathcal{P},\ \|T/P_0\|_\infty \le 75\right) \qquad (10.11)$$

(T/P_0 is the transfer function from the reference input to the actuator signal).

10.2.4 Neglected Nonlinearities

Example: Actuator Saturation

A common nonlinearity encountered in systems to be controlled is saturation of the actuator signals, shown in figure 10.5. This system is described by

$$\begin{bmatrix} z \\ y \end{bmatrix} = P^{\mathrm{lin}} \begin{bmatrix} w \\ u^{\mathrm{sat}} \end{bmatrix}, \qquad (10.12)$$

$$u_i^{\mathrm{sat}}(t) = S_i\,\mathbf{Sat}(u_i(t)/S_i), \quad i = 1, \ldots, n_u, \qquad (10.13)$$

where P^{lin} is an LTI system and the *unit saturation function* $\mathbf{Sat} : \mathbf{R} \to \mathbf{R}$ is defined by

$$\mathbf{Sat}(a) \triangleq \begin{cases} a & |a| \le 1 \\ 1 & a > 1 \\ -1 & a < -1. \end{cases} \tag{10.14}$$

S_i is called the saturation threshold of the ith actuator: if the magnitude of each actuator signal is always below its saturation threshold, then the system is LTI, with transfer matrix P^{lin}.

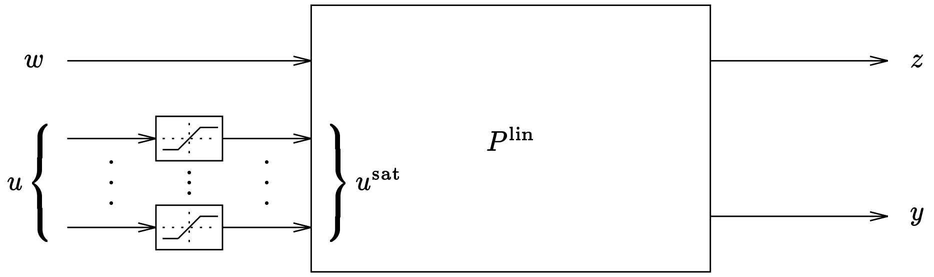


Figure 10.5 A system to be controlled consists of the LTI system P^{lin}, driven by a saturated actuator signal, u^{sat}. The block diagram for the saturation nonlinearity shows its graph.

One approach to designing a controller for this nonlinear system is to define the LTI plant $P = P^{\text{lin}}$, and consider the saturation as a nonlinear perturbation of P. Thus we consider the perturbed plant set consisting of the single nonlinear system given by (10.12–10.13):

$$\mathcal{P} = \{P^{\text{nonlin}}\}. \tag{10.15}$$

By designing an LTI controller for P that yields robust stability for (10.15), we are guaranteed that when the controller is connected to the nonlinear system to be controlled, the resulting nonlinear closed-loop system will at least be stable. If the actuator signals only occasionally exceed their thresholds, then the closed-loop transfer matrix H can give a good approximation of the behavior of the nonlinear closed-loop system. (Another useful approach, also based on the idea of considering the saturators as a perturbation of an LTI plant, is described in the Notes and References.)

We should also mention the *describing function method*, a heuristic approach that is often successful in practice. Roughly speaking, the describing function method approximates the effect of the saturators as a gain reduction in the actuator channels: such perturbations are handled via gain margin specifications. Of course, gain margin specifications do not guarantee that the nonlinear closed-loop system is stable.

10.3 Perturbation Feedback Form

In many cases the perturbed plant set $\mathcal{P}$ can be represented as the nominal plant with an internal feedback, as shown in figure 10.6. When the internal feedback Δ is zero, we recover the nominal plant P; each perturbed plant in $\mathcal{P}$ corresponds to a particular feedback $\Delta \in \mathbf{\Delta}$, where $\mathbf{\Delta}$ is a set of transfer matrices of the appropriate size.

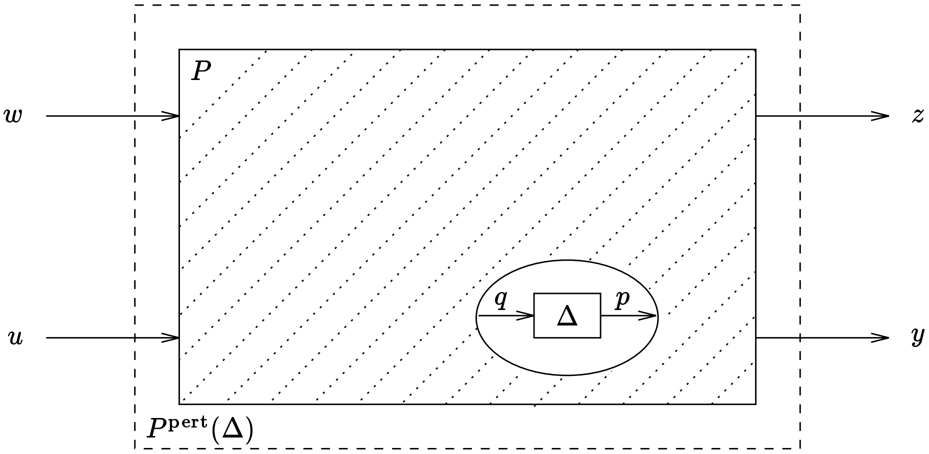


Figure 10.6 Each perturbed plant is equivalent to the nominal plant modified by the internal feedback Δ.

We will call Δ the *feedback perturbation.* The perturbed plant that results from the feedback perturbation Δ will be denoted $P^{\rm pert}(\Delta)$, and $\mathbf{\Delta}$ will be called the *feedback perturbation set* that corresponds to $\mathcal{P}$:

$$\mathcal{P} = \left\{ P^{\rm pert}(\Delta) \;\middle|\; \Delta \in \mathbf{\Delta} \right\}. \qquad (10.16)$$

The symbol Δ emphasizes its role in "changing" the plant P into the perturbed plant $P^{\rm pert}$.

The input signal to the perturbation feedback, denoted q, can be considered an *output signal* of the plant P. Similarly, the output signal from the perturbation feedback, denoted p, can be considered an *input signal* to the plant P. Throughout this chapter we will assume that the exogenous input signal w and the regulated output signal z are augmented to contain p and q, respectively:

$$w = \begin{bmatrix} \tilde{w} \\ p \end{bmatrix}, \qquad z = \begin{bmatrix} \tilde{z} \\ q \end{bmatrix},$$

where $\tilde{w}$ and $\tilde{z}$ denote the original signals from figure 10.6. This is shown in figure 10.7.

To call p an exogenous input signal can be misleading, since this signal does not originate "outside" the plant, like command inputs or disturbance signals, as

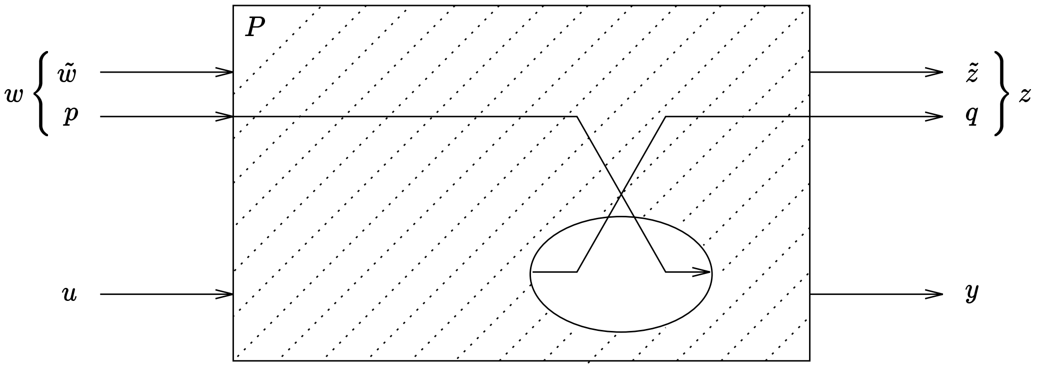


Figure 10.7 The plant P showing p as part of the exogenous input signal w and q as part of the regulated output signal z. By connecting the feedback perturbation Δ between q and p, we recover the perturbed plant $P^{\rm pert}(\Delta)$.

the term exogenous implies. We can think of the signal p as originating outside the *nominal* plant, as in figure 10.6.

To describe a perturbation feedback form of a perturbed plant set $\mathcal{P}$, we give the (augmented) plant transfer matrix

$$P = \left[\begin{array}{c|c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}p} & P_{\tilde{z}u} \\ \hline P_{q\tilde{w}} & P_{qp} & P_{qu} \\ \hline P_{y\tilde{w}} & P_{yp} & P_{yu} \end{array}\right],$$

along with the set $\boldsymbol{\Delta}$ of perturbation feedbacks. Our original perturbed plant can be expressed as

$$P^{\rm pert}(\Delta) = \left[\begin{array}{c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}u} \\ \hline P_{y\tilde{w}} & P_{yu} \end{array}\right] + \left[\begin{array}{c} P_{\tilde{z}p} \\ \hline P_{yp} \end{array}\right] \Delta(I - P_{qp}\Delta)^{-1} \left[\begin{array}{c|c} P_{q\tilde{w}} & P_{qu} \end{array}\right]. \quad (10.17)$$

The perturbation feedback form, *i.e.*, the transfer matrix P in (10.17) and the set $\boldsymbol{\Delta}$, is not uniquely determined by the perturbed plant set $\mathcal{P}$. This fact will be important later.

When $\mathcal{P}$ contains nonlinear or time-varying systems, the perturbation feedback form consists of an LTI P and a set $\boldsymbol{\Delta}$ of nonlinear or time-varying systems. Roughly speaking, the feedback perturbation Δ represents the extracted nonlinear or time-varying part of the system. We will see an example of this later.

10.3.1 Perturbation Feedback Form: Closed-Loop

Suppose now that the controller K is connected to the perturbed plant $P^{\rm pert}(\Delta)$, as shown in figures 10.8 and 10.9.

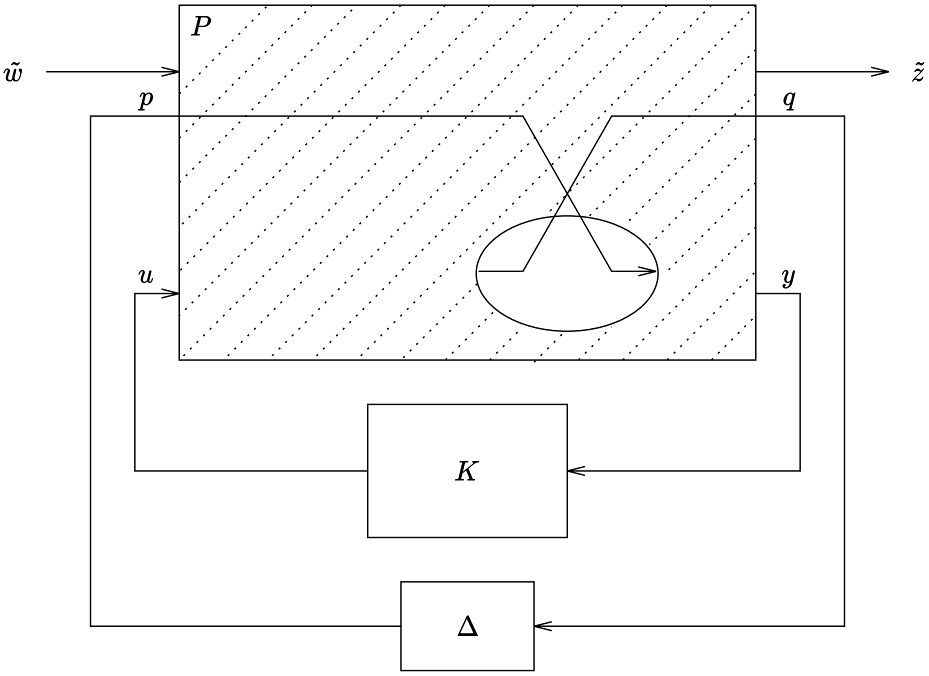


Figure 10.8 When the perturbed plant set is expressed in the perturbation feedback form shown in figure 10.6, the perturbed closed-loop system can be represented as the nominal plant P, with the controller K connected between y and u as usual, and the perturbation feedback Δ connected between q and p.

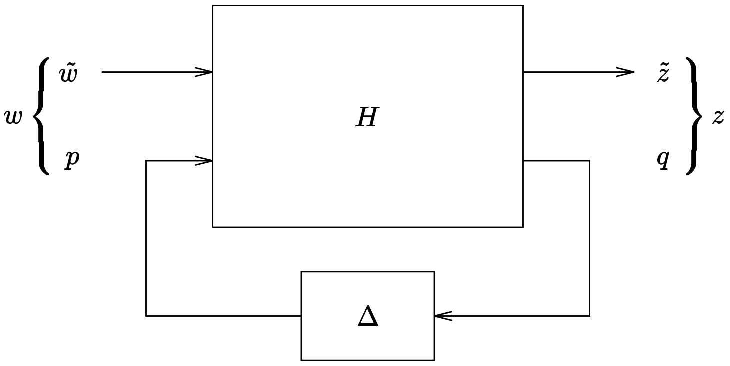


Figure 10.9 The perturbed closed-loop system can be represented as the nominal closed-loop system with feedback Δ connected from q (a part of z) to p (a part of w). Note the similarity to figure 2.2.

By substituting (10.17) into (2.7) we find that the transfer matrix of the perturbed closed-loop system is

$$H^{\text{pert}}(\Delta) = H_{\tilde{z}\tilde{w}} + H_{\tilde{z}p}\Delta(I - H_{qp}\Delta)^{-1}H_{q\tilde{w}}, \tag{10.18}$$

where

$$H_{\tilde{z}\tilde{w}} = P_{\tilde{z}\tilde{w}} + P_{\tilde{z}u}K(I - P_{yu}K)^{-1}P_{y\tilde{w}} \tag{10.19}$$
$$H_{\tilde{z}p} = P_{\tilde{z}p} + P_{\tilde{z}u}K(I - P_{yu}K)^{-1}P_{yp} \tag{10.20}$$
$$H_{q\tilde{w}} = P_{q\tilde{w}} + P_{qu}K(I - P_{yu}K)^{-1}P_{y\tilde{w}} \tag{10.21}$$
$$H_{qp} = P_{qp} + P_{qu}K(I - P_{yu}K)^{-1}P_{yp}. \tag{10.22}$$

Note the similarities between figures 10.9 and 2.2, and the corresponding equations (10.18) and (2.7). Figure 2.2 and equation (2.7) show the effect of connecting the controller to the nominal plant to form the nominal closed-loop system; figure 10.9 and equation (10.18) show the effect of connecting the feedback perturbation Δ to the nominal closed-loop system to form the perturbed closed-loop system.

We may interpret

$$H^{\text{pert}}(\Delta) - H_{\tilde{z}\tilde{w}} = H_{\tilde{z}p}\Delta(I - H_{qp}\Delta)^{-1}H_{q\tilde{w}} \tag{10.23}$$

as the change in the closed-loop transfer matrix that is caused by the feedback perturbation Δ. We have the following interpretations:

- $H_{\tilde{z}\tilde{w}}$ is the closed-loop transfer matrix of the nominal system, before its exogenous input and regulated output were augmented with the signals p and q.

- $H_{q\tilde{w}}$ is the closed-loop transfer matrix from the original exogenous input signal $\tilde{w}$ to q. If $H_{q\tilde{w}}$ is "large", then so will be the signal q that drives or excites the feedback perturbation Δ.

- $H_{\tilde{z}p}$ is the closed-loop transfer matrix from p to the original regulated output signal $\tilde{z}$. If $H_{\tilde{z}p}$ is "large", then so will be the effect on $\tilde{z}$ of the signal p, which is generated by the feedback perturbation Δ.

- H_{qp} is the closed-loop transfer matrix from p to q. We can interpret H_{qp} as the feedback seen by Δ, looking into the nominal closed-loop system.

Thus, if the three closed-loop transfer matrices $H_{\tilde{z}p}$, $H_{q\tilde{w}}$, and H_{qp} are all "small", then our design will be "robust" to the perturbations, *i.e.*, the change in the closed-loop transfer matrix, which is given in (10.23), will also be "small". This vague idea will be made more precise later in this chapter.

10.3.2 Examples of Perturbation Feedback Form

In this section $\star$ will denote a transfer function that we have already given elsewhere. In this way we emphasize the transfer functions that are directly relevant to the perturbation feedback form.

Neglected Dynamics

Figure 10.10 shows one way to represent the perturbed plant set $\mathcal{P} = \{P^{\mathrm{cmplx}}\}$ described in section 10.2.1 in perturbation feedback form. In this block diagram, the perturbation feedback Δ acts as a switch: $\Delta = 0$ yields the nominal plant; $\Delta = I$ turns on the perturbation, to yield the perturbed plant P^{cmplx}.

This perturbation feedback form is described by the augmented plant

$$\left[\begin{array}{c|c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}p} & P_{\tilde{z}u} \\ \hline P_{q\tilde{w}} & P_{qp} & P_{qu} \\ \hline P_{y\tilde{w}} & P_{yp} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|cc|c} \star & \star & \star & P_{\mathrm{err}}^{(1)} & 0 & \star \\ \star & \star & \star & 0 & 0 & \star \\ \hline P_0^{\mathrm{std}} & 0 & 0 & 0 & 0 & P_0^{\mathrm{std}} \\ P_0^{\mathrm{std}} & 1 & 0 & P_{\mathrm{err}}^{(1)} & 0 & P_0^{\mathrm{std}} \\ \hline \star & \star & \star & -P_{\mathrm{err}}^{(1)} & -P_{\mathrm{err}}^{(2)} & \star \end{array}\right], \qquad (10.24)$$

where

$$P_{\mathrm{err}}^{(1)}(s) = \frac{-1.25(s/100) - (s/100)^2}{1 + 1.25(s/100) + (s/100)^2}, \qquad P_{\mathrm{err}}^{(2)}(s) = \frac{-s/80}{1 + s/80},$$

and the feedback perturbation set

$$\boldsymbol{\Delta} = \{I\}. \qquad (10.25)$$

When the controller is connected, we have

$$H_{q\tilde{w}} = \left[\begin{array}{ccc} P_0^{\mathrm{std}} S & -T & T \\ P_0^{\mathrm{std}} S & S & T \end{array}\right], \qquad (10.26)$$

$$H_{\tilde{z}p} = \left[\begin{array}{cc} P_{\mathrm{err}}^{(1)} S & -P_{\mathrm{err}}^{(2)} T \\ -P_{\mathrm{err}}^{(1)} T/P_0^{\mathrm{std}} & -P_{\mathrm{err}}^{(2)} T/P_0^{\mathrm{std}} \end{array}\right], \qquad (10.27)$$

$$H_{qp} = \left[\begin{array}{cc} -P_{\mathrm{err}}^{(1)} T & -T P_{\mathrm{err}}^{(2)} \\ P_{\mathrm{err}}^{(1)} S & -T P_{\mathrm{err}}^{(2)} \end{array}\right]. \qquad (10.28)$$

Gain Margin: Perturbation Feedback Form 1

The perturbed plant set for the classical gain margin specification, given by $\mathcal{P}$ in (10.4), can be expressed in the perturbation feedback form shown in figure 10.11.

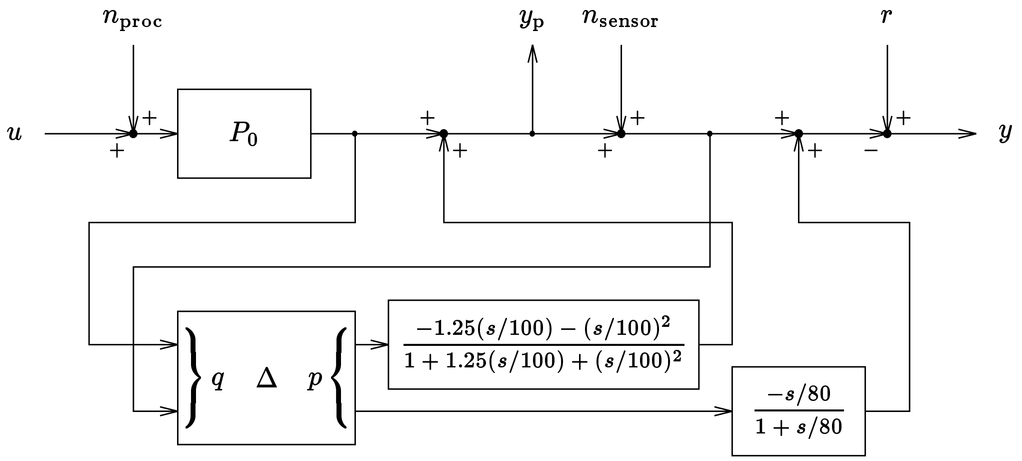


Figure 10.10 Perturbation feedback form for the perturbed plant set that consists of the single transfer matrix $P^{\rm cmplx}$. The feedback perturbation Δ acts as a switch: $\Delta=0$ yields the nominal plant; $\Delta=I$ yields the perturbed plant $P^{\rm cmplx}$.

This perturbation feedback form is described by the augmented plant

$$\left[\begin{array}{c|c|c} P_{\tilde z\tilde w} & P_{\tilde z p} & P_{\tilde z u} \\ \hline P_{q\tilde w} & P_{qp} & P_{qu} \\ \hline P_{y\tilde w} & P_{yp} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c|c} \star & \star & \star & P_0 & \star \\ \star & \star & \star & 0 & \star \\ \hline 0 & 0 & 0 & 0 & 1 \\ \hline \star & \star & \star & -P_0 & \star \end{array}\right], \tag{10.29}$$

and the perturbation feedback set

$$\mathbf{\Delta} = [L-1, U-1], \tag{10.30}$$

which is an interval. Thus, the feedback perturbations are real constants, or gains. Informally, the perturbation Δ causes P_0 to become $(1+\Delta)P_0$.

For this perturbation feedback form, the transfer matrices $H_{q\tilde w}$, $H_{\tilde z p}$, and H_{qp} are given by

$$H_{q\tilde w} = \left[\begin{array}{ccc} -T & -T/P_0 & T/P_0 \end{array}\right], \tag{10.31}$$

$$H_{\tilde z p} = \left[\begin{array}{c} P_0 S \\ -T \end{array}\right], \tag{10.32}$$

$$H_{qp} = -T. \tag{10.33}$$

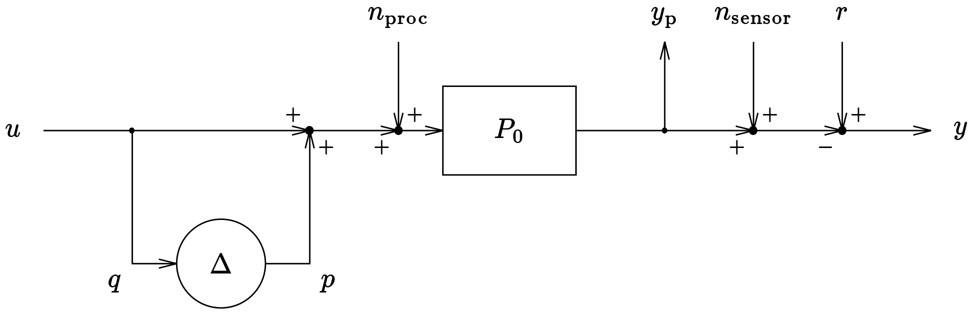


Figure 10.11 One possible perturbation feedback form for the classical gain margin specification.

Gain Margin: Perturbation Feedback Form 2

The same perturbed plant set, (10.4), can be described in the different perturbation feedback form shown in figure 10.12, for which

$$\left[\begin{array}{c|c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}p} & P_{\tilde{z}u} \\ \hline P_{q\tilde{w}} & P_{qp} & P_{qu} \\ \hline P_{y\tilde{w}} & P_{yp} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c|c} \star & \star & \star & P_0 & \star \\ \star & \star & \star & 0 & \star \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline \star & \star & \star & -P_0 & \star \end{array}\right] \tag{10.34}$$

(only one entry differs from (10.29)), and

$$\boldsymbol{\Delta} = [1 - 1/L, 1 - 1/U], \tag{10.35}$$

which is a different interval than (10.30). For this perturbation feedback form, Δ represents a constant that causes P_0 to become $P_0/(1 - \Delta)$.

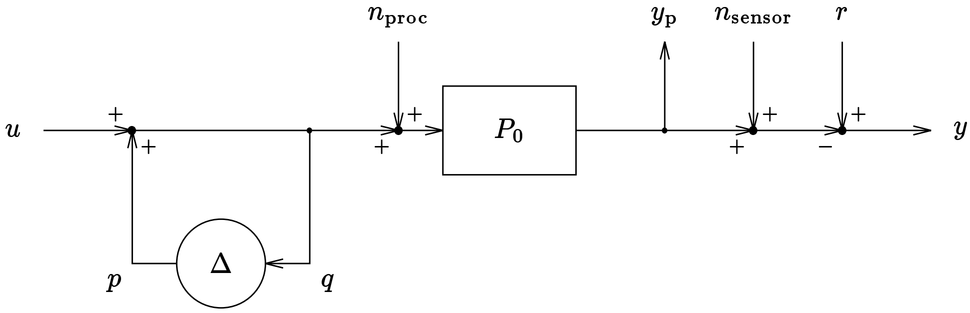


Figure 10.12 Another perturbation feedback form for the classical gain margin specification.

For this second perturbation feedback form, the transfer matrices $H_{q\tilde{w}}$, $H_{\tilde{z}p}$, and H_{qp} are given by

$$H_{q\tilde{w}} = \left[\begin{array}{ccc} -T & -T/P_0 & T/P_0 \end{array}\right], \tag{10.36}$$

$$H_{\tilde{z}p} = \begin{bmatrix} P_0 S \\ -T \end{bmatrix}, \tag{10.37}$$

$$H_{qp} = S. \tag{10.38}$$

Only H_{qp} differs from its corresponding expression for the previous perturbation feedback form.

Pole Variation

In some cases, a plant pole or zero that depends on a parameter can be expressed in perturbation feedback form. As a specific example, figure 10.13 shows one way to express the specific example described in section 10.2.2 in perturbation feedback form.

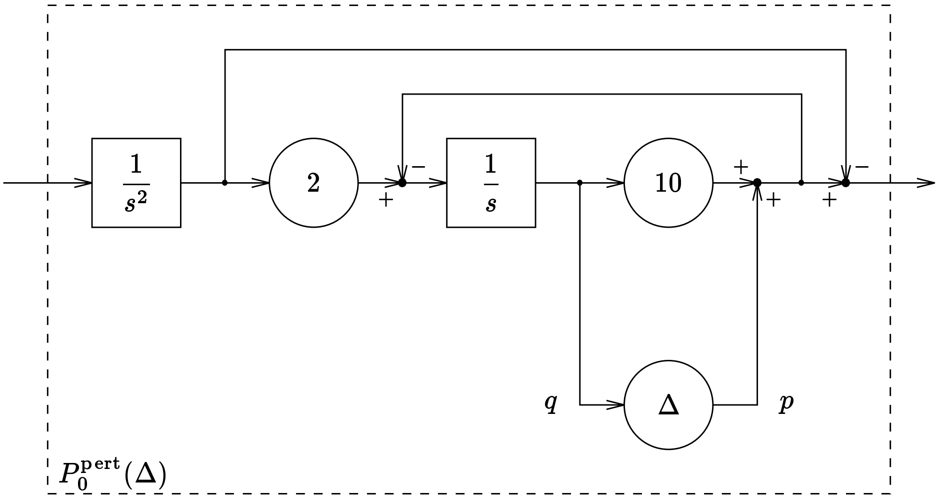


Figure 10.13 The variation in the phase shift of P_0, described by (10.6), can be represented as the effect of a varying feedback gain Δ inside the plant.

Here we have

$$\left[\begin{array}{c|c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}p} & P_{\tilde{z}u} \\ \hline P_{q\tilde{w}} & P_{qp} & P_{qu} \\ \hline P_{y\tilde{w}} & P_{yp} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c|c} \star & \star & \star & \dfrac{s}{s+10} & \star \\ \star & \star & \star & 0 & \star \\ \hline \dfrac{2}{s^2(s+10)} & 0 & 0 & \dfrac{-1}{s+10} & \dfrac{2}{s^2(s+10)} \\ \hline \star & \star & \star & \dfrac{-s}{s+10} & \star \end{array}\right] \tag{10.39}$$

with feedback perturbation set

$$\boldsymbol{\Delta} = [-5,\ 5]. \tag{10.40}$$

The closed-loop transfer matrices are given by

$$H_{q\tilde{w}} = \left[\begin{array}{ccc} \dfrac{2}{s^2(s+10)}S & \dfrac{2}{s-10}T & \dfrac{-2}{s-10}T \end{array}\right], \tag{10.41}$$

$$H_{\tilde{z}p} = \left[\begin{array}{c} \dfrac{s}{s+10}S \\ \dfrac{s^3}{s-10}T \end{array}\right], \tag{10.42}$$

$$H_{qp} = \frac{2s}{s^2-100}T - \frac{1}{s+10}. \tag{10.43}$$

(The reader worried that these transfer matrices may be unstable should recall that the interpolation conditions of section 7.2.5 require that $T(10) = 0$; similar conditions guarantee that these transfer matrices are proper, and have no pole at $s = 0$.)

Relative Uncertainty in P_0

The plant perturbation set (10.10) can be expressed in the perturbation feedback form shown in figure 10.14, for which

$$\left[\begin{array}{c|c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}p} & P_{\tilde{z}u} \\ \hline P_{q\tilde{w}} & P_{qp} & P_{qu} \\ \hline P_{y\tilde{w}} & P_{yp} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c|c} \star & \star & \star & W_{\mathrm{rel\_err}}P_0 & \star \\ \star & \star & \star & 0 & \star \\ \hline 0 & 0 & 0 & 0 & 1 \\ \hline \star & \star & \star & -W_{\mathrm{rel\_err}}P_0 & \star \end{array}\right], \tag{10.44}$$

and

$$\mathbf{\Delta} = \{\Delta \mid \|\Delta\|_\infty \le 1\}. \tag{10.45}$$

In this case the feedback perturbations are normalized unknown-but-bounded transfer functions, that cause the transfer function P_0 to become $(1 + W_{\mathrm{rel\_err}}\Delta)P_0$.

For this perturbation feedback form, the transfer matrices $H_{q\tilde{w}}$, $H_{\tilde{z}p}$, and H_{qp} are given by

$$H_{q\tilde{w}} = \left[\begin{array}{ccc} -T & -T/P_0 & T/P_0 \end{array}\right], \tag{10.46}$$

$$H_{\tilde{z}p} = \left[\begin{array}{c} W_{\mathrm{rel\_err}}P_0 S \\ -W_{\mathrm{rel\_err}}T \end{array}\right], \tag{10.47}$$

$$H_{qp} = -W_{\mathrm{rel\_err}}T. \tag{10.48}$$

Saturating Actuators

We consider the perturbed plant set (10.15) which consists of the single nonlinear system $\{P^{\mathrm{nonlin}}\}$. This can be expressed in perturbation feedback form by express-

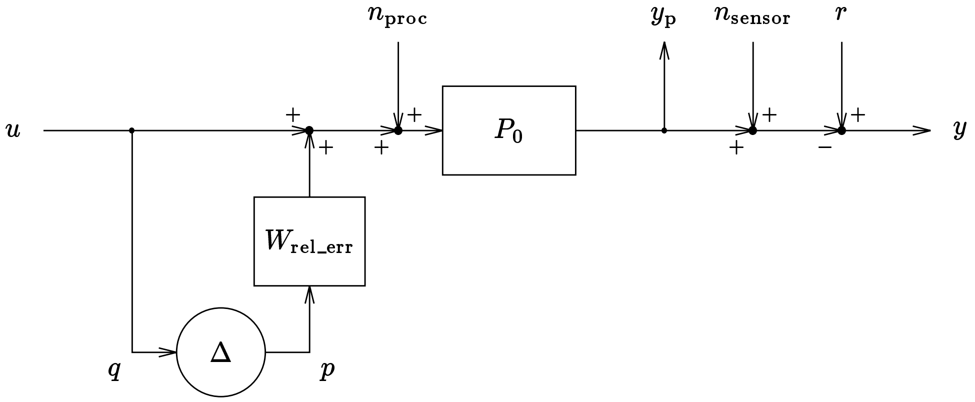


Figure 10.14 A perturbation feedback form for the plant perturbation set (10.10).

ing each saturator as a straight signal path perturbed by a *dead-zone nonlinearity*,

$$\mathbf{Dz}(a) \stackrel{\Delta}{=} a - \mathbf{Sat}(a),$$

as shown in figure 10.15.

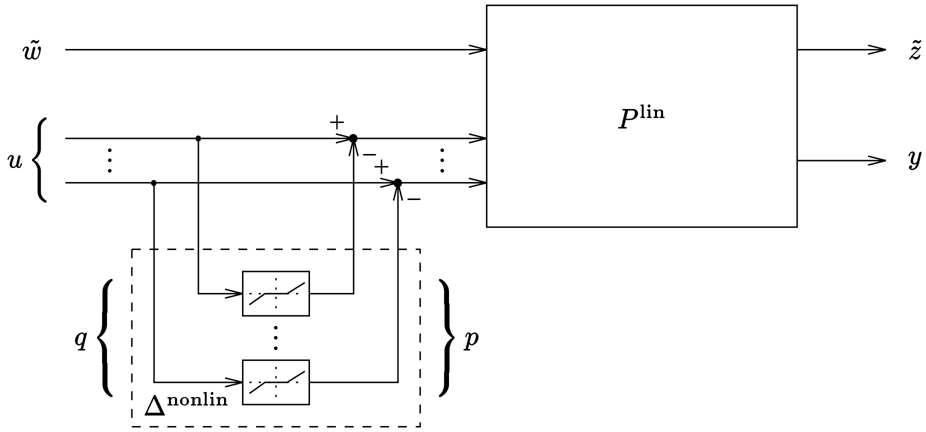


Figure 10.15 The nonlinear system shown in figure 10.5 is redrawn as the nominal plant, which is LTI, connected to Δ^{nonlin}, which is a dead-zone nonlinearity.

This perturbation feedback form is described by the augmented plant

$$\left[\begin{array}{c|c|c} P_{\tilde{z}\tilde{w}} & P_{\tilde{z}p} & P_{\tilde{z}u} \\ \hline P_{q\tilde{w}} & P_{qp} & P_{qu} \\ \hline P_{y\tilde{w}} & P_{yp} & P_{yu} \end{array}\right] = \left[\begin{array}{c|c|c} P^{\text{lin}}_{zw} & -P^{\text{lin}}_{zu} & P^{\text{lin}}_{zu} \\ \hline 0 & 0 & I \\ \hline P^{\text{lin}}_{yw} & -P^{\text{lin}}_{yu} & P^{\text{lin}}_{yu} \end{array}\right], \tag{10.49}$$

and $\mathbf{\Delta} = \{\Delta^{\mathrm{nonlin}}\}$, where $p = \Delta^{\mathrm{nonlin}}(q)$ is defined by

$$p_i(t) = S_i\, \mathbf{Dz}(q_i(t)/S_i), \quad i = 1, \ldots, n_u. \tag{10.50}$$

In this case the feedback perturbation Δ^{nonlin} is a memoryless nonlinearity.

For this perturbation feedback form, the transfer matrices $H_{q\tilde{w}}$, $H_{\tilde{z}p}$, and H_{qp} are given by

$$H_{q\tilde{w}} = K(I - P_{yu}^{\mathrm{lin}}K)^{-1}P_{yw}^{\mathrm{lin}}, \tag{10.51}$$

$$H_{\tilde{z}p} = -P_{zu}^{\mathrm{lin}}(I - KP_{yu}^{\mathrm{lin}})^{-1}, \tag{10.52}$$

$$H_{qp} = -KP_{yu}^{\mathrm{lin}}(I - KP_{yu}^{\mathrm{lin}})^{-1}. \tag{10.53}$$

10.4 Small Gain Method for Robust Stability

10.4.1 A Convex Inner Approximation

We consider a perturbed plant set $\mathcal{P}$ that is given by a perturbation feedback form. Suppose that the norm $\|\cdot\|_{\mathrm{gn}}$ is a gain (see chapter 5). Let M denote the maximum gain of the possible feedback perturbations, *i.e.*,

$$M = \sup_{\Delta \in \mathbf{\Delta}} \|\Delta\|_{\mathrm{gn}}. \tag{10.54}$$

M is thus a measure of how “big” the feedback perturbations can be.

Then from the small gain theorem described in section 5.4.2 (equations (5.29–5.31) with $H_1 = \Delta$ and $H_2 = H_{qp}$) we know that if

$$\|H_{qp}\|_{\mathrm{gn}} M < 1 \tag{10.55}$$

then we have for all $\Delta \in \mathbf{\Delta}$,

$$\left\|\Delta(I - H_{qp}\Delta)^{-1}\right\|_{\mathrm{gn}} \le \frac{M}{1 - M\|H_{qp}\|_{\mathrm{gn}}}.$$

From (10.23) we therefore have

$$\left\|H^{\mathrm{pert}}(\Delta) - H_{\tilde{z}\tilde{w}}\right\|_{\mathrm{gn}} \le \frac{M\|H_{\tilde{z}p}\|_{\mathrm{gn}}\|H_{q\tilde{w}}\|_{\mathrm{gn}}}{1 - M\|H_{qp}\|_{\mathrm{gn}}} \quad \text{for all } \Delta \in \mathbf{\Delta}. \tag{10.56}$$

We will refer to the closed-loop convex specification (10.55) as the *small gain condition* (for the perturbation feedback form and gain used). (10.55) and (10.56) are a precise statement of the idea expressed in section 10.3.1: the closed-loop system will be robust if the three closed-loop transfer matrices $H_{\tilde{z}p}$, $H_{q\tilde{w}}$, and H_{qp} are “small enough”.

It follows that the closed-loop convex specification on H given by

$$\|H_{qp}\|_{\rm gn} < 1/M, \tag{10.57}$$

$$\|H_{\tilde{z}p}\|_{\rm gn} < \infty, \tag{10.58}$$

$$\|H_{q\tilde{w}}\|_{\rm gn} < \infty, \tag{10.59}$$

$$\|H_{\tilde{z}\tilde{w}}\|_{\rm gn} < \infty, \tag{10.60}$$

implies that

$$\|H^{\rm pert}\|_{\rm gn} < \infty \qquad \text{for all} \quad \Delta \in \mathbf{\Delta}, \tag{10.61}$$

i.e., the robustness specification formed from the perturbed plant set $\mathcal{P}$ and the specification $\|H\|_{\rm gn} < \infty$ holds. If the gain $\|\cdot\|_{\rm gn}$ is finite only for stable transfer matrices, then the specification (10.57–10.60) implies that $H^{\rm pert}$ is stable, and thus the specification (10.57–10.60) is stronger than the specification of robust stability. In this case, we may think of the specification (10.57–10.60) as a closed-loop convex specification that guarantees robust stability.

As a more specific example, the RMS gain ($\mathbf{H}_\infty$ norm) $\|\cdot\|_\infty$ is finite only for stable transfer matrices, so the specification $\|H_{qp}\|_\infty < 1/M$, along with stability of $H_{\tilde{z}p}$, $H_{q\tilde{w}}$, and $H_{\tilde{z}\tilde{w}}$ (which is usually implied by internal stability), guarantees that the robust stability specification $\mathcal{D}_{\rm rob\_stab}$ holds for H.

The specification (10.57–10.60) can be used to form a convex inner approximation of a robust generalized stability specification, for various generalizations of internal stability (see section 7.5), by using other gains. As an example, consider the a-shifted $\mathbf{H}_\infty$ norm, which is finite if and only if the poles of its argument have real parts less than $-a$. If the specification (10.57–10.60) holds for this norm, then we may conclude that the feedback perturbations cannot cause the poles of the closed-loop system to have real parts equal to or exceeding $-a$. (We comment that changing the gain used will generally change M, and will give a different specification.)

Since the small gain condition (10.55) depends only on M, the largest gain of the feedback perturbations, it follows that the conclusion (10.56) actually holds for a set of feedback perturbations that may be larger than $\mathbf{\Delta}$:

$$\mathbf{\Delta}^{\rm sgt} = \{\Delta \mid \|\Delta\|_{\rm gn} \le M\} \supseteq \mathbf{\Delta}.$$

By using different perturbation feedback forms of a perturbed plant set, and different gains that are finite only for stable transfer matrices, the small gain condition (10.55) can be used to form different convex inner approximations of the robust stability specification.

10.4.2 An Extrapolated First Order Bound

It is interesting to compare (10.56) to a corresponding bound that is based on a first order differential analysis. Since

$$\Delta(I - H_{qp}\Delta)^{-1} \simeq \Delta$$

(recall that $\simeq$ means equals, to first order in Δ), the first order variation in the closed-loop transfer matrix is given by

$$H^{\mathrm{pert}}(\Delta) - H_{\tilde{z}\tilde{w}} \simeq H_{\tilde{z}p}\Delta H_{q\tilde{w}} \tag{10.62}$$

(*c.f.* the exact expression given in (10.23)). If we use this first order analysis to extrapolate the effects of any $\Delta \in \mathbf{\Delta}$, we have the approximate bound

$$\left\|H^{\mathrm{pert}}(\Delta) - H_{\tilde{z}\tilde{w}}\right\|_{\mathrm{gn}} \lesssim M\|H_{\tilde{z}p}\|_{\mathrm{gn}}\|H_{q\tilde{w}}\|_{\mathrm{gn}} \tag{10.63}$$

(*c.f.* the small gain bound given in (10.56); $\lesssim$ means that the inequality holds to first order in Δ).

The small gain bound (10.56) can be interpreted as the extrapolated first order differential bound (10.63), aggravated (increased) by a term that represents the "margin" in the small gain condition, *i.e.*

$$\frac{1}{1 - M\|H_{qp}\|_{\mathrm{gn}}}. \tag{10.64}$$

Of course, the small gain bound (10.56) is correct, whereas extrapolations from the first order bound (10.63) need not hold.

Continuing this comparison, we can interpret the term (10.64) as representing the higher order effects of the feedback perturbation Δ, since

$$\frac{1}{1 - M\|H_{qp}\|_{\mathrm{gn}}} = 1 + (M\|H_{qp}\|_{\mathrm{gn}}) + (M\|H_{qp}\|_{\mathrm{gn}})^2 + (M\|H_{qp}\|_{\mathrm{gn}})^3 + \cdots.$$

The bound (10.63), which keeps just the first term in this series, only accounts for the first order effects.

It is interesting that the transfer matrix H_{qp}, which is important in the small gain based approximation of robust stability, has no effect whatever on the first order change in H, given by (10.62). Thus H_{qp}, the feedback "seen" by the feedback perturbation, has no first order effects, but is key to robust stability.

Conversely, the transfer matrices $H_{\tilde{z}p}$ and $H_{q\tilde{w}}$, which by (10.62) determine the first order variation of H with respect to changes in Δ, have little to do with robust stability. Thus, differential sensitivity of H, which depends only on $H_{q\tilde{w}}$ and $H_{\tilde{z}p}$, and robust stability, which depends only on H_{qp}, measure different aspects of the robustness of the closed-loop system.

10.4.3 Nonlinear or Time-Varying Perturbations

A variation of the small gain method can be used to form an inner approximation of the robust stability specification, when the feedback perturbations are nonlinear or time-varying. In this case we define M by

$$M = \sup\left\{ \frac{\|\Delta w\|_{\mathrm{sig}}}{\|w\|_{\mathrm{sig}}} \;\middle|\; \Delta \in \mathbf{\Delta},\ \|w\|_{\mathrm{sig}} > 0 \right\} \tag{10.65}$$

where $\|\cdot\|_{\rm sig}$ denotes the norm on signals that determines the gain $\|\cdot\|_{\rm gn}$.

With this definition of M (which coincides with (10.54) when each $\Delta \in \mathbf{\Delta}$ is LTI), the specification (10.57–10.60) implies robust stability. In fact, a close analog of (10.56) holds: for all $\Delta \in \mathbf{\Delta}$ and exogenous inputs $\tilde{w}$, we have

$$\left\|\tilde{z}^{\rm pert} - \tilde{z}\right\|_{\rm sig} \le \alpha \|\tilde{w}\|_{\rm sig},$$

where

$$\alpha = \frac{M\|H_{\tilde{z}p}\|_{\rm gn}\|H_{q\tilde{w}}\|_{\rm gn}}{1 - M\|H_{qp}\|_{\rm gn}}.$$

Thus, we have a bound on the perturbations in the regulated variables that can be induced by the nonlinear or time-varying perturbations.

10.4.4 Examples

In this section we apply the small gain method to form inner approximations of some of the robust stability specifications that we have considered so far. In a few cases we will derive *different* convex inner approximations of the same robust stability specification, either by using different perturbation feedback forms for the same perturbed plant set, or by using different gains in the small gain theorem.

Neglected Dynamics: RMS Gain

We now consider the specification of internal stability along with the robust stability specification for the perturbation plant set (10.3), *i.e.*, $\mathcal{D}_{\rm rob\_stab} \cap \mathcal{D}_{\rm stable}$. We will use the perturbation feedback form given by (10.24–10.25). The specification that $H_{\tilde{z}\tilde{w}}$, $H_{q\tilde{w}}$, and $H_{\tilde{z}p}$ are stable is weaker than $\mathcal{D}_{\rm stable}$, so we will concentrate on the small gain condition (10.57). No matter which gain we use, M is one, since $\mathbf{\Delta}$ contains only one transfer matrix, the 2×2 identity matrix.

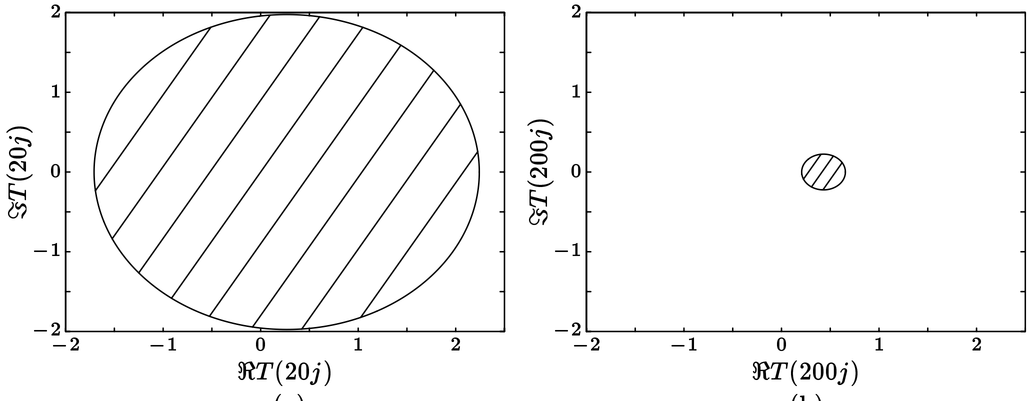
We first use the RMS gain, *i.e.*, the $\mathbf{H}_\infty$ norm $\|\cdot\|_\infty$, which is finite only for stable transfer matrices. The small gain condition is then

$$\|H_{qp}\|_\infty = \left\| \begin{bmatrix} -P_{\rm err}^{(1)}T & -TP_{\rm err}^{(2)} \\ P_{\rm err}^{(1)}S & -TP_{\rm err}^{(2)} \end{bmatrix} \right\|_\infty < 1. \qquad (10.66)$$

The closed-loop convex specification (10.66) (together with internal stability) is stronger than the robust stability specification $\mathcal{D}_{\rm rob\_stab}$ with the plant perturbation set (10.25) (together with internal stability): if (10.66) is satisfied, then the corresponding controller also stabilizes $P^{\rm cmplx}$.

We can interpret the specification (10.66) as limiting the bandwidth of the closed-loop system. The specification (10.66) can be crudely considered a frequency-dependent limit on the size of T; since $P_{\rm err}^{(1)}$ and $P_{\rm err}^{(2)}$ are each highpass filters, this limit is large for low frequencies, but less than one at high frequencies where $P_{\rm err}^{(1)}$ and

$P_{\rm err}^{(2)}$ have magnitudes nearly one, *e.g.*, $\omega \geq 100$. Thus, (10.66) requires that $|T(j\omega)|$ is less than one above about $\omega = 100$; in classical terms, the control bandwidth is less than 100rad/sec. Figure 10.16(a) shows the actual region of the complex plane that the specification (10.66) requires $T(20j)$ to lie in; figure 10.16(b) shows the same region for $T(200j)$.



(a) (b)

Figure 10.16 The specification (10.66) requires $T(j\omega)$ to lie in the shaded region (a) for $\omega = 20$, and (b) for $\omega = 200$.

Neglected Dynamics: Scaled RMS Gain

We now apply the small gain method to the same example, using the same perturbation feedback form, substituting a scaled $\mathbf{H}_\infty$ norm for the $\mathbf{H}_\infty$ norm used above (see section 5.3.6). We use the scaling

$$D = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix},$$

and the associated scaled gain,

$$\|DH_{qp}D^{-1}\|_\infty = \left\| \begin{bmatrix} -P_{\rm err}^{(1)}T & -2TP_{\rm err}^{(2)} \\ P_{\rm err}^{(1)}S/2 & -TP_{\rm err}^{(2)} \end{bmatrix} \right\|_\infty .$$

This norm is finite only for stable transfer matrices, so the small gain condition

$$\left\| \begin{bmatrix} -P_{\rm err}^{(1)}T & -2TP_{\rm err}^{(2)} \\ P_{\rm err}^{(1)}S/2 & -TP_{\rm err}^{(2)} \end{bmatrix} \right\|_\infty < 1, \qquad (10.67)$$

together with internal stability, is stronger than the robust stability specification $\mathcal{D}_{\rm rob\_stab}$ with the plant perturbation set (10.25) (together with internal stability). Like the specification (10.66), the specification (10.67) can be thought of as limiting the bandwidth of the closed-loop system.

Figure 10.17(a) shows the actual region of the complex plane that the specification (10.67) requires $T(20j)$ to lie in; figure 10.17(b) shows the same region for $T(200j)$; comparing these figures to figures 10.16(a) and 10.16(b), we see that the two inner approximations of robust stability given by (10.66) and (10.67) are different: neither is stronger than the other.

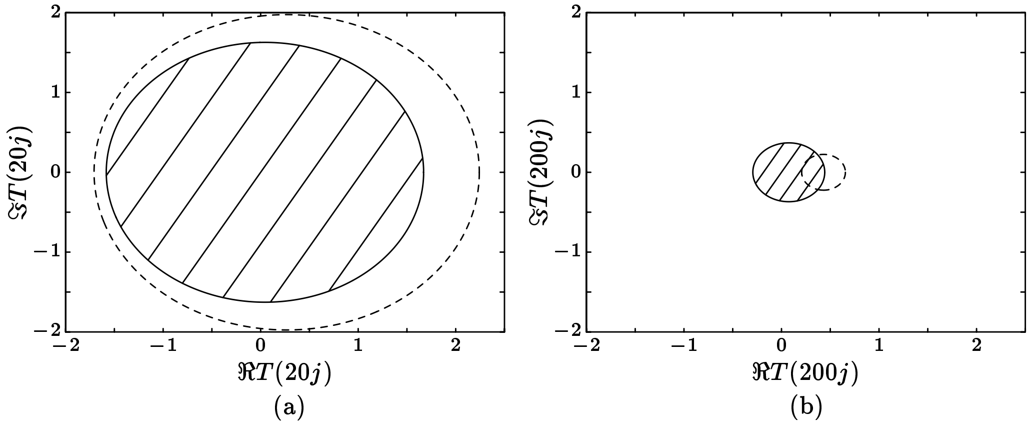


Figure 10.17 The specification (10.67) requires $T(j\omega)$ to lie in the shaded region (a) for $\omega = 20$, and (b) for $\omega = 200$. The boundaries of the regions from figure 10.16, for the specification (10.66), are shown with a dashed line.

Gain Margin: Perturbation Feedback Form 1

We now consider the gain margin specification, using the RMS gain in the small gain theorem, with the perturbation feedback form given by (10.29–10.30). The maximum of the RMS gains of the perturbations is

$$M = \max\{\|L-1\|_\infty,\ \|U-1\|_\infty\} = \max\{1-L,\ U-1\}.$$

For this perturbation feedback form, $H_{qp} = -T$, so the small gain condition is

$$\|T\|_\infty < 1/M = \min\left\{\frac{1}{1-L},\ \frac{1}{U-1}\right\}.$$

For the specific gain margins of +4dB and −3.5dB, *i.e.*, the robustness specification (10.5), the convex inner approximation is

$$\|T\|_\infty < 1.71. \tag{10.68}$$

The closed-loop convex specification (10.68) is stronger than the gain margin specification (10.5).

Gain Margin: Perturbation Feedback Form 2

We now use the second perturbation feedback form for the gain margin problem, given by (10.34–10.35). For this perturbation feedback form, we have

$$M = \max\{1/L - 1,\ 1 - 1/U\},$$

and $H_{qp} = S$, so the small gain condition is

$$\|S\|_\infty < 1/M = \min\left\{\frac{L}{1-L},\ \frac{U}{U-1}\right\}. \tag{10.69}$$

For the specific gain margins of +4dB, −3.5dB, we have

$$\|S\|_\infty < 2.02. \tag{10.70}$$

The closed-loop convex specification (10.70) is a *different* inner approximation of the gain margin specification (10.5) than (10.68).

Thus we have

$$\|T\|_\infty \le 1.71 \Longrightarrow \mathcal{D}_{+4,-3.5\mathrm{db\_gm}}, \qquad \|S\|_\infty \le 2.02 \Longrightarrow \mathcal{D}_{+4,-3.5\mathrm{db\_gm}},$$

but neither of the convex specifications on the left-hand sides is stronger than the other. We can contrast the two specifications by expressing (10.70) as $\|1-T\|_\infty < 2.02$: see figure 10.18.

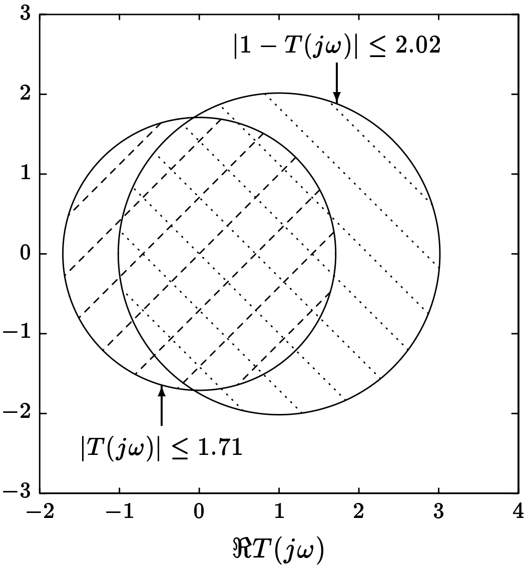


Figure 10.18 The specifications (10.68) and (10.70) require the complex number $T(j\omega)$ to lie in the indicated circles at each frequency ω.

A Generalized Gain Margin

We consider again the perturbed plant set for the gain margin specification, but tighten our robustness specification to require that the perturbed closed-loop system should have a stability degree that exceeds $a > 0$, *i.e.*, the poles of H^{pert} should have real parts less than $-a$. To form a convex inner approximation of this robustness specification, we apply the small gain method to the perturbation feedback form given by (10.34–10.35) and the a-shifted $\mathbf{H}_\infty$ norm. For this norm (indeed, for any gain) we find that

$$M = \max\{1/L - 1,\ 1 - 1/U\},$$

just as for the unshifted $\mathbf{H}_\infty$ norm.

The convex inner approximation is then: $H_{\tilde{z}\tilde{w}}$, $H_{\tilde{z}p}$, and $H_{q\tilde{w}}$ (given in (10.36–10.38)) have stability degrees exceeding a (*i.e.*, finite $\|\cdot\|_{\infty,a}$ norms) and

$$\|S\|_{\infty,a} < \min\left\{\frac{L}{1-L},\ \frac{U}{U-1}\right\}.$$

For the specific generalized gain margin of +4dB, −3.5dB, with a minimum stability degree of 0.2, we have the convex inner approximation

$$\|S\|_{\infty,0.2} < 2.02. \tag{10.71}$$

Pole Variation

We now consider the perturbed plant set (10.6) from section 10.2.2. We will form the small gain condition using the perturbation feedback form (10.39–10.40), and the RMS gain. The maximum RMS gain of the feedback perturbations is 5, so we have the approximation

$$\left\|\frac{2s}{s^2-100}T - \frac{1}{s+10}\right\|_\infty < 1/5. \tag{10.72}$$

We can interpret this closed-loop convex specification as follows. The weighting on T is a bandpass filter, whose peak magnitude is 0.1, at $\omega = 10$. The specification (10.72), roughly speaking, constrains T for frequencies near $\omega = 10$.

Relative Uncertainty in P_0

We will use the perturbation feedback form (10.44) described above, with the RMS gain. In this case we have $M = 1$, so the small gain condition becomes the weighted $\mathbf{H}_\infty$ norm specification

$$\|W_{\mathrm{rel\_err}}T\|_\infty < 1. \tag{10.73}$$

This requires that the magnitude of T lie below the frequency-dependent limit $1/|W_{\mathrm{rel\_err}}(j\omega)|$, as shown in figure 10.19.

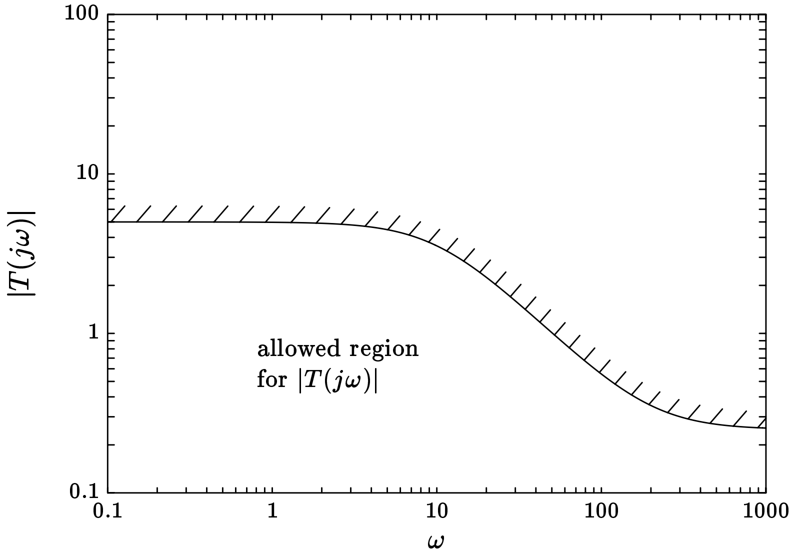


Figure 10.19 A convex inner approximation of robust stability with the relative plant uncertainty (10.7) requires that the magnitude of T lie below the frequency-dependent limit $1/|W_{\mathrm{rel\_err}}(j\omega)|$.

Saturating Actuators

We consider the perturbed plant set (10.15), with the perturbation feedback form given by (10.49) and (10.50). We will use the RMS gain or $\mathbf{H}_\infty$ norm. From formula (10.65) we find that $M = 1$, since the RMS value of the dead-zone output is less than the RMS value of its input, and for large constant signals, the two RMS values are close. The small gain condition for robust stability is thus

$$\|KP_{yu}^{\mathrm{lin}}(I - KP_{yu}^{\mathrm{lin}})^{-1}\|_\infty < 1. \tag{10.74}$$

Thus, if the closed-loop convex specification (10.74) is satisfied, then an LTI controller designed on the basis of the linear model P^{lin} will at least stabilize the nonlinear system that includes the actuator saturation.

10.5 Small Gain Method for Robust Performance

10.5.1 A Convex Inner Approximation

A variation of the small gain method can be used to form convex inner approximations of robustness specifications that involve a gain bound such as

$$\|H_{\tilde{z}\tilde{w}}\|_{\mathrm{gn}} \le \alpha,$$

where $H_{\tilde{z}\tilde{w}}$ is some entry or submatrix of H (*c.f.* robust stability, which involves the gain bound $\|H_{\tilde{z}\tilde{w}}\|_\infty < \infty$).

Throughout this section, we will consider the robustness specification that is formed from the perturbed plant set $\mathcal{P}$ and the RMS gain bound specification

$$\|H_{\tilde{z}\tilde{w}}\|_\infty \leq 1. \tag{10.75}$$

We will refer to this robust performance specification as $\mathcal{D}_{\mathrm{rob\_perf}}$. We will also assume that the perturbed plant set $\mathcal{P}$ is described by a perturbation feedback form for which the maximum RMS gain of the feedback perturbations is one, *i.e.*, $M = 1$ in (10.54).

The inner approximation of $\mathcal{D}_{\mathrm{rob\_perf}}$ is

$$\left\| \begin{bmatrix} H_{\tilde{z}\tilde{w}} & H_{\tilde{z}p} \\ H_{q\tilde{w}} & H_{qp} \end{bmatrix} \right\|_\infty < 1. \tag{10.76}$$

Like the inner approximation (10.57–10.60) of the robust stability specification $\mathcal{D}_{\mathrm{rob\_stab}}$, we can interpret (10.76) as limiting the size of $H_{\tilde{z}p}$, $H_{q\tilde{w}}$, and H_{qp}.

Let us show that (10.76) implies that the specification (10.75) holds robustly, *i.e.*,

$$\|H_{\tilde{z}\tilde{w}} + H_{\tilde{z}p}\Delta\left(I - H_{qp}\Delta\right)^{-1} H_{q\tilde{w}}\|_\infty \leq 1 \text{ for all } \Delta \in \boldsymbol{\Delta}. \tag{10.77}$$

Assume that (10.76) holds, so that for any signals $\tilde{w}$ and p we have

$$\left\| \begin{bmatrix} \tilde{z} \\ q \end{bmatrix} \right\|_{\mathrm{rms}} < \left\| \begin{bmatrix} \tilde{w} \\ p \end{bmatrix} \right\|_{\mathrm{rms}}, \tag{10.78}$$

where

$$\begin{bmatrix} \tilde{z} \\ q \end{bmatrix} = \begin{bmatrix} H_{\tilde{z}\tilde{w}} & H_{\tilde{z}p} \\ H_{q\tilde{w}} & H_{qp} \end{bmatrix} \begin{bmatrix} \tilde{w} \\ p \end{bmatrix}.$$

The inequality (10.78) can be rewritten

$$\|\tilde{z}\|_{\mathrm{rms}}^2 + \|q\|_{\mathrm{rms}}^2 < \|\tilde{w}\|_{\mathrm{rms}}^2 + \|p\|_{\mathrm{rms}}^2. \tag{10.79}$$

Now assume that $p = \Delta q$, where $\Delta \in \boldsymbol{\Delta}$, so that these signals correspond to closed-loop behavior of the perturbed system, *i.e.*,

$$\tilde{z} = \left(H_{\tilde{z}\tilde{w}} + H_{\tilde{z}p}\Delta\left(I - H_{qp}\Delta\right)^{-1} H_{q\tilde{w}}\right)\tilde{w}. \tag{10.80}$$

Since $\|\Delta\|_\infty \leq 1$, we have

$$\|p\|_{\mathrm{rms}} \leq \|q\|_{\mathrm{rms}}. \tag{10.81}$$

From (10.79–10.81) we conclude that

$$\begin{aligned} \|\tilde{z}\|_{\mathrm{rms}} &= \left\| \left(H_{\tilde{z}\tilde{w}} + H_{\tilde{z}p}\Delta\left(I - H_{qp}\Delta\right)^{-1} H_{q\tilde{w}}\right)\tilde{w} \right\|_{\mathrm{rms}} \\ &\leq \|\tilde{w}\|_{\mathrm{rms}}. \end{aligned}$$

Since this holds for any $\tilde{w}$, (10.77) follows.

Doyle has interpreted the specification (10.76) as a small gain based robust stability condition (10.55) for a perturbed plant set that includes an unknown-but-bounded transfer matrix connected from $\tilde{z}$ to $\tilde{w}$. This "performance loop" is shown in figure 10.20.

If the condition (10.76) holds, then the closed-loop system in figure 10.20 is robustly stable for all $\tilde{\Delta}$ with $\|\tilde{\Delta}\|_\infty \le 1$. In particular, the closed-loop system in figure 10.20 will be robustly stable for all $\tilde{\Delta}$ of the form

$$\tilde{\Delta} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta^{\rm perf} \end{bmatrix},$$

with $\|\Delta\|_\infty \le 1$ and $\|\Delta^{\rm perf}\|_\infty \le 1$. This is equivalent to the specification (10.75) holding robustly for all Δ with $\|\Delta\|_\infty \le 1$.

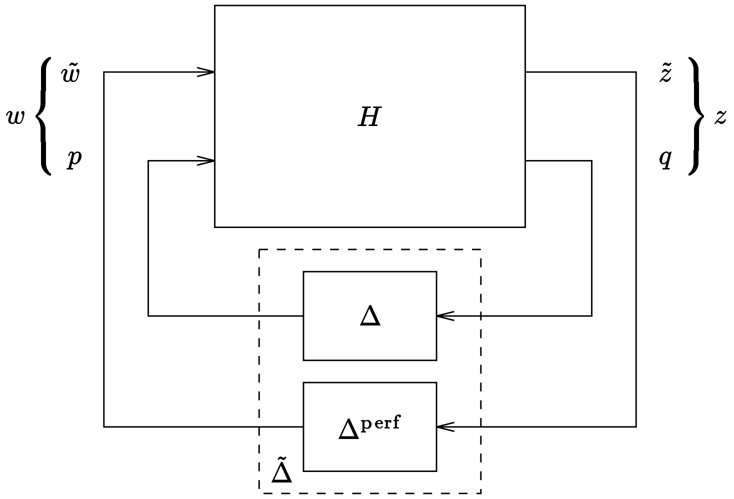


Figure 10.20 Doyle's performance loop connects a feedback $\Delta^{\rm perf}$ from the critical regulated variable $\tilde{z}$ back to the critical exogenous input $\tilde{w}$. If the resulting system is robustly stable (with $\|\Delta^{\rm perf}\|_\infty \le 1$ and $\|\Delta\|_\infty \le 1$), then the original system robustly satisfies the performance specification $\|H_{\tilde{z}\tilde{w}}\|_\infty \le 1$.

10.5.2 An Example

We consider the plant perturbation set (10.10), *i.e.*, frequency-dependent relative uncertainty in the transfer function P_0, and the robustness specification (10.11) which limits the RMS gain from the reference input to the actuator signal to be less than or equal to 75, for all possible perturbations.

Applying the method above yields the convex specification

$$\left\| \begin{bmatrix} T/(75P_0) & -W_{\mathrm{rel\_err}}T \\ T/(75P_0) & -W_{\mathrm{rel\_err}}T \end{bmatrix} \right\|_\infty < 1, \tag{10.82}$$

which guarantees that the robust performance specification holds.

The inner approximation (10.82) can be simplified by factoring T out of the matrix, and assuming T is stable, in which case it is equivalent to

$$\begin{aligned} &|T(j\omega)|\, \overline{\sigma}\left(\begin{bmatrix} 1/(75P_0(j\omega)) & -W_{\mathrm{rel\_err}}(j\omega) \\ 1/(75P_0(j\omega)) & -W_{\mathrm{rel\_err}}(j\omega) \end{bmatrix} \right) \\ &\qquad = |T(j\omega)|\, \sqrt{2\left(|W_{\mathrm{rel\_err}}(j\omega)|^2 + |1/(75P_0(j\omega))|^2\right)} \;<\; 1 \quad \text{for} \;\; \omega \in \mathbf{R}. \end{aligned}$$

Hence, (10.82) is equivalent to the specification that T be stable and satisfy the frequency-dependent magnitude limit

$$|T(j\omega)| < \frac{1}{\sqrt{2\left(|W_{\mathrm{rel\_err}}(j\omega)|^2 + |1/(75P_0(j\omega))|^2\right)}} \quad \text{for} \;\; \omega \in \mathbf{R}, \tag{10.83}$$

which is plotted in figure 10.21. The reader should compare the specification (10.83) to (10.73), which guarantees robust stability, and is also plotted in figure 10.21.

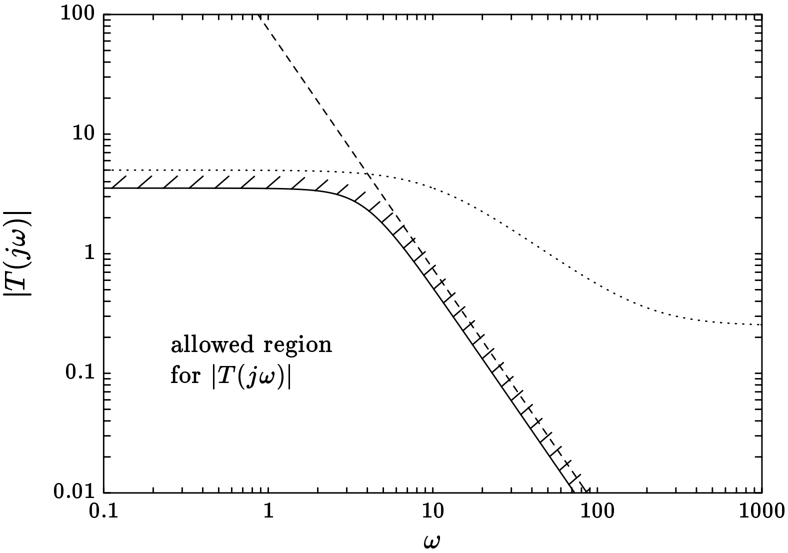


Figure 10.21 If the closed-loop transfer function T is stable and its magnitude lies below the limit shown, then the relative uncertainty in P_0 cannot cause the perturbed closed-loop transfer function from reference input to actuator signal to have RMS gain exceeding 75. The dotted limit shows the previously determined bound that guarantees closed-loop stability despite the relative uncertainty in P_0 (see figure 10.19). The dashed line is the bound imposed by the performance specification $\|T/P_0\|_\infty \leq 75$. For this example, the specification of robust stability and nominal performance is not much looser than the small gain based inner approximation of robust performance.

Notes and References

Comparison with Differential Sensitivity

In [LUN89, P48–49], Lunze states:

> Sensitivity is a local property that describes how strongly the system performance is affected by very small perturbations around a given nominal point $\hat{a}$. No information about the amount of perturbations is used. However, as many properties depend continuously on the parameter vector a, extrapolations can be made from very small to finite deviations. Therefore, sensitivity analysis yields guidelines for the attenuation of severe parameter perturbation and, hence, the achievement of robustness. But sensitivity analysis alone cannot ensure this robustness because the range of validity of the results is not known.

(See also table 3.1 in [LUN89, P49].)

The distinction between parametrized and other perturbed plant sets is not as clear as it might seem. In [BOY86], it is observed that robust stability specifications with unknown-but-bounded transfer function perturbations can be recast as robust stability specifications with parametrized plant perturbation sets. (There does not appear to be any advantage in doing so.)

Relation to Classical Control Ideas

Many of the small gain based approximations of robustness specifications that we have seen can be interpreted as limiting the magnitude of T or S. The idea that a closed-loop system with “large” T or S can be very sensitive to changes in P_0 is well-known in classical control; the specification $\|T\|_\infty \le M$ is called an M-circle constraint. Horowitz [HOR63, P148] states

> ...it is not necessarily a useful practical system, if the locus [of L] passes very close to the -1 point. In the first place, a slight change of gain or time constant may sufficiently shift the locus so as to lead to an unstable system. In the second place, the closed-loop system response has $1 + L$ for its denominator. At those frequencies for which L is close to -1, $1 + L$ is close to zero, leading to large peaking in the system frequency response.

A classical interpretation of the small gain specification $\|S\|_\infty \le \alpha$ is that the Nyquist plot maintains a distance of at least $1/\alpha$ from the critical point -1 (see [BBN90]).

For a discussion of singular perturbations of control systems, see Kokotovic, Khalil, and O'Reilly [KKO86].

Small Gain Methods

Small gain methods are discussed in the books by Desoer and Vidyasagar [DV75] and Vidyasagar [VID78]. In most discussions the method is used to establish stability despite nonlinear and time-varying perturbations.

The small gain theorem for linear Δ is a basic result of Functional Analysis, often attributed to Banach; see, *e.g.*, Kantorovich and Akilov [KA82, P154]. Its use in the analysis of feedback systems was introduced by Zames [ZAM66A].

Many papers that discuss robustness specifications and small gain approaches are reprinted in the volume edited by Dorato [DOR87]; see also the recent books by Lunze [LUN89, CH.8] and Maciejowski [MAC89, §3.10].

Conservatism of Small Gain Methods

Since the small gain method yields inner approximations of robustness specifications, it is natural to ask how "conservative" these approximations can be. Doyle, Wall, Stein, Chen, and Desoer [DWS82, DS81, CD82B] observed that for the special case when

$$\boldsymbol{\Delta} = \{\Delta \mid \|\Delta\|_\infty \le M\},$$

the small gain condition for robust stability, *i.e.*,

$$\|H_{qp}\|_\infty < 1/M,$$

is exactly equivalent to the robust stability specification, and not just an approximation. In such cases, therefore, the specification of robust stability is closed-loop convex. Thus in our example of robust stability despite relative uncertainty in P_0, the small gain condition (10.73) is the same as the robust stability specification.

In other cases the approximation can be arbitrarily conservative; see for example the papers by Doyle [DOY82, DOY78] and Safonov and Doyle [SD84]. These papers suggest various ways this conservatism can be reduced, for example, by choosing an optimally-scaled norm for the small gain theorem. (We saw in our unmodeled plant dynamics example that scaling can affect the inner approximation produced by the small gain method.) But limits on the optimally-scaled gain are not closed-loop convex.

In these papers, Doyle introduces the *structured singular value*; if the structured singular value is substituted for the norm-bound in the small gain theorem, then there is no conservatism for robust stability problems with certain types of feedback perturbation sets. Specifications that limit the structured singular value are, however, not closed-loop convex.

Small Gain Theorem and Lyapunov Stability

Many of the specifications that we have encountered in this chapter have the form of a (possibly weighted) $\mathbf{H}_\infty$ norm-bound on H_{qp}. If such a specification is satisfied, then we can compute the positive definite solution X of the ARE (5.42) as we described in section 5.6.3. This matrix provides a *Lyapunov function*, $V(z) = z^T X z$, that *proves* that robust stability holds. See [BBK89] and [WIL73].

Circle Theorem

Several of the small gain robustness specifications that we encountered can be interpreted as instances of the *circle criterion*, developed by Zames [ZAM66B], Sandberg [SAN64], and Narendra and Goldwyn [NG64]. Multivariable versions were developed by Safonov and Athans [SAF80, SA81] and others. Part III of the collection edited by MacFarlane [MAC79] contains reprints of many of these original articles.

Boyd, Barratt, and Norman [BBN90] show that a general circle criterion specification is closed-loop convex.

About the Examples

The gain margin specification has been completely analyzed by Tannenbaum in [TAN80] and [TAN82]; his analysis shows that it is not closed-loop convex; indeed, the set of transfer matrices that satisfy a gain margin specification need not be connected.

The convex inner approximation (10.71) of the generalized gain margin, which limits the shifted $\mathbf{H}_\infty$ norm of S, is a special form of a generalized circle theorem (Moore [MOO68]).

The robust performance specification example turns out to be closed-loop convex, since it can be shown to be equivalent to

$$|W_{\rm rel\_err}(j\omega)T(j\omega)| + |(1/75)T(j\omega)/P_0(j\omega)| < 1 \quad \text{for all } \omega.$$

In fact, for the 1-DOF control system, most robust performance specifications that are expressed in terms of weighted $\mathbf{H}_\infty$ norm-bounds also turn out to be closed-loop convex. Another example is described in Francis [FRA88], and discussed in Boyd, Barratt and Norman [BBN90].

Robust Performance Method

This method was introduced by Doyle in [DOY82, DOY78], and is discussed in Maciejowski [MAC89, §3.12].

Describing Function Method

The describing function method is described in [GV68] and [VID78, CH4]. Several modifications can make the describing function method nonheuristic; see, *e.g.*, Mees and Bergen [MB75].

An Extension of the Saturating Actuators Example

We saw that it may be possible to design an LTI controller for a plant that is linear except for saturating actuators, in such a way that we can guarantee that the resulting nonlinear closed-loop system is stable, by requiring the specification (10.74) to be met. In this section we briefly describe an extension of this idea that has been very useful in practice, and is interesting because the method effectively synthesizes a *nonlinear* controller.

The control system architecture is shown in figure 10.22. The nonlinear controller $K^{\rm nonlin}$ consists of a two-input, one-output LTI system, with its output saturated and fed back to its first input; the output is identical to the signal that drives $P^{\rm lin}$.

We redraw this control system as shown in figure 10.23, and consider the dead-zone nonlinearity as a perturbation. We augment our design specifications with the small gain condition $\|H_{qp}\|_\infty < 1$; any resulting design is guaranteed to yield a stable nonlinear closed-loop system. (Indeed, by our comments above, we can produce a Lyapunov function that proves stability of the closed-loop system.)

This scheme is discussed in, for example, Åström and Wittenmark [AW90], and Morari and Zafirou [MZ89, §3.2.3].

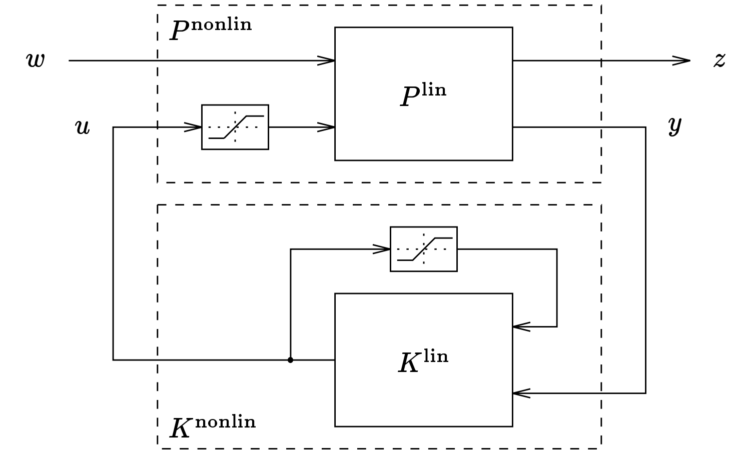


Figure 10.22 The nonlinear controller K^{nonlin} consists of a two-input, one-output LTI system, with its output saturated and fed back to its first input (this is the signal that drives P^{lin}).

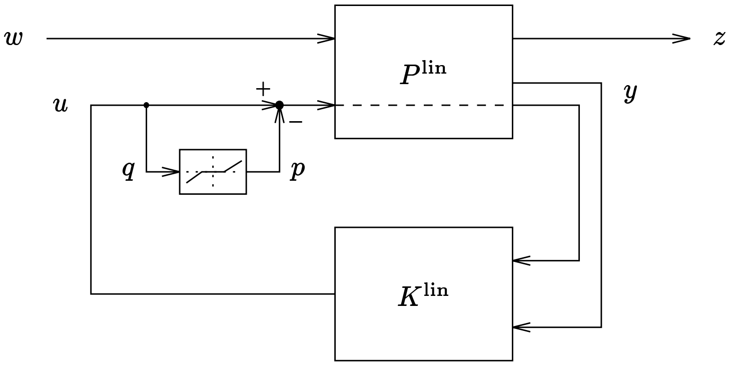


Figure 10.23 The saturators in figure 10.22 are treated as a dead-zone nonlinearity perturbation to a linear system.

Chapter 11

A Pictorial Example

The sets of transfer matrices that satisfy specifications are generally infinite-dimensional. In this chapter we consider our standard example described in section 2.4 with an additional two-dimensional affine specification. This allows us to visualize a two-dimensional "slice" through the various specifications we have encountered. The reader can directly see, for this example, that specifications we have claimed are convex are indeed convex.

Recall from section 2.4 that $H^{(\mathrm{a})}$, $H^{(\mathrm{b})}$, and $H^{(\mathrm{c})}$ are the closed-loop transfer matrices resulting from the three controllers $K^{(\mathrm{a})}$, $K^{(\mathrm{b})}$, and $K^{(\mathrm{c})}$ given there. The closed-loop affine specification

$$\mathcal{H}_{\mathrm{slice}} = \left\{ H \;\middle|\; H = \alpha H^{(\mathrm{a})} + \beta H^{(\mathrm{b})} + (1-\alpha-\beta) H^{(\mathrm{c})} \text{ for some } \alpha,\ \beta \in \mathbf{R} \right\}$$

requires H to lie on the plane passing through these three transfer matrices. The specification $\mathcal{H}_{\mathrm{slice}}$ has no practical use, but we will use it throughout this chapter to allow us to plot two-dimensional "slices" through other (useful) specifications.

Figure 11.1 shows the subset $-1 \le \alpha \le 2$, $-1 \le \beta \le 2$ of $\mathcal{H}_{\mathrm{slice}}$. Most plots that we will see in this chapter use this range. Each point in figure 11.1 corresponds to a closed-loop transfer matrix; for example, $H^{(\mathrm{a})}$ corresponds to the point $\alpha = 1$, $\beta = 0$, $H^{(\mathrm{b})}$ corresponds to the point $\alpha = 0$, $\beta = 1$, and $H^{(\mathrm{c})}$ corresponds to the point $\alpha = 0$, $\beta = 0$. Also shown in figure 11.1 are the points

$$0.6H^{(\mathrm{a})} + 0.3H^{(\mathrm{b})} + 0.1H^{(\mathrm{c})} \quad \text{and} \quad -0.2H^{(\mathrm{a})} - 0.6H^{(\mathrm{b})} + 1.8H^{(\mathrm{c})}.$$

Each point in figure 11.1 also corresponds to a particular controller, although we will not usually be concerned with the controller itself. The controller that realizes the closed-loop transfer matrix

$$\alpha H^{(\mathrm{a})} + \beta H^{(\mathrm{b})} + (1-\alpha-\beta) H^{(\mathrm{c})}$$

can be computed by two applications of equation (7.10) from chapter 7.

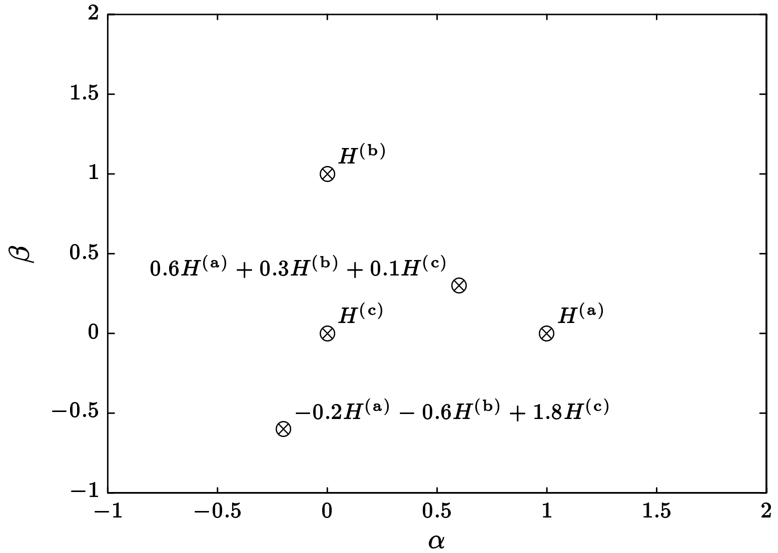


Figure 11.1 Each point $(\alpha,\ \beta)$ corresponds to a closed-loop transfer matrix H that lies in the plane through $H^{(a)}$, $H^{(b)}$, and $H^{(c)}$.

Many of the figures in this chapter show level curves of functionals on $\mathcal{H}_{\rm slice}$, *i.e.*,

$$\left\{ [\alpha \ \ \beta]^T \ \middle| \ \phi\left(\alpha H^{(a)} + \beta H^{(b)} + (1-\alpha-\beta)H^{(c)}\right) = \gamma \right\}.$$

For quasiconvex functionals this set might not be a connected curve, so we define the level curve to be the boundary of the convex set

$$\left\{ [\alpha \ \ \beta]^T \ \middle| \ \phi\left(\alpha H^{(a)} + \beta H^{(b)} + (1-\alpha-\beta)H^{(c)}\right) \le \gamma \right\}.$$

In most cases these two definitions agree.

11.1 I/O Specifications

11.1.1 A Settling Time Limit

The step responses from the reference input r to $y_{\rm p}$ for the three closed-loop systems are shown in figure 11.2. Figure 11.3 shows the level curves of the reference r to $y_{\rm p}$ settling-time functional $\phi_{\rm settle}$, *i.e.*,

$$\phi_{\rm settle}\left(\alpha H_{13}^{(a)} + \beta H_{13}^{(b)} + (1-\alpha-\beta)H_{13}^{(c)}\right). \qquad (11.1)$$

Recall from section 8.1.1 that $\phi_{\rm settle}$, and therefore (11.1), are quasiconvex. From figure 11.3 it can be seen that the level curves bound convex subsets.

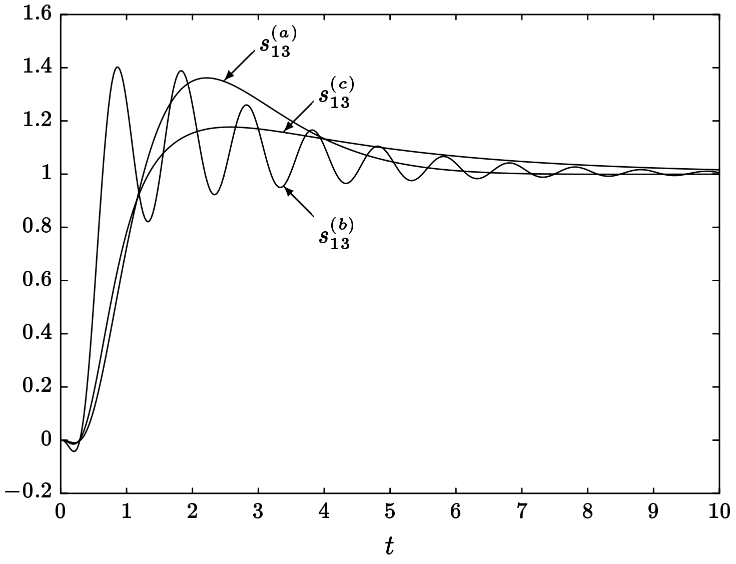


Figure 11.2 The step responses from the reference input, r, to plant output, y_{p}, for the closed-loop transfer matrices $H^{(\mathrm{a})}$, $H^{(\mathrm{b})}$, and $H^{(\mathrm{c})}$.

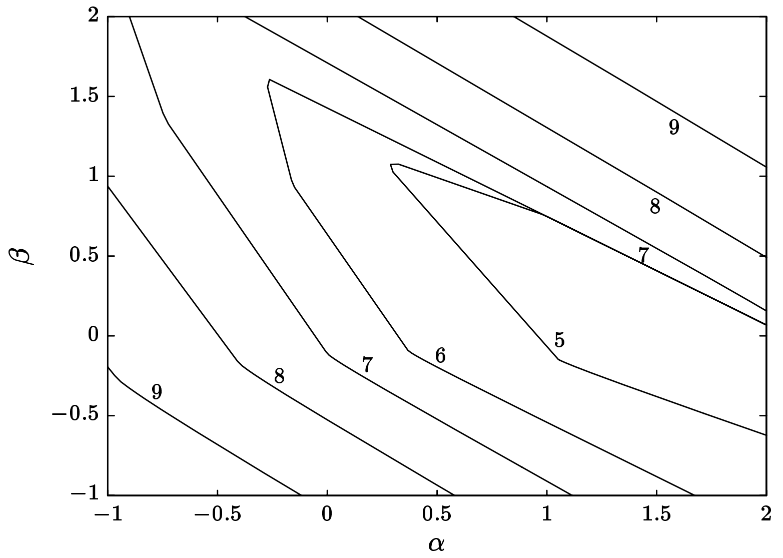


Figure 11.3 Level curves of the step response settling time, from the reference r to y_{p}, given by (11.1).

11.1.2 Some Worst Case Tracking Error Specifications

We now consider the tracking error. For our example, the tracking error transfer function is given by

$$H_{13} - 1$$

(we have not set up our example with the tracking error explicitly included in the regulated variables). Figure 11.4 shows the level curves of the weighted peak tracking error, *i.e.*,

$$\varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta) \triangleq \left\| W\left(\alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})} - 1 \right) \right\|_{\mathrm{pk\_gn}}, \quad (11.2)$$

where the weight is

$$W(s) = \frac{1}{2s+1}.$$

(We use the symbol φ to denote the restriction of a functional to a finite-dimensional domain.) Recall from section 8.1.2 that the weighted peak tracking error is a convex functional of H, and therefore $\varphi_{\mathrm{pk\_trk}}$ is a convex function of α and β. From figure 11.4 it can be seen that the level curves bound convex subsets.

Figure 11.5 shows the level curves of the peak tracking error, for reference inputs bounded and slew-rate limited by 1, *i.e.*,

$$\left\| \alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})} - 1 \right\|_{\mathrm{wc}}. \quad (11.3)$$

In section 6.3.2 we showed that a function of the form (11.3) is convex; as expected, the level curves in figure 11.5 bound convex subsets of $\mathcal{H}_{\mathrm{slice}}$.

In section 5.5.2 we showed that, for any transfer function H,

$$\|HW\|_{\mathrm{pk\_gn}} \le \|H\|_{\mathrm{wc}} \le 3\|HW\|_{\mathrm{pk\_gn}}.$$

The reader should compare the level curves in figures 11.4 and 11.5 with this relation in mind.

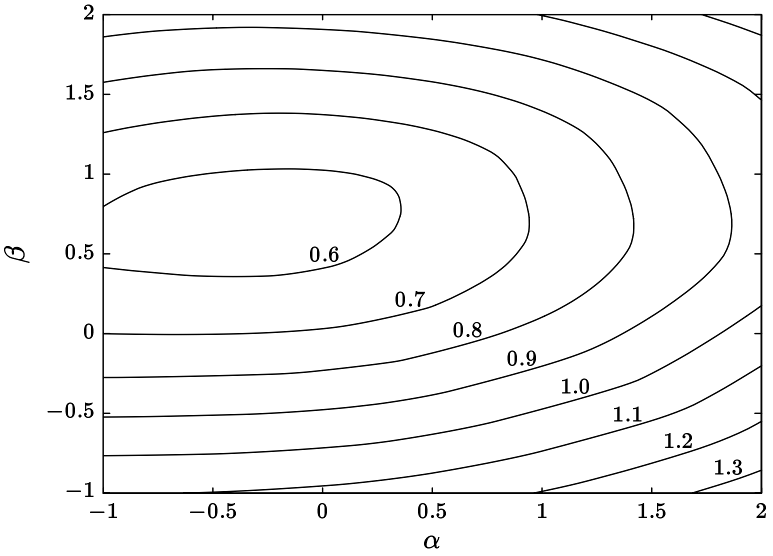


Figure 11.4 Level curves of the weighted peak tracking error, given by (11.2).

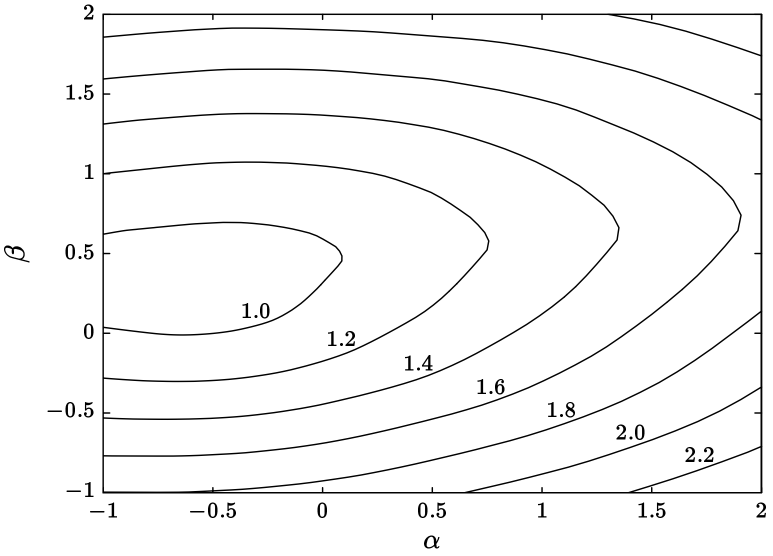


Figure 11.5 Level curves of the peak tracking error, for reference inputs bounded and slew-rate limited by 1, given by (11.3).

11.2 Regulation

11.2.1 Asymptotic Rejection of Constant Disturbances

We first consider the effect on y_{p} of a constant disturbance applied to n_{proc}. Figure 11.6 shows the subset of $\mathcal{H}_{\mathrm{slice}}$ where such a disturbance asymptotically has no effect on y_{p}, *i.e.*, where the affine function

$$\alpha H_{11}^{(\mathrm{a})}(0) + \beta H_{11}^{(\mathrm{b})}(0) + (1-\alpha-\beta) H_{11}^{(\mathrm{c})}(0) \tag{11.4}$$

vanishes.

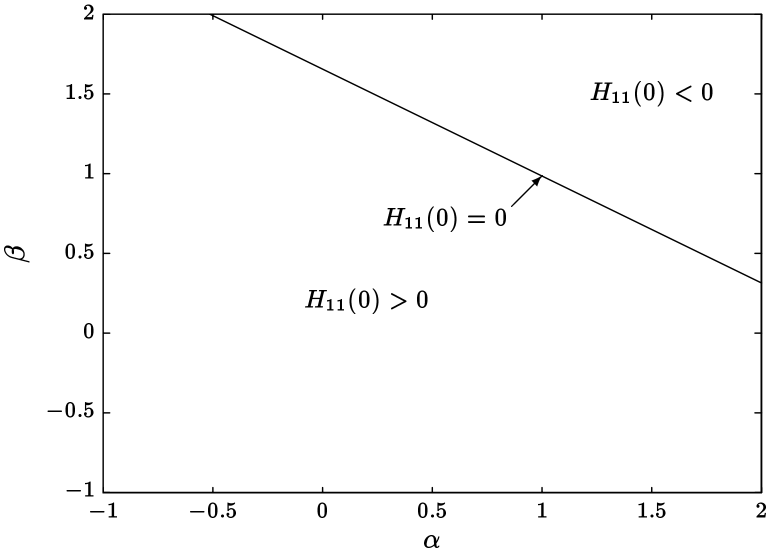


Figure 11.6 Asymptotic rejection of constant actuator-referred disturbances on y_{p}.

11.2.2 Rejection of a Particular Disturbance

Suppose now that an actuator-referred disturbance is the waveform $d_{\mathrm{part}}(t)$ shown in figure 11.7. Figure 11.8 shows the level curves of the peak output y_{p} due to the actuator-referred disturbance d_{part}, *i.e.*,

$$\left\| \left(\alpha H_{11}^{(\mathrm{a})} + \beta H_{11}^{(\mathrm{b})} + (1-\alpha-\beta) H_{11}^{(\mathrm{c})} \right) d_{\mathrm{part}} \right\|_{\infty}, \tag{11.5}$$

which is a convex function on $\mathbf{R}^2$.

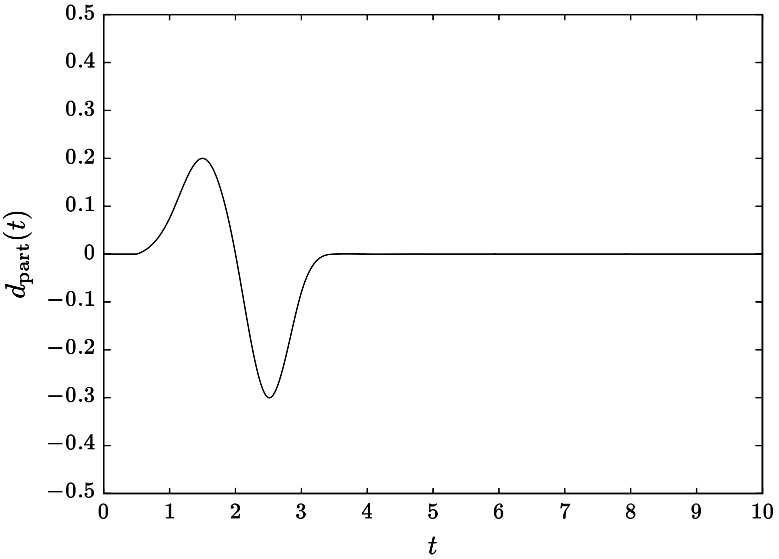


Figure 11.7 A particular actuator-referred process disturbance signal, $d_{\mathrm{part}}(t)$.

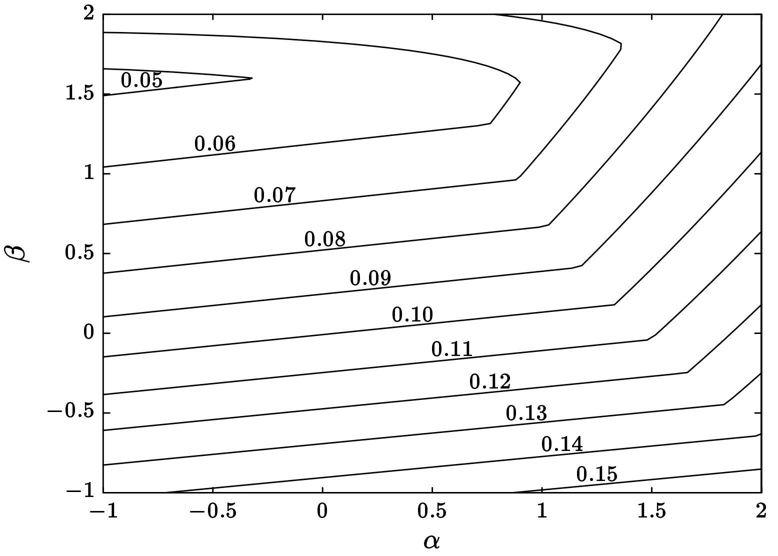


Figure 11.8 Level curves of the peak of y_{p} due to the particular actuator-referred disturbance $d_{\mathrm{part}}(t)$ shown in figure 11.7, given by (11.5).

11.2.3 RMS Regulation

Suppose that $n_{\rm proc}$ and $n_{\rm sensor}$ are independent, zero-mean stochastic processes with power spectral densities

$$\begin{aligned} S_{\rm proc}(\omega) &= W_{\rm proc}^2, \\ S_{\rm sensor}(\omega) &= W_{\rm sensor}^2, \end{aligned}$$

where

$$\begin{aligned} W_{\rm proc} &= 0.04, \\ W_{\rm sensor} &= 0.01 \end{aligned}$$

(*i.e.*, scaled white noises). Figure 11.9 shows the level curves of the RMS value of $y_{\rm p}$ due to these noises, *i.e.*, the level curves of the function

$$\varphi_{\rm rms\_yp}(\alpha,\ \beta) \stackrel{\Delta}{=} \phi_{\rm rms\_yp}\left(\alpha H^{(\rm a)} + \beta H^{(\rm b)} + (1-\alpha-\beta)H^{(\rm c)}\right), \tag{11.6}$$

where

$$\phi_{\rm rms\_yp}(H) \stackrel{\Delta}{=} \left(\|H_{11}W_{\rm proc}\|_2^2 + \|H_{12}W_{\rm sensor}\|_2^2\right)^{1/2}. \tag{11.7}$$

Recall from section 8.2.2 that the RMS response to independent stochastic inputs with known power spectral densities is a convex functional of H; therefore $\phi_{\rm rms\_yp}$ is a convex function of H, and $\varphi_{\rm rms\_yp}$ is a convex function of α and β.

11.3 Actuator Effort

11.3.1 A Particular Disturbance

We consider again the particular actuator-referred disturbance $d_{\rm part}(t)$ shown in figure 11.7. Figure 11.10 shows the peak actuator signal u due to the actuator-referred disturbance $d_{\rm part}$, *i.e.*,

$$\left\| \left(\alpha H_{21}^{(\rm a)} + \beta H_{21}^{(\rm b)} + (1-\alpha-\beta)H_{21}^{(\rm c)}\right) d_{\rm part} \right\|_\infty, \tag{11.8}$$

which is a convex function on $\mathbf{R}^2$.

11.3.2 RMS Limit

Figure 11.11 shows the level curves of the RMS value of u due to the noises described in section 11.2.3, *i.e.*, the level curves of the function

$$\phi_{\rm rms\_u}\left(\alpha H^{(\rm a)} + \beta H^{(\rm b)} + (1-\alpha-\beta)H^{(\rm c)}\right), \tag{11.9}$$

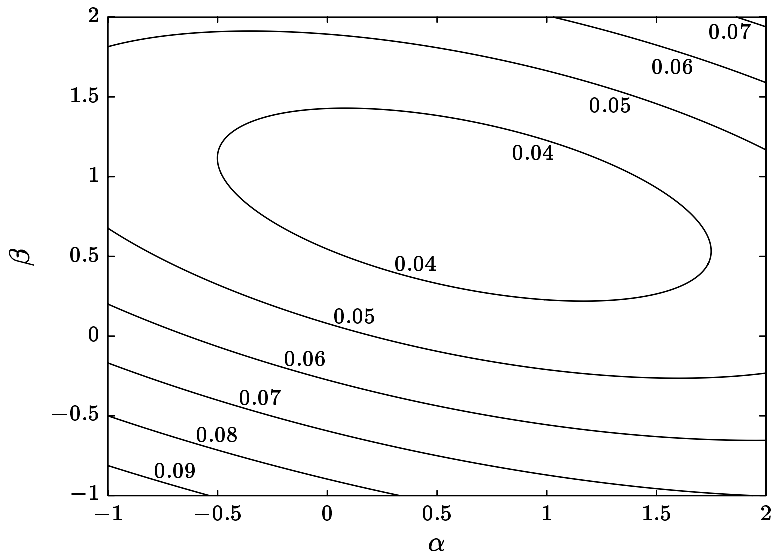


Figure 11.9 Level curves of the RMS value of y_p, with sensor and actuator noises, given by (11.6).

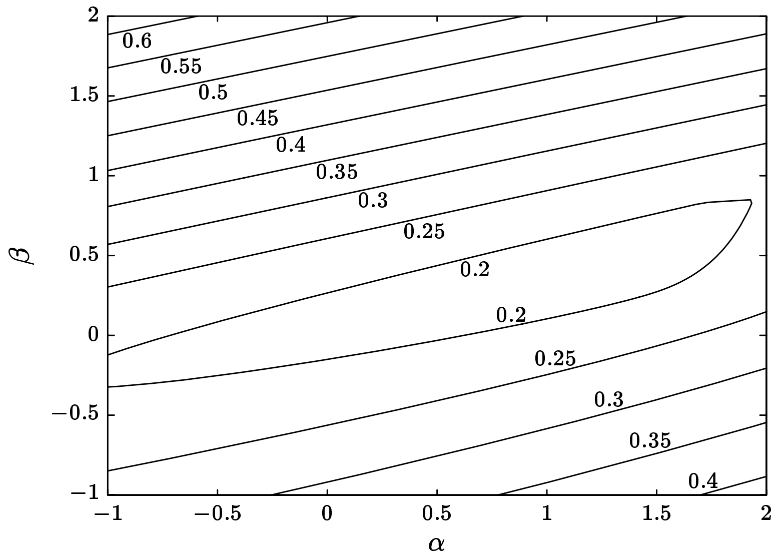


Figure 11.10 Level curves of the peak actuator signal u, due to the particular actuator-referred disturbance $d_\mathrm{part}(t)$ shown in figure 11.7, given by (11.8).

where

$$\phi_{\mathrm{rms\_u}}(H) \triangleq \left(\|H_{21}W_{\mathrm{proc}}\|_2^2 + \|H_{22}W_{\mathrm{sensor}}\|_2^2 \right)^{1/2}. \qquad (11.10)$$

Recall from section 8.2.2 that the RMS response to independent stochastic inputs with known power spectral densities is a convex functional of H; therefore $\phi_{\mathrm{rms\_u}}$ is a convex function of H, and (11.9) is a convex function of α and β.

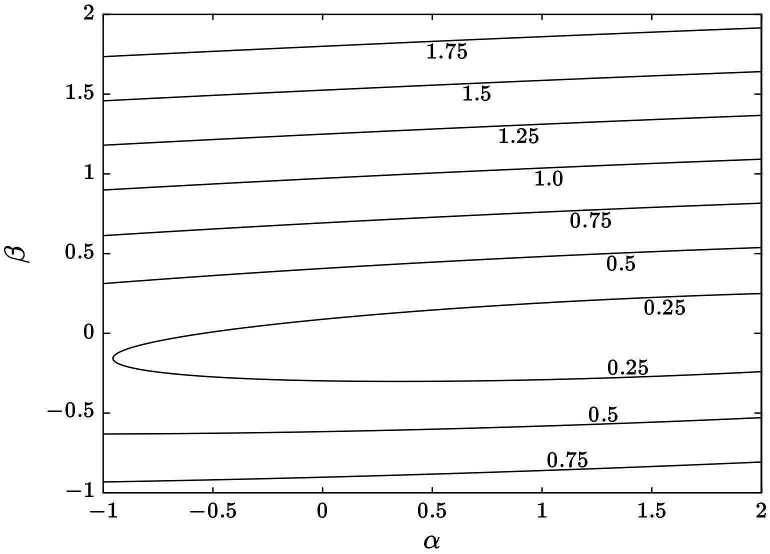


Figure 11.11 Level curves of the RMS value of the actuator signal u, with sensor and actuator noises, given by (11.9).

11.3.3 RMS Gain Limit

Figure 11.12 shows the level curves of the worst case RMS actuator signal u for any reference input r with RMS value bounded by 1, *i.e.*,

$$\left\| \alpha H_{23}^{(\mathrm{a})} + \beta H_{23}^{(\mathrm{b})} + (1-\alpha-\beta) H_{23}^{(\mathrm{c})} \right\|_\infty . \qquad (11.11)$$

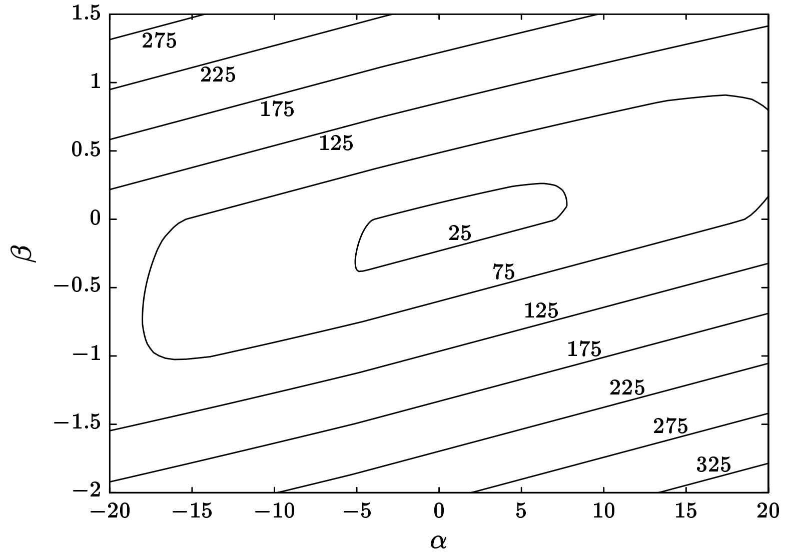


Figure 11.12 Level curves of the worst case RMS actuator signal u for any reference input r with RMS value bounded by 1, given by (11.11).

11.4 Sensitivity Specifications

11.4.1 A Log Sensitivity Specification

We consider the plant perturbation

$$\delta P_0^{\mathrm{std}}(s) = \gamma P_0^{\mathrm{std}}(s),$$

i.e., a gain variation in P_0^{std} (see section 9.1.3). Figure 11.13 shows the level curves of the maximum logarithmic sensitivity of the magnitude of the I/O transfer function H_{13}, over the frequency range $0 \le \omega \le 1$, to these gain changes, *i.e.*,

$$\sup_{0\le\omega\le1} \left| \left. \frac{\partial}{\partial\gamma} \right|_{\gamma=0} \log |H_{13}(j\omega)| \right| = \max_{0\le\omega\le1} |\Re S(j\omega)|, \tag{11.12}$$

where

$$S(j\omega) = 1 - \left(\alpha H_{13}^{(\mathrm{a})}(j\omega) + \beta H_{13}^{(\mathrm{b})}(j\omega) + (1-\alpha-\beta) H_{13}^{(\mathrm{c})}(j\omega) \right).$$

As expected, the level curves in figure 11.13 bound convex subsets of $\mathcal{H}_{\mathrm{slice}}$.

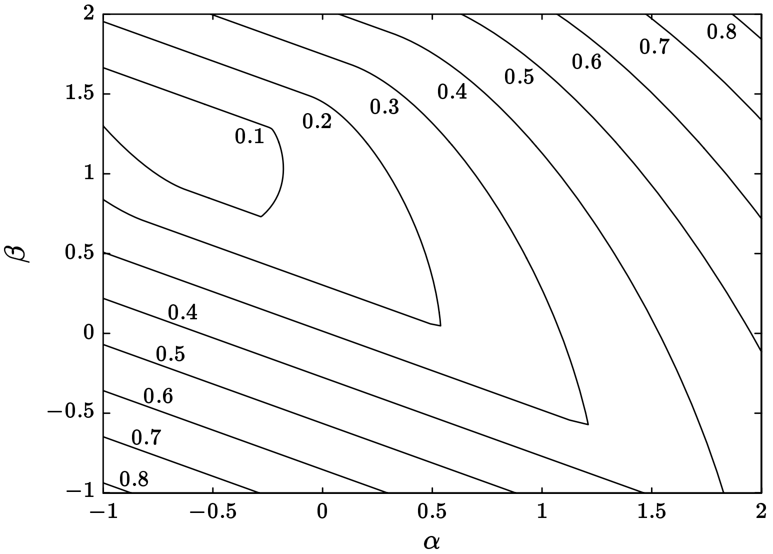


Figure 11.13 Level curves of the logarithmic sensitivity of the magnitude of the I/O transfer function H_{13}, over the frequency range $0 \le \omega \le 1$, to gain changes in the plant P_0^{std}, given by (11.12).

When the function (11.12) takes on the value 0.3, the maximum first order change in $|H_{13}(j\omega)|$, over $0 \le \omega \le 1$, with a 25% plant gain change is exp(0.075), or 0.65dB. In figure 11.14 the actual maximum change in $|H_{13}(j\omega)|$ is shown for points on the 0.3 contour of the function (11.12).

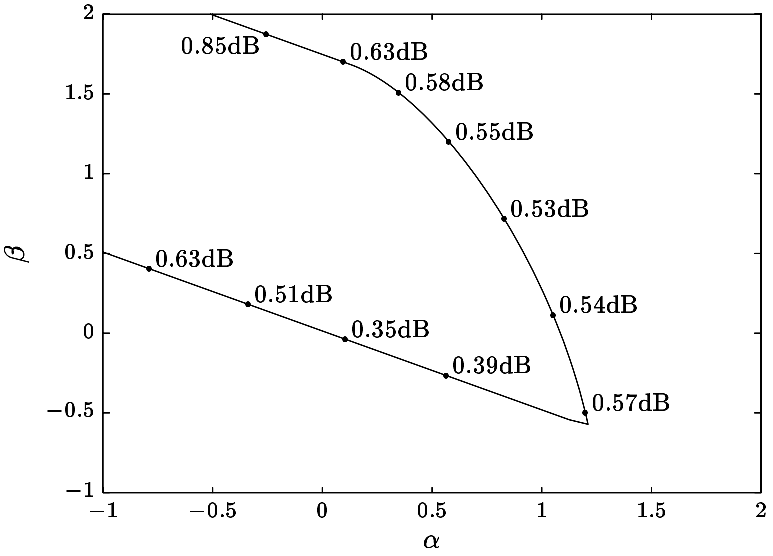


Figure 11.14 To first order, the peak change in $|H_{13}(j\omega)|$ for $0 \leq \omega \leq 1$ along the 0.3 contour in figure 11.13, for a 25% gain change in P_0^{std}, will be 0.65dB. The 0.3 contour from figure 11.13 is shown, together with the actual peak change in $|H_{13}(j\omega)|$ for $0 \leq \omega \leq 1$ for several points on the contour.

11.4.2 A Step Response Sensitivity Specification

In section 9.3 we considered the sensitivity of the I/O step response at $t = 1$ to plant gain changes, *i.e.*, $\delta P_0^{\mathrm{std}} = \gamma P_0^{\mathrm{std}}$:

$$s_\gamma(1) \triangleq \left.\frac{\partial s(1)}{\partial \gamma}\right|_{\gamma=0}.$$

Figure 11.15 shows the subset of $\mathcal{H}_{\mathrm{slice}}$ for which

$$|s_\gamma(1)| \leq 0.75.$$

This specification is equivalent to

$$\left|\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{(1-T(j\omega))T(j\omega)}{j\omega}e^{j\omega}\,d\omega\right| \leq 0.75, \tag{11.13}$$

where

$$T(j\omega) = \alpha H_{13}^{(\mathrm{a})}(j\omega) + \beta H_{13}^{(\mathrm{b})}(j\omega) + (1-\alpha-\beta)H_{13}^{(\mathrm{c})}(j\omega).$$

As we showed in section 9.3, and as is clear from figure 11.15, the step response sensitivity specification (11.13) is not convex.

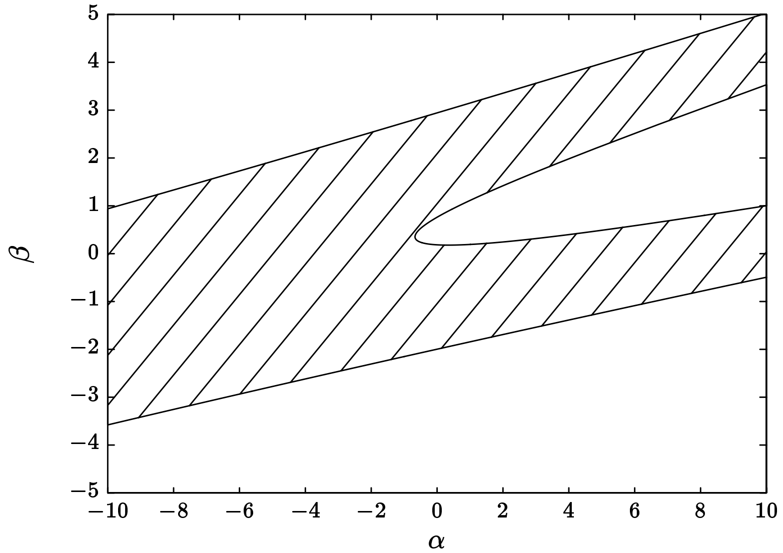


Figure 11.15 The subset of $\mathcal{H}_{\rm slice}$ that has an I/O step response sensitivity magnitude, at $t = 1$, of less than 0.75. This is the set of points for which (11.13) holds.

11.5 Robustness Specifications

11.5.1 Gain Margin

We now consider the gain margin specification $\mathcal{D}_{+4,-3.5\text{db\_gm}}$: the system should remain stable for gain changes in $P_0^{\rm std}$ between +4dB and −3.5dB. In section 10.4.4 we used the small gain theorem to show that

$$\|T\|_\infty \le 1.71 \tag{11.14}$$

$$\|S\|_\infty \le 2.02 \tag{11.15}$$

are (different) inner approximations of the gain margin specification $\mathcal{D}_{+4,-3.5\text{db\_gm}}$.

Figure 11.16 shows the subset of $\mathcal{H}_{\rm slice}$ that meets the gain margin specification $\mathcal{D}_{+4,-3.5\text{db\_gm}}$, together with the two inner approximations (11.14–11.15), *i.e.*,

$$\left\|\alpha H_{13}^{(\rm a)} + \beta H_{13}^{(\rm b)} + (1-\alpha-\beta) H_{13}^{(\rm c)}\right\|_\infty \le 1.71 \tag{11.16}$$

$$\left\|1 - \left(\alpha H_{13}^{(\rm a)} + \beta H_{13}^{(\rm b)} + (1-\alpha-\beta) H_{13}^{(\rm c)}\right)\right\|_\infty \le 2.02. \tag{11.17}$$

In general, the specification $\mathcal{D}_{+4,-3.5\text{db\_gm}}$ is not convex, even though in this case the subset of $\mathcal{H}_{\rm slice}$ that satisfies $\mathcal{D}_{+4,-3.5\text{db\_gm}}$ is convex. The two inner approximations (11.16–11.17) are norm-bounds on H, and are therefore convex (see section 6.3.2). The two approximations (11.16–11.17) are convex subsets of the exact region that satisfies $\mathcal{D}_{+4,-3.5\mathrm{db\_gm}}$.

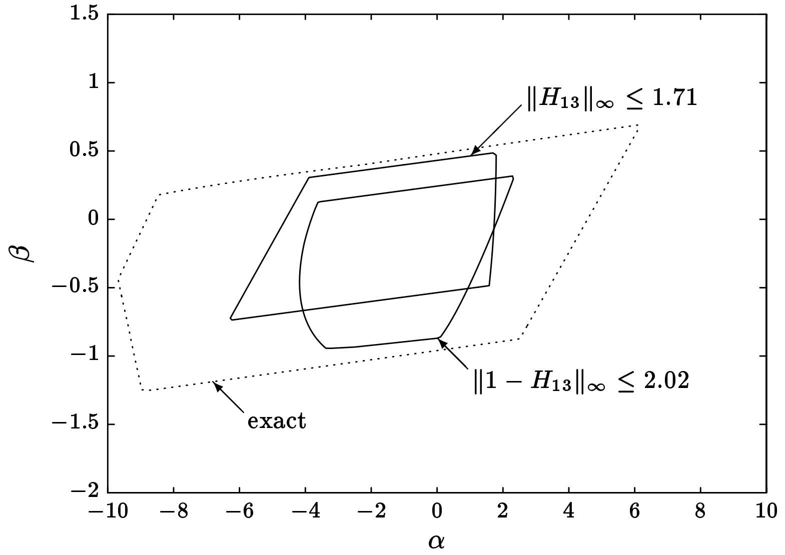


Figure 11.16 The boundary of the region where the gain margin specification $\mathcal{D}_{+4,-3.5\mathrm{db\_gm}}$ is met is shown, together with the boundaries of the two inner approximations (11.16–11.17). In this case the exact region turns out to be convex, but this is not generally so.

The bound on the sensitivity transfer function magnitude given by (10.69) is an inner approximation to each gain margin specification. Figure 11.17 shows the level curves of the peak magnitude of the sensitivity transfer function, *i.e.*,

$$\varphi_{\mathrm{max\_sens}}(\alpha,\ \beta) \stackrel{\Delta}{=} \left\| 1 - \left(\alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})} \right) \right\|_\infty . \qquad (11.18)$$

In section 6.3.2 we showed that a function of the form (11.18) is convex; the level curves in figure 11.17 do indeed bound convex subsets of $\mathcal{H}_{\mathrm{slice}}$.

11.5.2 Generalized Gain Margin

We now consider a tighter gain margin specification than $\mathcal{D}_{+4,-3.5\mathrm{db\_gm}}$: for plant gain changes between +4dB and −3.5dB the stability degree exceeds 0.2, *i.e.*, the closed-loop system poles have real parts less than −0.2. In section 10.4.4 we used the small gain theorem to show that

$$\|S\|_{\infty,0.2} \leq 2.02 \qquad (11.19)$$

is an inner approximation of this generalized gain margin specification.

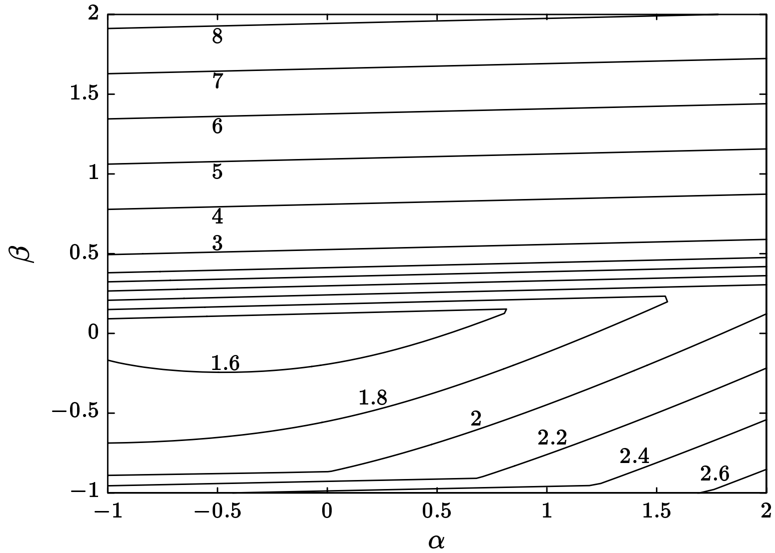


Figure 11.17 The level curves of the sensitivity transfer function magnitude, given by (11.18).

Figure 11.18 shows the subset of $\mathcal{H}_{\rm slice}$ that meets the generalized gain margin specification, together with the inner approximation (11.19), *i.e.*,

$$\left\| 1 - \left(\alpha H_{13}^{(\rm a)} + \beta H_{13}^{(\rm b)} + (1-\alpha-\beta) H_{13}^{(\rm c)} \right) \right\|_{\infty,0.2} \le 2.02. \tag{11.20}$$

The generalized gain margin specification is not in general convex, even though in this case the subset of $\mathcal{H}_{\rm slice}$ that satisfies the generalized gain margin specification is convex. The inner approximation (11.20) is a norm-bound on H, and is therefore convex (see section 6.3.2).

11.5.3 Robust Stability with Relative Plant Uncertainty

Figure 11.19 shows the subset of $\mathcal{H}_{\rm slice}$ that meets the specification

$$\mathcal{D}_{\rm rob\_stab}(\mathcal{P}), \tag{11.21}$$

where $\mathcal{P}$ is the plant perturbation set (10.10), *i.e.*, robust stability with the relative plant uncertainty $W_{\rm rel\_err}$ described in section 10.2.3. The specification (11.21) is equivalent to the convex inner approximation

$$\left\| W_{\rm rel\_err} \left(\alpha H_{13}^{(\rm a)} + \beta H_{13}^{(\rm b)} + (1-\alpha-\beta) H_{13}^{(\rm c)} \right) \right\|_{\infty} < 1 \tag{11.22}$$

derived from the small gain theorem in section 10.4.4, *i.e.*, in this case the small gain theorem is not conservative (see the Notes and References in chapter 10).

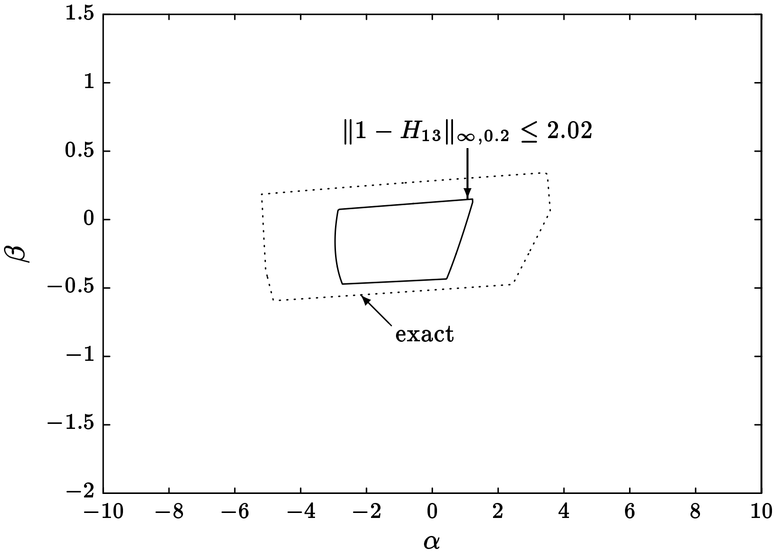


Figure 11.18 The boundary of the exact region where the generalized gain margin specification is met is shown, together with the boundary of the inner approximation (11.20). This specification is tighter than the specification $\mathcal{D}_{+4,-3.5\text{db\_gm}}$ shown in figure 11.16.

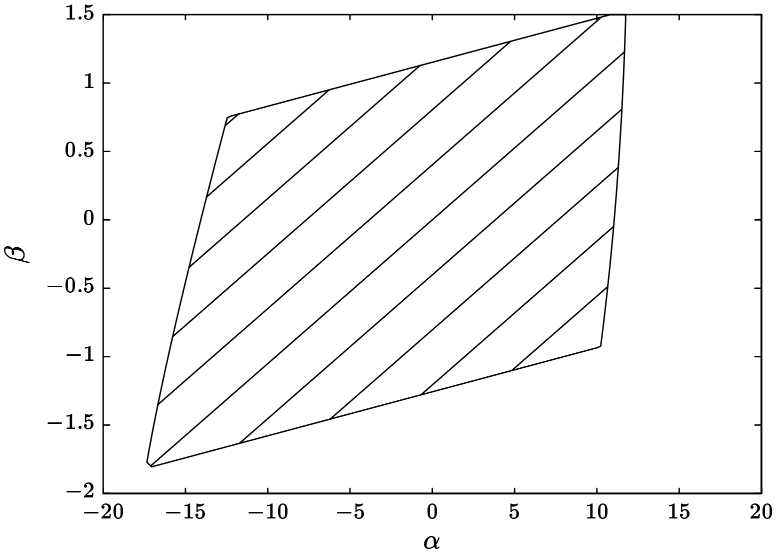


Figure 11.19 The region where the robust stability specification (11.21) is met is shaded. This region is the same as the inner approximation (11.22).

11.5.4 Robust Performance

Figure 11.20 shows the subset of $\mathcal{H}_{\text{slice}}$ that meets the specification

$$\mathcal{D}_{\text{rob}}(\mathcal{P}, \|H_{23}\|_\infty \le 75), \tag{11.23}$$

where $\mathcal{P}$ is the plant perturbation set (10.10), *i.e.*, the plant perturbations described in section 10.2.3 never cause the RMS gain from the reference input to the actuator signal to exceed 75. In section 10.5.2 we showed that an inner approximation of the specification (11.23) is

$$\left|\alpha H_{13}^{(\mathrm{a})}(j\omega) + \beta H_{13}^{(\mathrm{b})}(j\omega) + (1-\alpha-\beta) H_{13}^{(\mathrm{c})}(j\omega)\right| < l(\omega) \quad \text{for } \omega \in \mathbf{R} \tag{11.24}$$

(note that $H_{23} = H_{13}/P_0^{\text{std}}$), where

$$l(\omega) = \frac{1}{\sqrt{2\left(|W_{\text{rel\_err}}(j\omega)|^2 + |1/(75 P_0^{\text{std}}(j\omega))|^2\right)}},$$

which is also shown in figure 11.20. The exact region is not in general convex, although in this case it happens to be convex (see the Notes and References in chapter 10).

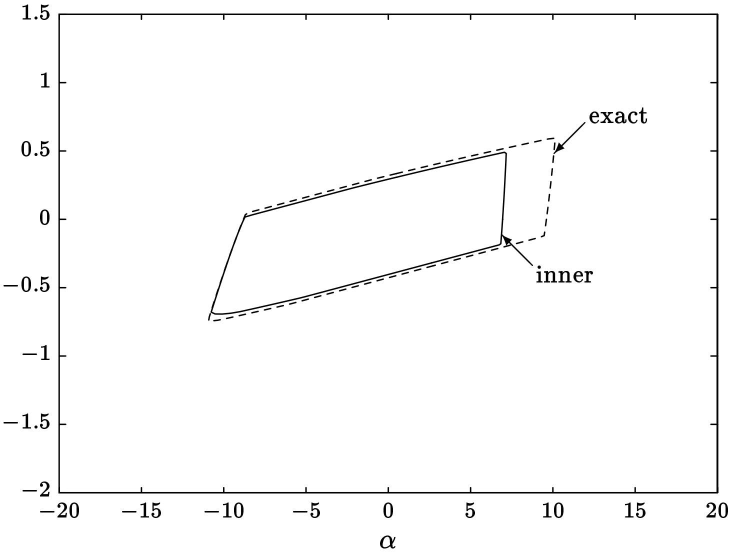


Figure 11.20 The boundary of the exact region where the robust performance specification (11.23) is met is shown, together with the inner approximation (11.24).

Figure 11.21 shows the exact specification (11.23), together with the convex inner approximation (11.24) and a convex *outer* approximation (*i.e.*, a specification that is weaker than (11.23)). The outer approximation is the simultaneous satisfaction of the specifications

$$\mathcal{D}_{\mathrm{rob\_stab}}(\mathcal{P}) \quad \text{and} \quad \|H_{23}\|_\infty \le 75, \tag{11.25}$$

described in sections 11.5.3 and 11.3.3 respectively. The specifications (11.25) require robust stability, and that the *nominal* system has an RMS gain from the reference to actuator signal not exceeding 75. The robust performance specification (11.23) therefore implies (11.25), so (11.25) is an outer approximation of the robust performance specification (11.23). The outer approximation in figure 11.21 is the set of α, β for which

$$\begin{aligned}
&\left\| W_{\mathrm{rel\_err}} \left(\alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})} \right) \right\|_\infty < 1, \\
&\left\| \alpha H_{23}^{(\mathrm{a})} + \beta H_{23}^{(\mathrm{b})} + (1-\alpha-\beta) H_{23}^{(\mathrm{c})} \right\|_\infty \le 75,
\end{aligned} \tag{11.26}$$

(see figure 10.21).

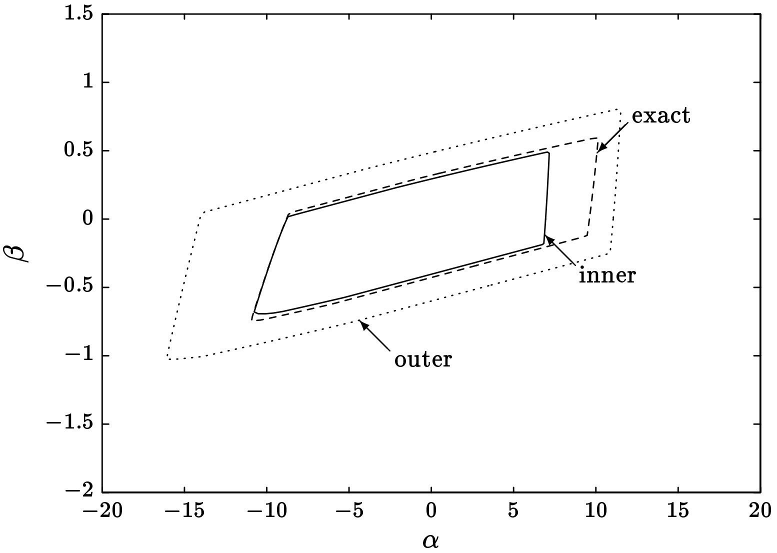


Figure 11.21 The boundary of the exact region where the robust performance specification (11.23) is met is shown, together with the inner approximation (11.24) and the outer approximation (11.26). The outer approximation is the intersection of the nominal performance specification and the robust stability specification (see figures 11.12 and 11.19).

11.6 Nonconvex Design Specifications

11.6.1 Controller Stability

Figure 11.22 shows the subset of $\mathcal{H}_{\mathrm{slice}}$ that is achieved by an open-loop stable controller, *i.e.*,

$$\left\{ [\alpha \;\; \beta]^T \;\middle|\; \begin{array}{l} \alpha H^{(\mathrm{a})} + \beta H^{(\mathrm{b})} + (1-\alpha-\beta) H^{(\mathrm{c})} \text{ is} \\ \text{achieved by a stable controller } K \end{array} \right\}. \tag{11.27}$$

From figure 11.22, we see that (11.27) is a nonconvex subset of $\mathcal{H}_{\mathrm{slice}}$. We conclude that a specification requiring open-loop controller stability,

$$\mathcal{H}_{\mathrm{k\_stab}} = \left\{ H \;\middle|\; \begin{array}{l} H = P_{zw} + P_{zu} K (I - P_{yu} K)^{-1} P_{yw} \\ \text{for some stable } K \text{ that stabilizes } P \end{array} \right\},$$

is not in general convex.

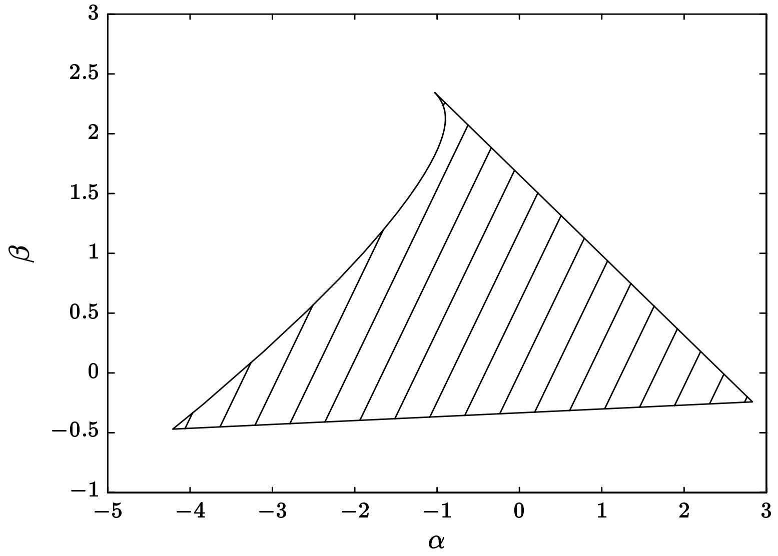


Figure 11.22 Region where the closed-loop transfer matrix H is achieved by a controller that is open-loop stable. It is not convex.

11.7 A Weighted-Max Functional

Consider the functional

$$\varphi_{\mathrm{wt\_max}}(\alpha,\beta) \overset{\Delta}{=} \max\left\{ \varphi_{\mathrm{pk\_trk}}(\alpha,\beta),\; 0.5\varphi_{\mathrm{max\_sens}}(\alpha,\beta),\; 15\varphi_{\mathrm{rms\_yp}}(\alpha,\;\beta) \right\},$$

where the functions $\varphi_{\mathrm{pk\_trk}}$, $\varphi_{\mathrm{max\_sens}}$, and $\varphi_{\mathrm{rms\_yp}}$ are given by (11.2), (11.18), and (11.6). The level curves of the function $\varphi_{\mathrm{wt\_max}}(\alpha, \beta)$ are shown in figure 11.23. The function $\varphi_{\mathrm{wt\_max}}$ will be used for several examples in chapter 14.

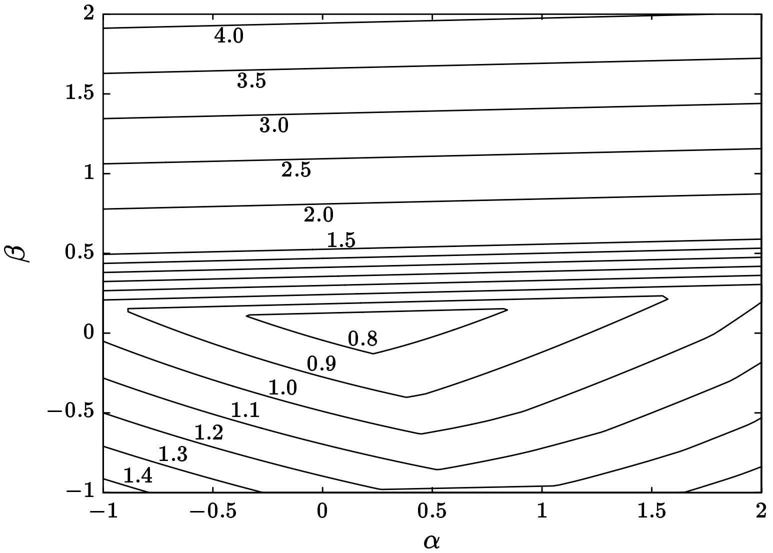


Figure 11.23 The level curves of $\varphi_{\mathrm{wt\_max}}(\alpha, \beta)$.

Notes and References

How These Figures Were Computed

Most level curves of convex functions were plotted using a radial bisection method. Consider the problem of plotting the $\phi(H) = \gamma$ level curve on $\mathcal{H}_{\rm slice}$, where ϕ is convex. Assume that we know some point (α_0, β_0) inside this level curve, *i.e.*

$$\phi\left(\alpha_0 H^{(\rm a)} + \beta_0 H^{(\rm b)} + (1-\alpha_0-\beta_0)H^{(\rm c)}\right) < \gamma.$$

The value of ϕ along the radial line segment

$$\alpha = \alpha_0 + \lambda\cos(\theta), \qquad \beta = \beta_0 + \lambda\sin(\theta), \tag{11.28}$$

where $\lambda \geq 0$, is

$$\varphi_\theta(\lambda) \stackrel{\Delta}{=} \phi\left((\alpha_0+\lambda\cos(\theta))\left(H^{(\rm a)}-H^{(\rm c)}\right) + (\beta_0+\lambda\sin(\theta))\left(H^{(\rm b)}-H^{(\rm c)}\right) + H^{(\rm c)}\right).$$

For each θ, φ_θ is a convex function from $\mathbf{R}_+$ to $\mathbf{R}$ with $\varphi_\theta(0) < \gamma$, so there is no more than one $\lambda \geq 0$ for which

$$\varphi_\theta(\lambda) = \gamma. \tag{11.29}$$

(11.29) can be solved using a number of standard methods, such as bisection or regula falsi, with only an evaluation of φ_θ required at each iteration. As θ sweeps out the angles $0 \leq \theta < 2\pi$, the solution λ to (11.29), together with (11.28), sweeps out the desired level curve. This method was used for figures 11.4, 11.5, 11.12, 11.13, 11.17, 11.19, 11.20, 11.21, and 11.23.

The above method also applies to quasiconvex functionals with the following modification: in place of (11.29) we need to find the largest λ for which $\varphi_\theta(\lambda) \leq \gamma$.

In certain cases (11.29), or its quasiconvex modification, can be solved directly. For example, consider the quasiconvex settling-time functional $\phi_{\rm settle}$ from section 11.1.1. For a fixed θ, the step response along the line (11.28) is of the form

$$s_0(t) + \lambda s_1(t),$$

where s_0 is the step response of

$$\alpha_0 H_{13}^{(\rm a)} + \beta_0 H_{13}^{(\rm b)} + (1-\alpha_0-\beta_0)H_{13}^{(\rm c)},$$

and s_1 is the step response of

$$\cos(\theta)H_{13}^{(\rm a)} + \sin(\theta)H_{13}^{(\rm b)} + (-\cos(\theta)-\sin(\theta))H_{13}^{(\rm c)}.$$

The largest value of λ for which $\varphi_\theta(\lambda) \leq T_{\max}$ is given by

$$\lambda^* = \max_{0.95 \leq s_0(t)+\lambda s_1(t) \leq 1.05 \text{ for } t \geq T_{\max}} \lambda.$$

This is a linear program in the scalar variable λ that can be directly solved.

A similar method was used to produce figures 11.8 and 11.10. A similar method could also be used to produce level curves of $\mathbf{H}_\infty$ norms; at each frequency ω a quadratic in λ can be solved to find the positive value of λ that makes the frequency response magnitude tight at ω. Taking the minimum λ over all ω gives the desired λ^*.

Figures 11.9 and 11.11 were plotted by directly computing the equations of the level curves using a state-space method. For example, consider $\|H_{12}\|_2^2$, which is one term in the functional $\phi_{\mathrm{rms\_yp}}$. Since

$$\alpha H_{12}^{(\mathrm{a})} + \beta H_{12}^{(\mathrm{b})} + (1-\alpha-\beta) H_{12}^{(\mathrm{c})} = H^{(\mathrm{c})} + \alpha\left(H^{(\mathrm{a})} - H^{(\mathrm{c})}\right) + \beta\left(H^{(\mathrm{a})} - H^{(\mathrm{c})}\right)$$

is affine in α and β, it has a state-space realization

$$C(sI-A)^{-1}(B_0 + \alpha B_1 + \beta B_2),$$

where C is a row vector, and B_0, B_1 and B_2 are column vectors. From section 5.6.1, if W_{obs} is the solution to the Lyapunov equation

$$A^T W_{\mathrm{obs}} + W_{\mathrm{obs}} A + C^T C = 0,$$

we have

$$\left\|\alpha H_{12}^{(\mathrm{a})} + \beta H_{12}^{(\mathrm{b})} + (1-\alpha-\beta) H_{12}^{(\mathrm{c})}\right\|_2^2 = \begin{bmatrix} 1 & \alpha & \beta \end{bmatrix} E \begin{bmatrix} 1 \\ \alpha \\ \beta \end{bmatrix},$$

where

$$E = \begin{bmatrix} B_0^T \\ B_1^T \\ B_2^T \end{bmatrix} W_{\mathrm{obs}} \begin{bmatrix} B_0 & B_1 & B_2 \end{bmatrix}$$

is a positive definite 3×3 matrix. The level curves of $\|H_{12}\|_2$ on $\mathcal{H}_{\mathrm{slice}}$ are therefore ellipses.

The convex sets shown in figures 11.16 and 11.18 were produced using the standard radial method described above. The exact contour along which the gain margin specification was tight was also found by a radial method. However, since this specification need not be convex, a fine grid search was also used to verify that the entire set had been correctly determined.

The step response sensitivity plot in figure 11.15 was produced by forming an indefinite quadratic in α and β. The sensitivity, from (11.13), is the step response of

$$\left(1 - \left(\alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})}\right)\right)\left(\alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})}\right)$$

at $t = 1$. After expansion, the step response of each term, at $t = 1$, gives each of the coefficients in an indefinite quadratic form in α and β. Figure 11.15 shows the α, β for which

$$-0.75 \le \begin{bmatrix} 1 & \alpha & \beta \end{bmatrix} \begin{bmatrix} 0.7149 & -0.0055 & 0.1180 \\ -0.0055 & -0.0074 & 0.0447 \\ 0.1180 & 0.0447 & -0.2493 \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \beta \end{bmatrix} \le 0.75.$$

The controller stability plot in figure 11.22 was produced by finding points where the transfer function $S(j\omega)/\omega^2$ vanished for some frequency ω. Since $S = 1/(1 + P_0^{\mathrm{std}}K)$ vanishes wherever P_0^{std} or K has a $j\omega$ axis pole, the $j\omega$ axis poles of K are exactly the $j\omega$ axis zeros of $S(j\omega)/\omega^2$; the factor of ω^2 cancels the two zeros at $s = 0$ that S inherits from the two $s = 0$ poles of P_0^{std}. At each frequency ω, the linear equations in α and β

$$\Re\left(1 - \left(\alpha H_{13}^{(\mathrm{a})}(j\omega) + \beta H_{13}^{(\mathrm{b})}(j\omega) + (1-\alpha-\beta)H_{13}^{(\mathrm{c})}(j\omega)\right)\right)\Big/\omega^2 = 0,$$
$$\Im\left(1 - \left(\alpha H_{13}^{(\mathrm{a})}(j\omega) + \beta H_{13}^{(\mathrm{b})}(j\omega) + (1-\alpha-\beta)H_{13}^{(\mathrm{c})}(j\omega)\right)\right)\Big/\omega^2 = 0$$

may be dependent, independent, or inconsistent; their solution in the first two cases gives either a line or point in the $(\alpha,\ \beta)$ plane. When these lines and points are plotted over all frequencies they determine subsets of $\mathcal{H}_{\mathrm{slice}}$ over which the controller K has a constant number of unstable (right half-plane) poles. By checking any one controller inside each subset of $\mathcal{H}_{\mathrm{slice}}$ for open-loop stability, each subset of $\mathcal{H}_{\mathrm{slice}}$ can be labeled as being achieved by stable or unstable controllers.

Part IV

NUMERICAL METHODS

Chapter 12

Some Analytic Solutions

We describe several families of controller design problems that can be solved rapidly and exactly using standard methods.

12.1 Linear Quadratic Regulator

The linear quadratic regulator (LQR) from optimal control theory can be used to solve a family of regulator design problems in which the state is accessible and regulation and actuator effort are each measured by mean-square deviation. A stochastic formulation of the LQR problem is convenient for us; a more usual formulation is as an optimal control problem (see the Notes and References at the end of this chapter). The system is described by

$$\dot{x} = Ax + Bu + w,$$

where w is a zero-mean white noise, *i.e.*, w has power spectral density matrix $S_w(\omega) = I$ for all ω. The state x is available to the controller, so $y = x$ in our framework.

The LQR cost function is the sum of the steady-state mean-square weighted state x, and the steady-state mean-square weighted actuator signal u:

$$J_{\text{lqr}} = \lim_{t\to\infty} \mathbf{E}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right),$$

where Q and R are positive semidefinite weight matrices; the first term penalizes deviations of x from zero, and the second term represents the cost of using the actuator signal. We can express this cost in our framework by forming the regulated output signal

$$z = \begin{bmatrix} R^{\frac{1}{2}} u \\ Q^{\frac{1}{2}} x \end{bmatrix},$$

so that

$$J_{\mathrm{lqr}} = \lim_{t\to\infty} \mathbf{E}\, z(t)^T z(t),$$

the mean-square deviation of z. Since w is a white noise, we have (see section 5.2.2)

$$J_{\mathrm{lqr}} = \|H\|_2^2,$$

the square of the $\mathbf{H}_2$ norm of the closed-loop transfer matrix.

In our framework, the plant for the LQR regulator problem is given by

$$\begin{aligned} A_P &= A \\ B_u &= B \\ B_w &= I \\ C_z &= \begin{bmatrix} 0 \\ Q^{\frac{1}{2}} \end{bmatrix} \\ C_y &= I \\ D_{zw} &= 0 \\ D_{zu} &= \begin{bmatrix} R^{\frac{1}{2}} \\ 0 \end{bmatrix} \\ D_{yw} &= 0 \\ D_{yu} &= 0 \end{aligned}$$

(the matrices on left-hand side refer to the state-space equations from section 2.5). This is shown in figure 12.1.

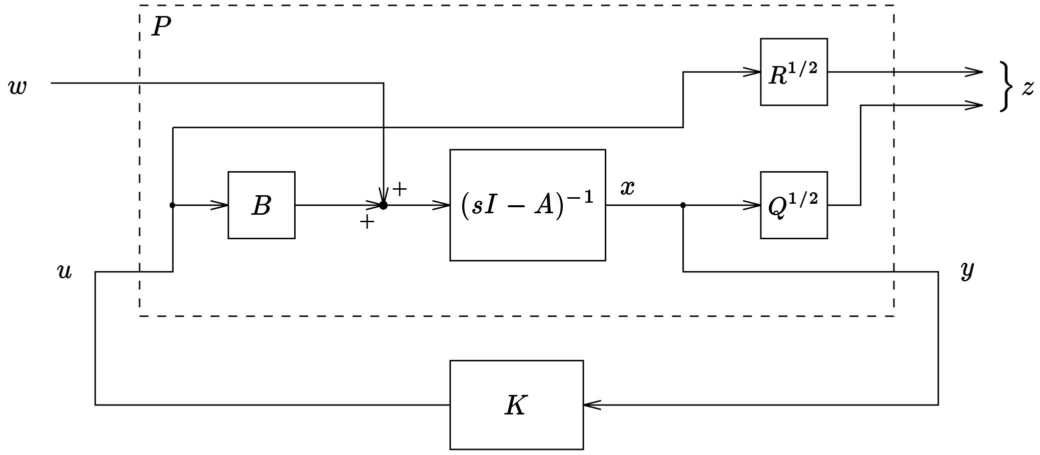


Figure 12.1 The LQR cost is $\|H\|_2^2$.

The specifications that we consider are realizability and the functional inequality specification

$$\|H\|_2 \le \alpha. \tag{12.1}$$

Standard assumptions are that (Q, A) is observable, (A, B) is controllable, and $R > 0$, in which case the specification (12.1) is stronger than (*i.e.*, implies) internal stability. (Recall our comment in chapter 7 that internal stability is often a redundant addition to a sensible set of specifications.) With these standard assumptions, there is actually a controller that achieves the smallest achievable LQR cost, and it turns out to be a constant state-feedback,

$$K_{\mathrm{lqr}}(s) = -K_{\mathrm{sfb}},$$

which can be found as follows.

Let X_{lqr} denote the unique positive definite solution of the algebraic Riccati equation

$$A^T X_{\mathrm{lqr}} + X_{\mathrm{lqr}} A - X_{\mathrm{lqr}} B R^{-1} B^T X_{\mathrm{lqr}} + Q = 0. \qquad (12.2)$$

One method of finding this X_{lqr} is to form the associated Hamiltonian matrix

$$M = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}, \qquad (12.3)$$

and then compute any matrix T such that

$$T^{-1}MT = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix},$$

where $\tilde{A}_{11}$ is stable. (One good choice is to compute an ordered Schur form of M_γ; see the Notes and References in chapter 5.) We then partition T as

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix},$$

and the solution X_{lqr} is given by

$$X_{\mathrm{lqr}} = T_{21} T_{11}^{-1}.$$

(We encountered a similar ARE in section 5.6.3; this solution method is analogous to the one described there.)

Once we have found X_{lqr}, we have

$$K_{\mathrm{sfb}} = R^{-1} B^T X_{\mathrm{lqr}},$$

which achieves LQR cost

$$J_{\mathrm{lqr}}^* = \mathbf{Tr}\, X_{\mathrm{lqr}}.$$

In particular, the specification (12.1) (along with realizability) is achievable if and only if $\alpha \geq \sqrt{\mathbf{Tr}\, X_{\mathrm{lqr}}}$, in which case the LQR-optimal controller K_{lqr} achieves the specifications.

In practice, this analytic solution is not used to solve the feasibility problem for the one-dimensional family of specifications indexed by α; rather it is used to solve multicriterion optimization problems involving actuator effort and state excursion, by solving the LQR problem for various weights R and Q. This is explained further in section 12.2.1.

12.2 Linear Quadratic Gaussian Regulator

The linear quadratic Gaussian (LQG) problem is a generalization of the LQR problem to the case in which the state is not sensed directly. For the LQG problem we consider the system given by

$$\dot{x} = Ax + Bu + w_{\mathrm{proc}} \tag{12.4}$$

$$y = Cx + v_{\mathrm{sensor}}, \tag{12.5}$$

where the process noise w_{proc} and measurement noise v_{sensor} are independent and have constant power spectral density matrices W and V, respectively.

The LQG cost function is the sum of the steady-state mean-square weighted state x, and the steady-state mean-square weighted actuator signal u:

$$J_{\mathrm{lqg}} = \lim_{t\to\infty} \mathbf{E}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right), \tag{12.6}$$

where Q and R are positive semidefinite weight matrices.

This LQG problem can be cast in our framework as follows. Just as in the LQR problem, we extract the (weighted) plant state x and actuator signal u as the regulated output, *i.e.*,

$$z = \begin{bmatrix} R^{\frac{1}{2}} u \\ Q^{\frac{1}{2}} x \end{bmatrix}.$$

The exogenous input consists of the process and measurement noises, which we represent as

$$\begin{bmatrix} w_{\mathrm{proc}} \\ v_{\mathrm{sensor}} \end{bmatrix} = \begin{bmatrix} W^{\frac{1}{2}} \\ V^{\frac{1}{2}} \end{bmatrix} w,$$

with w a white noise signal, *i.e.*, $S_w(\omega) = I$. The state-space description of the plant for the LQG problem is thus

$$A_P = A \tag{12.7}$$

$$B_w = \begin{bmatrix} W^{\frac{1}{2}} & 0 \end{bmatrix} \tag{12.8}$$

$$B_u = B \tag{12.9}$$

$$C_z = \begin{bmatrix} 0 \\ Q^{\frac{1}{2}} \end{bmatrix} \tag{12.10}$$

$$C_y = C \tag{12.11}$$

$$D_{zw} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{12.12}$$

$$D_{zu} = \begin{bmatrix} R^{\frac{1}{2}} \\ 0 \end{bmatrix} \tag{12.13}$$

$$D_{yw} = \begin{bmatrix} 0 & V^{\frac{1}{2}} \end{bmatrix} \tag{12.14}$$

$$D_{yu} = 0. \tag{12.15}$$

This is shown in figure 12.2.

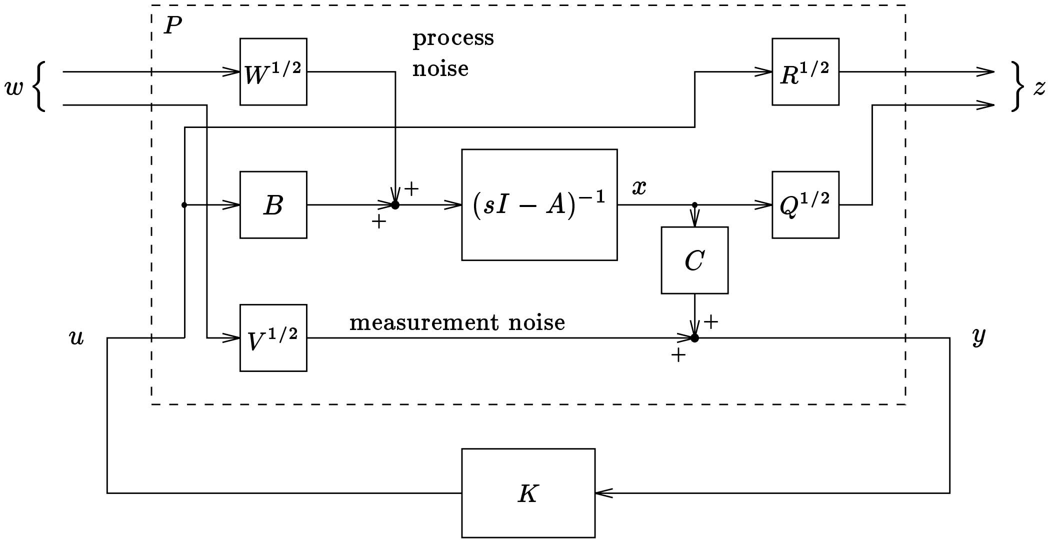


Figure 12.2 The LQG cost is $\|H\|_2^2$.

Since w is a white noise, the LQG cost is simply the variance of z, which is given by

$$J_{\mathrm{lqg}} = \|H\|_2^2.$$

The specifications for the LQG problem are therefore the same as for the LQR problem: realizability and the $\mathbf{H}_2$ norm-bound (12.1).

Standard assumptions for the LQG problem are that the plant is controllable from each of u and w, observable from each of z and y, a positive weight is used for the actuator signal ($R > 0$), and the sensor noise satisfies $V > 0$. With these standard assumptions in force, there is a unique controller K_{lqg} that minimizes the LQG objective. This controller has the form of an estimated-state-feedback controller (see section 7.4); the optimal state-feedback and estimator gains, K_{sfb} and L_{est}, can be determined by solving two algebraic Riccati equations as follows. The state-feedback gain is given by

$$K_{\mathrm{sfb}} = R^{-1}B^TX_{\mathrm{lqg}}, \tag{12.16}$$

where X_{lqg} is the unique positive definite solution of the Riccati equation

$$A^TX_{\mathrm{lqg}} + X_{\mathrm{lqg}}A - X_{\mathrm{lqg}}BR^{-1}B^TX_{\mathrm{lqg}} + Q = 0, \tag{12.17}$$

which is the same as (12.2). The estimator gain is given by

$$L_{\mathrm{est}} = Y_{\mathrm{lqg}}C^TV^{-1}, \tag{12.18}$$

where $Y_{\rm lqg}$ is the unique positive definite solution of

$$AY_{\rm lqg} + Y_{\rm lqg}A^T - Y_{\rm lqg}C^TV^{-1}CY_{\rm lqg} + W = 0 \qquad (12.19)$$

(which can be solved using the methods already described in sections 12.1 and 5.6.3). The LQG-optimal controller $K_{\rm lqg}$ is thus

$$K_{\rm lqg}(s) = -K_{\rm sfb}(sI - A + BK_{\rm sfb} + L_{\rm est}C)^{-1}L_{\rm est}, \qquad (12.20)$$

and the optimal LQG cost is

$$J^*_{\rm lqg} = \mathbf{Tr}\left(X_{\rm lqg}W + QY_{\rm lqg} + 2X_{\rm lqg}AY_{\rm lqg}\right). \qquad (12.21)$$

The specification $\|H\|_2 \leq \alpha$ (along with realizability) is therefore achievable if and only if $\alpha \geq \sqrt{J^*_{\rm lqg}}$, in which case the LQG-optimal controller $K_{\rm lqg}$ satisfies the specifications.

12.2.1 Multicriterion LQG Problem

The LQG objective (12.6) can be interpreted as a weighted-sum objective for a related multicriterion optimization problem. We consider the same system as in the LQG problem, given by (12.4–12.5); the objectives are the variances of the actuator signals,

$$\|u_1\|^2_{\rm rms}, \ldots, \|u_{n_u}\|^2_{\rm rms},$$

and some critical variables that are linear combinations of the system state,

$$\|c_1x\|^2_{\rm rms}, \ldots, \|c_mx\|^2_{\rm rms},$$

where the process and measurement noises are the same as for the LQG problem ($c_1, \ldots, c_m$ are row vectors that determine the critical variables).

We describe this multicriterion optimization problem in our framework as follows. We use the same plant as for the LQG problem, substituting

$$z = \begin{bmatrix} u_1 \\ \vdots \\ u_{n_u} \\ c_1x \\ \vdots \\ c_mx \end{bmatrix}$$

for the regulated output used there. The state-space plant equations for the multicriterion LQG problem are therefore given by (12.7–12.15), with

$$C_z = \begin{bmatrix} 0 \\ c_1 \\ \vdots \\ c_m \end{bmatrix}$$

substituted for (12.10) and

$$D_{zu} = \begin{bmatrix} I \\ 0 \end{bmatrix}$$

substituted for (12.13). The objectives are given by the squares of the $\mathbf{H}_2$ norms of the rows of the closed-loop transfer matrix:

$$\phi_i(H) \stackrel{\Delta}{=} \|H^{(i)}\|_2^2,$$

where $H^{(i)}$ is the ith row of H and $L = n_z = n_u + m$. The hard constraint for this multicriterion optimization problem is realizability.

For $1 \le i \le n_u$, $\phi_i(H)$ represents the variance of the ith actuator signal, and for $n_u + 1 \le i \le L$, $\phi_i(H)$ represents the variance of the critical variable $c_{i-n_u}x$. The design specification

$$\phi_1(H) \le a_1, \ldots, \phi_L(H) \le a_L, \tag{12.22}$$

therefore, limits the RMS values of the actuator signals and critical variables.

Consider the weighted-sum objective associated with this multicriterion optimization problem:

$$\phi_{\mathrm{wt\_sum}}(H) = \lambda_1\phi_1(H) + \cdots + \lambda_L\phi_L(H), \tag{12.23}$$

where $\lambda \ge 0$. We can express this as

$$\phi_{\mathrm{wt\_sum}}(H) = J_{\mathrm{lqg}}$$

if we choose weight matrices

$$Q = \lambda_{n_u+1}c_1^T c_1 + \cdots + \lambda_L c_m^T c_m, \tag{12.24}$$
$$R = \mathbf{diag}\,(\lambda_1, \ldots, \lambda_m) \tag{12.25}$$

($\mathbf{diag}\,(\cdot)$ is the diagonal matrix with diagonal entries given by the argument list).

Hence by solving an LQG problem, we can find the optimal design for the weighted-sum objective for the multicriterion optimization problem with functionals $\phi_1, \ldots, \phi_L$. These designs are Pareto optimal for the multicriterion optimization problem; moreover, because the objective functionals and the hard constraint are convex, *every* Pareto optimal design arises this way for some choice of the weights $\lambda_1, \ldots, \lambda_L$ (see section 6.5). Roughly speaking, by varying the weights for the LQG problem, we can "search" the whole tradeoff surface.

We note that by solving an LQG problem, we can evaluate the dual function ψ described in section 3.6.2:

$$\begin{aligned}\psi(\lambda) &= \min\,\{\lambda_1\phi_1(H) + \cdots + \lambda_L\phi_L(H) \mid H \text{ is realizable }\} \\ &= J_{\mathrm{lqg}}^*,\end{aligned}$$

given by (12.21), using the weights (12.24–12.25). We will use this fact in section 14.5, where we describe an algorithm for solving the feasibility problem (12.22).

12.3 Minimum Entropy Regulator

The LQG solution method described in section 12.2 was recently modified to find the controller that minimizes the γ-entropy of H, defined in section 5.3.5. Since the γ-entropy of H is finite if and only if its $\mathbf{H}_\infty$ norm is less than γ, this analytic solution method can be used to solve the feasibility problem with the inequality specification $\|H\|_\infty < \gamma$.

The plant is identical to the one considered for the LQG problem, given by (12.7–12.15); we also make the same standard assumptions for the plant that we made for the LQG case. The design specifications are realizability and the $\mathbf{H}_\infty$ norm inequality specification

$$\|H\|_\infty < \gamma \tag{12.26}$$

(which are stronger than internal stability under the standard assumptions). We will show how to solve the feasibility problem for this one-dimensional family of design specifications.

It turns out that if the design specification (12.26) (along with realizability) is achievable, then it is achievable by a controller that is, except for a scale factor, an estimated-state-feedback controller. This controller can be found as follows. If γ is such that the specification (12.26) is feasible, then the two algebraic Riccati equations

$$A^T X_{\mathrm{me}} + X_{\mathrm{me}} A - X_{\mathrm{me}}(BR^{-1}B^T - \gamma^{-2}W)X_{\mathrm{me}} + Q = 0 \tag{12.27}$$

(*c.f.* (12.17)), and

$$AY_{\mathrm{me}} + Y_{\mathrm{me}} A^T - Y_{\mathrm{me}}(C^T V^{-1} C - \gamma^{-2}Q)Y_{\mathrm{me}} + W = 0 \tag{12.28}$$

(*c.f.* (12.19)) have unique positive definite solutions X_{me} and Y_{me}, respectively. (The mnemonic subscript "me" stands for "minimum entropy".) These solutions can be found by the method described in section 12.1, using the associated Hamiltonian matrices

$$\begin{bmatrix} A & -(BR^{-1}B^T - \gamma^{-2}W) \\ -Q & -A^T \end{bmatrix}, \qquad \begin{bmatrix} A & -W \\ -(C^T V^{-1} C - \gamma^{-2}Q) & -A^T \end{bmatrix};$$

if either of these matrices has imaginary eigenvalues, then the corresponding ARE does not have a positive definite solution, and the specification (12.26) is not feasible.

From X_{me} and Y_{me} we form the matrix

$$X_{\mathrm{me}}(I - \gamma^{-2} Y_{\mathrm{me}} X_{\mathrm{me}})^{-1}, \tag{12.29}$$

which can be shown to be symmetric. If this matrix is not positive definite (or the inverse fails to exist), then the specification (12.26) (along with realizability) is not feasible.

If, on the other hand, the positive definite solutions $X_{\rm me}$ and $Y_{\rm me}$ exist, and the matrix (12.29) exists and is positive definite, then the specification (12.26) (along with realizability) is feasible. Let

$$K_{\rm sfb} = R^{-1}B^T X_{\rm me}(I - \gamma^{-2}Y_{\rm me}X_{\rm me})^{-1} \tag{12.30}$$

and

$$L_{\rm est} = Y_{\rm me}C^T V^{-1} \tag{12.31}$$

(*c.f.* (12.16) and (12.18)). A controller that achieves the design specifications is given by

$$K_{\rm me}(s) = -K_{\rm sfb}\left(sI - A + BK_{\rm sfb} + L_{\rm est}C - \gamma^{-2}Y_{\rm me}Q\right)^{-1} L_{\rm est}$$

(*c.f.* the LQG-optimal controller (12.20)).

12.4 A Simple Rise Time, Undershoot Example

In this section and the next we show how to find explicit solutions for two specific plants and families of design specifications.

We consider the classical 1-DOF system of section 2.3.2 with

$$P_0(s) = \frac{s-1}{(s+1)^2}.$$

It is well-known in classical control that since P_0 has a real unstable zero at $s = 1$, the step response from the reference input r to the system output $y_{\rm p}$, $s_{13}(t)$, must exhibit some undershoot. We will study exactly how much it must undershoot, when we require that a stabilizing controller also meet a minimum rise-time specification.

Our design specifications are internal stability, a limit on undershoot,

$$\phi_{\rm us}(H_{13}) \le U_{\rm max}, \tag{12.32}$$

and a limit on rise time,

$$\phi_{\rm rise}(H_{13}) \le T_{\rm max}. \tag{12.33}$$

Thus we have a two-parameter family of design specifications, indexed by $U_{\rm max}$ and $T_{\rm max}$.

These design specifications are simple enough that we can readily solve the feasibility problem for each $U_{\rm max}$ and $T_{\rm max}$. We will see, however, that these design specifications are not complete enough to guarantee reasonable controller designs; for example, they include no limit on actuator effort. We will return to this point later.

We can express the design specification of internal stability in terms of the interpolation conditions (section 7.2.5) for T, the I/O transfer function: T is stable and satisfies

$$T(1) = T(\infty) = 0. \tag{12.34}$$

This in turn can be expressed in terms of the step response $s_{13}(t)$: s_{13} is the step response of a stable transfer function and satisfies

$$\int_0^\infty s_{13}(t)e^{-t}\,dt = 0, \tag{12.35}$$

$$s_{13}(0) = 0. \tag{12.36}$$

Now if (12.33) holds then

$$\int_{T_{\max}}^\infty s_{13}(t)e^{-t}\,dt \geq 0.8\int_{T_{\max}}^\infty e^{-t}\,dt = 0.8e^{-T_{\max}}. \tag{12.37}$$

If (12.32) holds then

$$\int_0^{T_{\max}} s_{13}(t)e^{-t}\,dt \geq -U_{\max}\int_0^{T_{\max}} e^{-t}\,dt = -U_{\max}\left(1 - e^{-T_{\max}}\right). \tag{12.38}$$

Adding (12.37) and (12.38) we have

$$0 = \int_0^\infty s_{13}(t)e^{-t}\,dt \geq 0.8e^{-T_{\max}} - U_{\max}\left(1 - e^{-T_{\max}}\right).$$

Hence if the design specifications with $U_{\max}$ and $T_{\max}$ are feasible,

$$U_{\max} \geq \frac{0.8e^{-T_{\max}}}{1 - e^{-T_{\max}}}.$$

This relation is shown in figure 12.3. We have shown that every achievable undershoot, rise-time specification must lie in the shaded region of figure 12.3; in other words, the shaded region in figure 12.3 includes the region of achievable specifications in performance space.

In fact, the specifications with limits $U_{\max}$ and $T_{\max}$ are achievable if and only if

$$U_{\max} > \frac{0.8e^{-T_{\max}}}{1 - e^{-T_{\max}}}, \tag{12.39}$$

so that the shaded region in figure 12.3 is exactly the region of achievable specifications for our family of design specifications.

We will briefly explain why this is true. Suppose that $U_{\max}$ and $T_{\max}$ satisfy (12.39). We can then find a step response $s_{13}(t)$ of a stable rational transfer function, that satisfies the interpolation conditions (12.35–12.36) and the overshoot

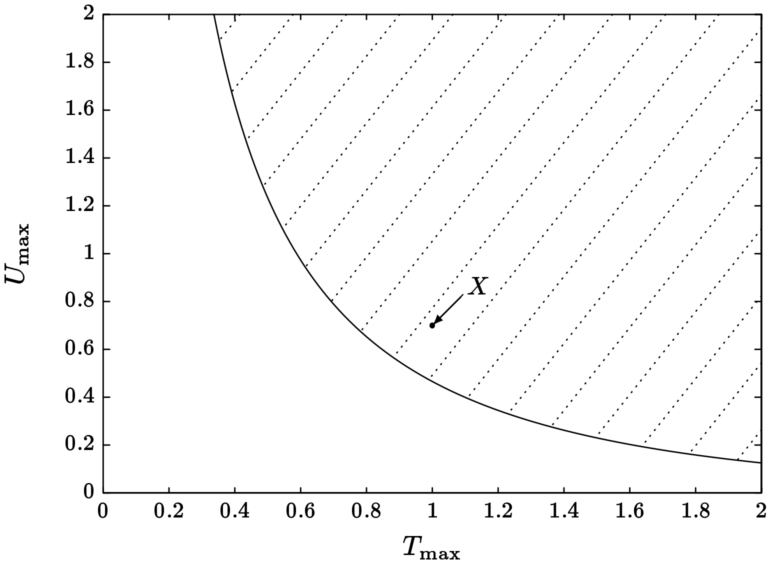


Figure 12.3 The tradeoff between achievable undershoot and rise-time specifications.

and undershoot limits. If $U_{\max}$ and $T_{\max}$ are near the boundary of the region of achievable specifications, this step response will have to "hug" (but not violate) the two constraints. For $U_{\max} = 0.70$ and $T_{\max} = 1.0$ (marked "X" in figure 12.3) a suitable step response is shown in figure 12.4; it is the step response of a 20th order transfer function (and corresponds to a controller K of order 22). (A detailed justification that we can always design such a step response is quite cumbersome; we have tried to give the general idea. See the Notes and References at the end of this chapter for more detail about this particular transfer function.)

The rapid changes near $t = 0$ and $t = 1$ of the step response shown in figure 12.4 suggest very large actuator signals, and this can be verified. It should be clear that for specifications $U_{\max}$, $T_{\max}$ that are nearly Pareto optimal, such rapid changes in the step response, and hence large actuator signals, will be necessary. So controllers that achieve specifications near the tradeoff curve are probably not reasonable from a practical point of view; but we point out that this "side information" that the actuator signal should be limited was not included in our design specifications. The fact that our specifications do not limit actuator effort, and therefore are probably not sensible, is reflected in the fact that the Pareto optimal specifications, which satisfy

$$U_{\max} = \frac{0.8e^{-T_{\max}}}{1 - e^{-T_{\max}}},$$

are not achievable (see the comments at the end of section 3.5).

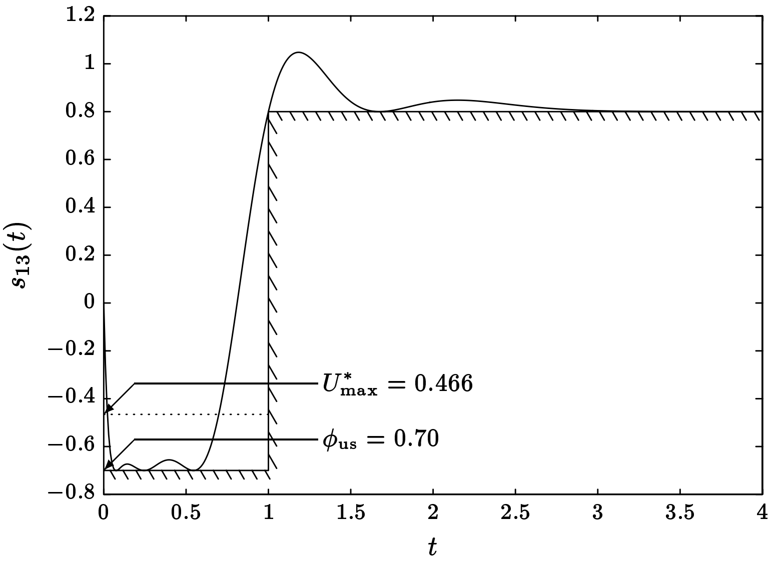


Figure 12.4 A step response with an undershoot of 0.70 and a rise time of 1.0, which achieves the specifications marked "X" in figure 12.3. Undershoot as small as 0.466 with a rise time of 1.0 are also achievable.

The tradeoff curve in figure 12.3 is valuable even though the design specifications do not limit actuator effort. If we add to our design specifications an appropriate limit on actuator effort, the new tradeoff curve will lie above the one we have found. Thus, our tradeoff curve identifies design specifications that are not achievable, *e.g.*, $U_{\rm max} = 0.4$, $T_{\rm max} = 1.0$, when no limit on actuator effort is made; a fortiori these design specifications are not achievable when a limit on actuator effort is included.

We remark that the tradeoff for this example is considerably more general than the reader might suspect. (12.39) holds for

- any LTI plant with $P_0(1) = 0$,
- the 2-DOF controller configuration,
- any nonlinear or time-varying controller.

This is because, no matter how the plant input u is generated, the output of P_0, $y_{\rm p}$, must satisfy conditions of the form (12.35–12.36).

12.5 A Weighted Peak Tracking Error Example

In this section we present a less trivial example of a plant and family of design specifications for which we can explicitly solve the feasibility problem.

We consider the classical 1-DOF system of section 2.3.2 with

$$P_0(s) = \frac{s-2}{s^2-1}.$$

Designing a controller for this plant is quite demanding, since it has an unstable zero at $s = 2$ along with an unstable pole only an octave lower, at $s = 1$.

Our design specifications will be internal stability and a limit on a weighted peak gain of the closed-loop tracking error transfer function:

$$\|WS\|_{\text{pk\_gn}} \leq E_{\max}, \tag{12.40}$$

where

$$W(s) = \frac{1}{1 + sT_{\text{trk}}},$$

and $-S$ is the closed-loop transfer function from the reference input r to the error $e = -r + y_{\text{p}}$ (see sections 5.2.5 and 8.1.2). Thus we have a two-parameter family of design specifications, indexed by $E_{\max}$ and T_{trk}.

Roughly speaking, $E_{\max}$ is an approximate limit on the worst case peak mistracking that can occur with reference inputs that are bounded by one and have a bandwidth $1/T_{\text{trk}}$. Therefore, $1/T_{\text{trk}}$ represents a sort of *tracking bandwidth* for the system. It seems intuitively clear, and turns out to be correct, that small $E_{\max}$ can only be achieved at the cost of large T_{trk}.

These design specifications are simple enough that we can explicitly solve the feasibility problem for each $E_{\max}$ and T_{trk}. As in the previous section, however, these design specifications are not complete enough to guarantee reasonable controller designs, so the comments made in the previous section hold here as well.

As we did for the previous example, we express internal stability in terms of the interpolation conditions: S is stable and satisfies

$$S(1) = 0, \quad S(2) = S(\infty) = 1.$$

Equivalently, WS is stable, and satisfies

$$WS(1) = 0, \tag{12.41}$$

$$WS(2) = (1 + 2T_{\text{trk}})^{-1}, \tag{12.42}$$

$$\lim_{s\to\infty} sWS(s) = T_{\text{trk}}^{-1}. \tag{12.43}$$

Let h be the impulse response of WS, so that

$$\|WS\|_{\text{pk\_gn}} = \int_0^\infty |h(t)|\, dt$$

(from (12.43), h does not contain any impulse at $t = 0$). We can express the interpolation conditions in terms of h as

$$\int_0^\infty h(t)e^{-t}\,dt = 0, \tag{12.44}$$

$$\int_0^\infty h(t)e^{-2t}\,dt = (1+2T_{\rm trk})^{-1}, \tag{12.45}$$

$$h(0) = T_{\rm trk}^{-1}. \tag{12.46}$$

We will solve the feasibility problem by solving the optimization problem

$$\begin{array}{c}\min\\ \text{subject to } (12.44\text{–}12.46)\end{array} \int_0^\infty |h(t)|\,dt. \tag{12.47}$$

In chapters 13–15 we will describe general numerical methods for solving an infinite-dimensional convex optimization problem such as (12.47); here we will use some specific features to analytically determine the solution. We will first guess a solution, based on some informal reasoning, and then prove, using only simple arguments, that our guess is correct.

We first note that the third constraint, on $h(0)$, should not affect the minimum, since we can always adjust h very near $t = 0$ to satisfy this constraint, without affecting the other two constraints, and only slightly changing the objective. It can be shown that the value of the minimum does not change if we ignore this constraint, so henceforth we will.

Now we consider the two integral constraints. From the second, we see that $h(t)$ will need to be positive over some time interval, and from the first we see that $h(t)$ will also have to be negative over some other time interval. Since the integrand e^{-2t} falls off more rapidly than e^{-t}, it seems that the optimal $h(t)$ should first be positive, and later negative, to take advantage of these differing decay rates. Similar reasoning finally leads us to guess that a nearly optimal h should satisfy

$$h(t) \approx \alpha\delta(t) - \beta\delta(t-T), \tag{12.48}$$

where α and β are positive, and T is some appropriate time lapse. The objective is then approximately $\alpha + \beta$.

Given this form, we readily determine that the optimal α, β, and T are given by

$$\alpha = \frac{1}{1+2T_{\rm trk}}\left(1+\frac{1}{2}\sqrt{2}\right), \tag{12.49}$$

$$\beta = \frac{1}{1+2T_{\rm trk}}\left(2+\frac{3}{2}\sqrt{2}\right), \tag{12.50}$$

$$T = \log(1+\sqrt{2}), \tag{12.51}$$

which corresponds to an objective of

$$\frac{1}{1+2T_{\mathrm{trk}}}\left(3+2\sqrt{2}\right). \tag{12.52}$$

Our guess that the value of (12.47) is given by (12.52) is correct. To verify this, we consider $\lambda : \mathbf{R}_+ \to \mathbf{R}$ given by

$$\lambda(t) = -(2+2\sqrt{2})e^{-t} + (3+2\sqrt{2})e^{-2t}, \tag{12.53}$$

and plotted in figure 12.5. This function has a maximum magnitude of one, *i.e.*, $\|\lambda\|_\infty = 1$.

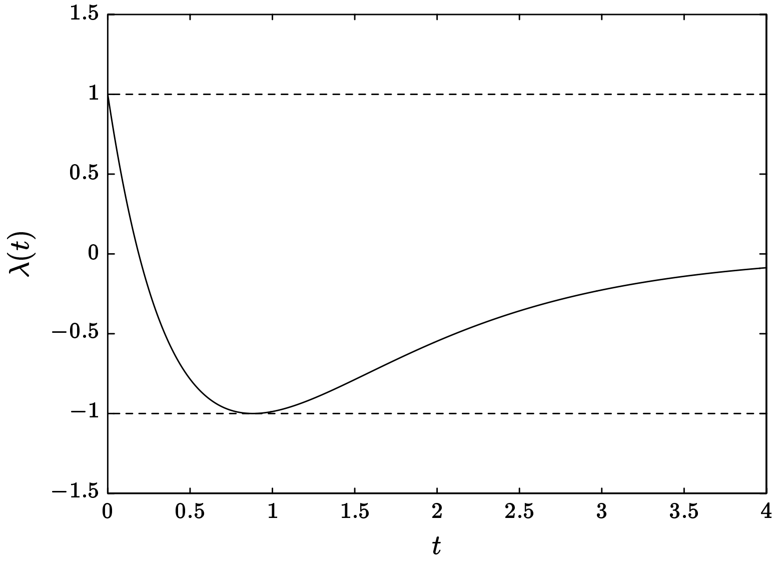


Figure 12.5 The function $\lambda(t)$ from (12.53).

Now suppose that h satisfies the two integral equality constraints in (12.47). Then by linearity we must have

$$\int_0^\infty h(t)\lambda(t)\,dt = \frac{1}{1+2T_{\mathrm{trk}}}\left(3+2\sqrt{2}\right). \tag{12.54}$$

Since $|\lambda(t)| \le 1$ for all t, we have

$$\int_0^\infty h(t)\lambda(t)\,dt \le \int_0^\infty |h(t)|\,dt = \|h\|_1. \tag{12.55}$$

Combining (12.54) and (12.55), we see that for any h,

$$\int_0^\infty h(t)e^{-t}\,dt = 0 \qquad \text{and} \qquad \int_0^\infty h(t)e^{-2t}\,dt = (1+2T_{\mathrm{trk}})^{-1}$$

$$\Longrightarrow \qquad \|h\|_1 \geq \frac{1}{1+2T_{\rm trk}}\left(3+2\sqrt{2}\right),$$

i.e., any h that satisfies the constraints in (12.47) has an objective that exceeds the objective of our candidate solution (12.52). This proves that our guess is correct. (The origin of this mysterious λ is explained in the Notes and References.)

From our solution (12.52) of the optimization problem (12.47), we conclude that the specifications corresponding to $E_{\rm max}$ and $T_{\rm trk}$ are achievable if and only if

$$E_{\rm max}(1+2T_{\rm trk}) > 3+2\sqrt{2}. \tag{12.56}$$

(We leave the construction of a controller that meets the specifications for $E_{\rm max}$ and $T_{\rm trk}$ satisfying (12.56) to the reader.) This region of achievable specifications is shown in figure 12.6.

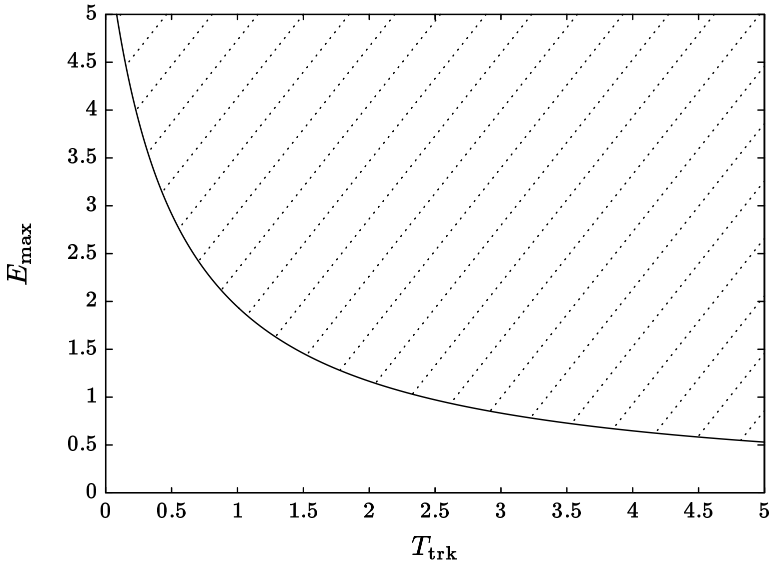


Figure 12.6 The tradeoff between peak tracking error and tracking bandwidth specifications.

Note that to guarantee that the worst case peak tracking error does not exceed 10%, the weighting filter smoothing time constant must be at least $T_{\rm trk} \geq 28.64$, which is much greater than the time constants in the dynamics of P_0, which are on the order of one second. In classical terminology, the tracking bandwidth is considerably smaller than the open-loop bandwidth. The necessarily poor performance implied by the tradeoff curve (12.56) is a quantitative expression that this plant is "hard to control".

Notes and References

LQR and LQG-Optimal Controllers

Standard references on LQR and LQG-optimal controllers are the books by Anderson and Moore [AM90], Kwakernaak and Sivan [KS72], Bryson and Ho [BH75], and the special issue edited by Athans [ATH71]. Åström and Wittenmark treat minimum variance regulators in [AW90]. The same techniques are readily extended to solve problems that involve an exponentially weighted $\mathbf{H}_2$ norm; see, *e.g.*, Anderson and Moore [AM69].

Multicriterion LQG

The articles by Toivonen [TOI84] and Toivonen and Mäkilä [TM89] discuss the multicriterion LQG problem; the latter article has extensive references to other articles on this topic. See also Koussoulas and Leondes [KL86].

Controllers that Satisfy an $\mathbf{H}_\infty$ Norm-Bound

In [ZAM81], Zames proposed that the $\mathbf{H}_\infty$ norm of some appropriate closed-loop transfer matrix be minimized, although control design specifications that limit the magnitude of closed-loop transfer functions appeared much earlier. The state-space solution of section 12.3 is recent, and is due to Doyle, Glover, Khargonekar, and Francis [DGK89, GD88]. Previous solutions to the feasibility problem with an $\mathbf{H}_\infty$ norm-bound on H were considerably more complex.

We noted above that the controller K_{me} of section 12.3 not only satisfies the specification (12.26); it minimizes the γ-entropy of H. This is discussed in Mustafa and Glover [MUS89, MG90, GM89]. The minimum entropy controller was developed independently by Whittle [WHI90], who calls it the *linear exponential quadratic Gaussian* (LEQG) optimal controller.

Some Other Analytic Solutions

In [OF85, OF86], O'Young and Francis use Nevanlinna-Pick theory to deduce exact tradeoff curves that limit the maximum magnitude of the sensitivity transfer function in two different frequency bands.

Some analytic solutions to discrete-time problems involving the peak gain have been found by Dahleh and Pearson; see [VID86, DP87B, DP88B, DP87A, DP88A].

About Figure 12.4

The step response shown in figure 12.4 was found as follows. We let

$$T(s) = \sum_{i=1}^{20} x_i \left(\frac{10}{s+10}\right)^i, \qquad (12.57)$$

where $x \in \mathbf{R}^{20}$ is to be determined. (See chapter 15 for an explanation of this *Ritz approximation.*) $T(s)$ must satisfy the condition (12.34). The constraint $T(\infty) = 0$ is automatically satisfied; the interpolation condition $T(1) = 0$ yields the equality constraint on x,

$$c^T x = 0, \qquad (12.58)$$

i	x_i	i	x_i	i	x_i	i	x_i
1	−3.027	6	−3.677	11	4.479	16	9.641
2	7.227	7	−9.641	12	−9.641	17	−1.660
3	−9.374	8	−3.018	13	−9.641	18	−9.641
4	1.836	9	9.641	14	−5.682	19	4.398
5	9.641	10	9.641	15	9.641	20	−0.343

Table 12.1 The coefficients in the parametrization (12.57) for the step response in figure 12.4.

where $c_i = (10/11)^i$. The undershoot and rise-time specifications are

$$\sum_{i=1}^{20} x_i s_i(t) \geq -0.7 \qquad \text{for } 0 \leq t \leq 1.0, \tag{12.59}$$

$$\sum_{i=1}^{20} x_i s_i(t) \geq 0.8 \qquad \text{for } t \geq 1.0, \tag{12.60}$$

where s_i is the step response of $(s/10+1)^{-i}$. By finely discretizing t, (12.59) and (12.60) yield (many) linear inequality constraints on x, *i.e.*

$$a_k^T x \leq b_k, \qquad k = 1, \ldots, L. \tag{12.61}$$

(12.58) and (12.61) can be solved as a feasibility linear program. The particular coefficients that we used, shown in table 12.1, were found by minimizing $\|x\|_\infty$ subject to (12.58) and (12.61).

About the Examples in Sections 12.4 and 12.5

These two examples can be expressed as infinite-dimensional linear programming problems. The references for the next two chapters are relevant; see also Luenberger [LUE69], Rockafellar [ROC74, ROC82], Reiland [REI80], Anderson and Philpott [AP84], and Anderson and Nash [AN87].

We solved the problem (12.47) (ignoring the third equality constraint) by first solving its *dual problem*, which is

$$\max_{\|\lambda_1 e^{-t} + \lambda_2 e^{-2t}\|_\infty \leq 1} \qquad \lambda_2 (1 + 2T_{\rm trk})^{-1}. \tag{12.62}$$

This is a convex optimization problem in $\mathbf{R}^2$, which is readily solved. The mysterious $\lambda(t)$ that we used corresponds exactly to the optimum λ_1 and λ_2 for this dual problem.

This dual problem is sometimes called a *semi-infinite optimization problem* since the constraint involves a "continuum" of inequalities (*i.e.*, $|\lambda_1 e^{-t} + \lambda_2 e^{-2t}| \leq 1$ for each $t \geq 0$). Special algorithms have been developed for these problems; see for example the surveys by Polak [POL83], Polak, Mayne, and Stimler [PMS84], and Hettich [HET78].

Chapter 13

Elements of Convex Analysis

We describe some of the basic tools of convex nondifferentiable analysis: subgradients, directional derivatives, and supporting hyperplanes, emphasizing their geometric interpretations. We show how to compute supporting hyperplanes and subgradients for the various specifications and functionals described in previous chapters.

Many of the specifications and functionals that we have encountered in chapters 8–10 are not smooth—the specifications can have “sharp corners” and the functionals need not be differentiable. Fortunately, for *convex* sets and functionals, some of the most important analytical tools do not depend on smoothness. In this chapter we study these tools. Perhaps more importantly, there are simple and effective algorithms for *convex optimization* that do not require smooth constraints or differentiable objectives. We will study some of these algorithms in the next chapter.

13.1 Subgradients

If $\phi\ :\ \mathbf{R}^n \to \mathbf{R}$ is convex and differentiable, we have

$$\phi(z) \geq \phi(x) + \nabla\phi(x)^T(z-x) \qquad \text{for all} \ \ z. \tag{13.1}$$

This means that the plane tangent to the graph of ϕ at x always lies below the graph of ϕ. If $\phi : \mathbf{R}^n \to \mathbf{R}$ is convex, but not necessarily differentiable, we will say that $g \in \mathbf{R}^n$ is a *subgradient of* ϕ *at* x if

$$\phi(z) \geq \phi(x) + g^T(z-x) \qquad \text{for all} \ \ z. \tag{13.2}$$

From (13.1), the gradient of a differentiable convex function is always a subgradient. A basic result of convex analysis is that every convex function always has at least one subgradient at every point. We will denote the set of all subgradients of ϕ at x as $\partial\phi(x)$, the *subdifferential of* ϕ *at* x.

We can think of the right-hand side of (13.2) as an affine approximation to $\phi(z)$, which is exact at $z = x$. The inequality (13.2) states that the right-hand side is a *global lower bound* on ϕ. This is shown in figure 13.1.

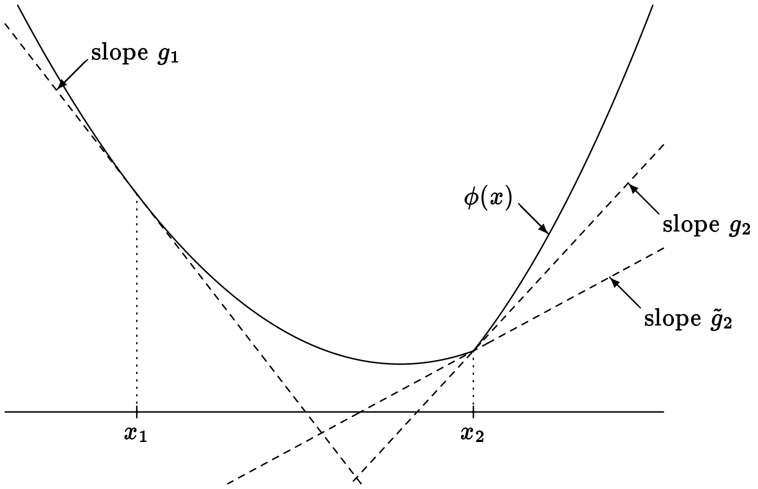


Figure 13.1 A convex function on **R** along with three affine global lower bounds on ϕ derived from subgradients. At x_1, ϕ is differentiable, and the slope of the tangent line is $g_1 = \phi'(x_1)$. At x_2, ϕ is not differentiable; two different tangent lines, corresponding to subgradients g_2 and $\tilde{g}_2$, are shown.

We mention two important consequences of $g \in \partial\phi(x)$. For $g^T(z - x) > 0$ we have $\phi(z) > \phi(x)$, in other words, in the half-space $\{z \mid g^T(z - x) > 0\}$, the values of ϕ exceed the value of ϕ at x. Thus if we are searching for an x^* that minimizes ϕ, and we know a subgradient g of ϕ at x, then we can rule out the entire half-space $g^T(z - x) > 0$. The hyperplane $g^T(z - x) = 0$ is called a *cut* because it cuts off from consideration the half-space $g^T(z - x) > 0$ in a search for a minimizer. This is shown in figure 13.2.

An extension of this idea will also be useful. From (13.2), every z that satisfies $\phi(z) \le \alpha$, where $\alpha < \phi(x)$, must also satisfy $g^T(z-x) \le \alpha - \phi(x)$. If we are searching for a z that satisfies $\phi(z) \le \alpha$, we need not consider the half-space $g^T(z - x) > \alpha - \phi(x)$. The hyperplane $g^T(z - x) = \alpha - \phi(x)$ is called a *deep-cut* because it rules out a larger set than the simple cut $g^T(z - x) = 0$. This is shown in figure 13.3.

13.1.1 Subgradients: Infinite-Dimensional Case

The notion of a subgradient can be generalized to apply to functionals on infinite-dimensional spaces. The books cited in the Notes and References at the end of this chapter contain a detailed and precise treatment of this topic; in this book we will give a simple (but correct) description.

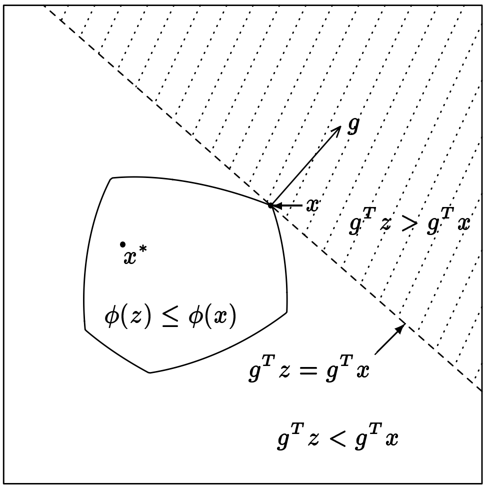


Figure 13.2 A point x and a subgradient g of ϕ at x. In the half-space $g^T z > g^T x$, $\phi(z)$ exceeds $\phi(x)$; in particular, any minimizer x^* of ϕ must lie in the half-space $g^T z \le g^T x$.

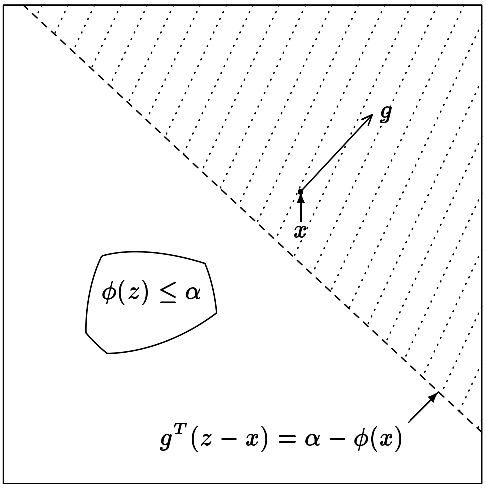


Figure 13.3 A point x and a subgradient g of ϕ at x determine a *deep-cut* in the search for points that satisfy $\phi(z) \le \alpha$ (assuming x does not satisfy this inequality). The points in the shaded region need not be considered since they all have $\phi(z) > \alpha$.

If ϕ is a convex functional on a (possibly infinite-dimensional) vector space V, then we say $\phi^{\rm sg}$ is a subgradient for ϕ at $v \in V$ if $\phi^{\rm sg}$ is a linear functional on V, and we have

$$\phi(z) \geq \phi(v) + \phi^{\rm sg}(z - v) \quad \text{for all } z \in V. \tag{13.3}$$

The subdifferential $\partial\phi(v)$ consists of all subgradients of ϕ at v; note that it is a set of linear functionals on V.

If $V = \mathbf{R}^n$, then every linear functional on V has the form $g^T z$ for some vector $g \in \mathbf{R}^n$, and our two definitions of subgradient are therefore the same, provided we ignore the distinction between the vector $g \in \mathbf{R}^n$ and the linear functional on $\mathbf{R}^n$ given by the inner product with g.

13.1.2 Quasigradients

For quasiconvex functions, there is a concept analogous to the subgradient. Suppose $\phi : \mathbf{R}^n \to \mathbf{R}$ is quasiconvex, which we recall from section 6.2.2 means that

$$\phi(\lambda x + (1-\lambda)\tilde{x}) \leq \max\{\phi(x), \phi(\tilde{x})\} \qquad \text{for all } 0 \leq \lambda \leq 1, \;\; x, \tilde{x} \in \mathbf{R}^n.$$

We say that g is a *quasigradient* for ϕ at x if

$$\phi(z) \geq \phi(x) \quad \text{whenever} \quad g^T(z - x) \geq 0. \tag{13.4}$$

This simply means that the hyperplane $g^T(z - x) = 0$ forms a simple cut for ϕ, exactly as in figure 13.2: if we are searching for a minimizer of ϕ, we can rule out the half-space $g^T(z - x) > 0$.

If ϕ is differentiable and $\nabla\phi(x) \neq 0$, then $\nabla\phi(x)$ is a quasigradient; if ϕ is convex, then (13.2) shows that any subgradient is also a quasigradient. It can be shown that every quasiconvex function has at least one quasigradient at every point. Note that the length of a quasigradient is irrelevant (for our purposes): all that matters is its direction, or equivalently, the cutting-plane for ϕ that it determines.

Any algorithm for convex optimization that uses only the cutting-planes that are determined by subgradients will also work for quasiconvex functions, if we substitute quasigradients for subgradients. It is not possible to form any deep-cut for a quasiconvex function.

In the infinite-dimensional case, we will say that a linear functional $\phi^{\rm qg}$ on V is a quasigradient for the quasiconvex functional ϕ at $v \in V$ if

$$\phi(z) \geq \phi(v) \quad \text{whenever} \quad \phi^{\rm qg}(z - v) \geq 0.$$

As discussed above, this agrees with our definition above for $V = \mathbf{R}^n$, provided we do not distinguish between vectors and the associated inner product linear functionals.

13.1.3 Subgradients and Directional Derivatives

In this section we briefly discuss the *directional derivative*, a concept of differential calculus that is more familiar than the subgradient. We will not use this concept in the optimization algorithms we present in the next chapter; we mention it because it is used in descent methods, the most common algorithms for optimization.

We define the directional derivative of ϕ at x in the direction δx as

$$\phi'(x;\ \delta x) \overset{\Delta}{=} \lim_{h \searrow 0} \frac{\phi(x + h\delta x) - \phi(x)}{h}$$

(the notation $h \searrow 0$ means that h converges to 0 from above). It can be shown that for convex ϕ this limit always exists. Of course, if ϕ is differentiable at x, then

$$\phi'(x;\ \delta x) = \nabla\phi(x)^T \delta x.$$

We say that δx is a *descent direction* for ϕ at x if $\phi'(x;\ \delta x) < 0$.

The directional derivative tells us how ϕ changes if x is moved slightly in the direction δx, since for small h,

$$\phi\left(x + h\frac{\delta x}{\|\delta x\|}\right) \approx \phi(x) + h\frac{\phi'(x;\ \delta x)}{\|\delta x\|}.$$

The *steepest descent direction* of ϕ at x is defined as

$$\delta x_{\rm sd} = \mathop{\rm arg\,min}_{\|\delta x\|=1} \phi'(x;\ \delta x).$$

In general the directional derivatives, descent directions, and the steepest descent direction of ϕ at x can be described in terms of the subdifferential at x (see the Notes and References at the end of the chapter). In many cases it is considerably more difficult to find a descent direction or the steepest descent direction of ϕ at x than a single subgradient of ϕ at x.

If ϕ is differentiable at x, and $\nabla\phi(x) \neq 0$, then $-\nabla\phi(x)$ is a descent direction for ϕ at x. It is not true, however, that the negative of any nonzero subgradient provides a descent direction: we can have $g \in \partial\phi(x)$, $g \neq 0$, but $-g$ not a descent direction for ϕ at x. As an example, the level curves of a convex function ϕ are shown in figure 13.4(a), together with a point x and a nonzero subgradient g. Note that ϕ *increases* for any movement along the directions $\pm g$, so, in particular, $-g$ is not a descent direction. Negatives of the subgradients at non-optimal points are, however, descent directions for the distance to a (or any) minimizer, *i.e.*, if $\phi(x^*) = \phi^*$ and $\psi(z) = \|z - x^*\|$, then $\psi'(x;\ -g) < 0$ for any $g \in \partial\phi(x)$. Thus, moving slightly in the direction $-g$ will *decrease* the distance to (any) minimizer x^*, as shown in figure 13.4(b).

A consequence of these properties is that the optimization algorithms described in the next chapter do not necessarily generate sequences of decreasing functional values (as would a descent method).

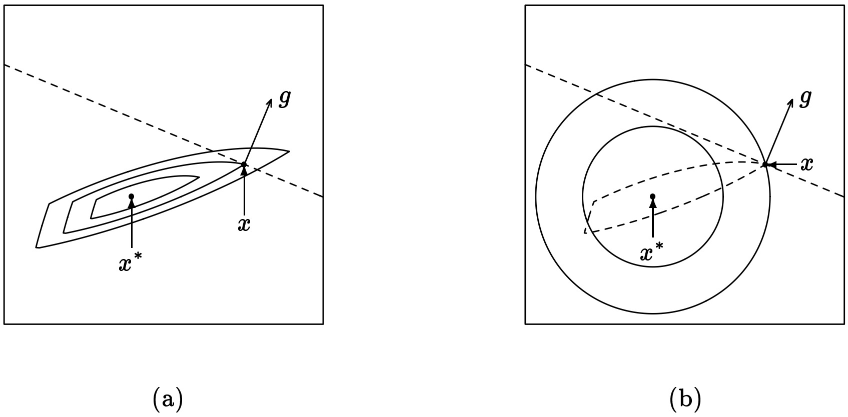


Figure 13.4 A point x and a subgradient g of ϕ at x is shown in (a), together with three level curves of ϕ. Note that $-g$ is *not* a descent direction for ϕ at x: ϕ increases for any movement from x in the directions $\pm g$. However, $-g$ is a descent direction for the distance to any minimizer. In (b) the level curves for the distance $\psi(z) = \|z - x^*\|$ are shown; $-g$ points into the circle through x.

13.2 Supporting Hyperplanes

If $\mathcal{C}$ is a convex subset of $\mathbf{R}^n$ and x is a point on its boundary, then we say that the hyperplane through x with normal g, $\{z \mid g^T(z-x) = 0\}$, is a *supporting hyperplane* to $\mathcal{C}$ at x if $\mathcal{C}$ is contained in the half-space $g^T(z-x) \leq 0$. Roughly speaking, if the set $\mathcal{C}$ is "smooth" at x, then the plane that is tangent to $\mathcal{C}$ at x is a supporting hyperplane, and g is its outward normal at x, as shown in figure 13.5. But the notion of supporting hyperplane makes sense even when the set $\mathcal{C}$ is not "smooth" at x. A basic result of convex analysis is that there is at least one supporting hyperplane at every boundary point of a convex set.

If $\mathcal{C}$ has the form of a functional inequality,

$$\mathcal{C} = \{z \mid \phi(z) \leq \alpha\},$$

where ϕ is convex (or quasiconvex), then a supporting hyperplane to $\mathcal{C}$ at a boundary point x is simply $g^T(z-x) = 0$, where g is any subgradient (or quasigradient).

If $\mathcal{C}$ is a convex subset of the infinite-dimensional space V and x is a point of its boundary, then we say that the hyperplane

$$\left\{z \mid \phi^{\mathrm{sh}}(z-x) = 0\right\},$$

where ϕ^{sh} is nonzero linear functional on V, is a supporting hyperplane for $\mathcal{C}$ at point x if

$$\phi^{\mathrm{sh}}(z-x) \leq 0 \quad \text{for all } z \in \mathcal{C}.$$

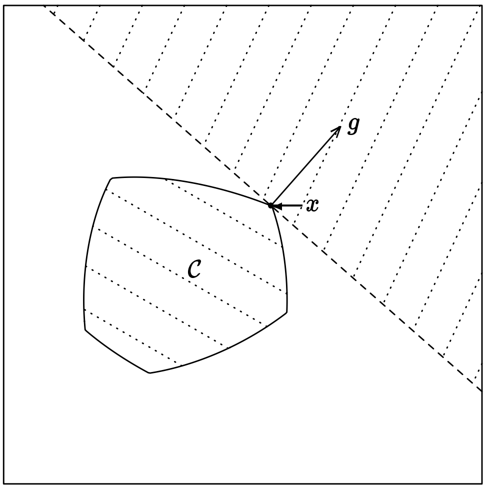


Figure 13.5 A point x on the boundary of a convex set $\mathcal{C}$. A supporting hyperplane $g^T(z-x) = 0$ for $\mathcal{C}$ at x is shown: $\mathcal{C}$ lies entirely in the half-space $g^T(z-x) \leq 0$.

Again we note that this general definition agrees with the one above for $V = \mathbf{R}^n$ if we do not distinguish between vectors in $\mathbf{R}^n$ and their associated inner product functionals.

13.3 Tools for Computing Subgradients

To use the algorithms that we will describe in the next two chapters we must be able to evaluate convex functionals and find at least one subgradient at any point. In this section, we list some useful tools for subgradient evaluation. Roughly speaking, if one can evaluate a convex functional at a point, then it is usually not much more trouble to determine a subgradient at that point.

These tools come from more general results that describe all subgradients of, for example, the sum or maximum of convex functionals. These results can be found in any of the references mentioned in the Notes and References at the end of this chapter. The more general results, however, are much more than we need, since our purpose is to show how to calculate *one* subgradient of a convex functional at a point, not *all* subgradients at a point, a task which in many cases is very difficult, and in any case not necessary for the algorithms we describe in the next two chapters. The more general results have many more technical conditions.

- *Differentiable functional*: If ϕ is convex and differentiable at x, then its derivative at x is an element of $\partial\phi(x)$. (In fact, it is the only element of $\partial\phi(x)$.)

- *Scaling*: If $w \geq 0$ and ϕ is convex, then a subgradient of $w\phi$ at x is given by wg, where g is any subgradient of ϕ at x.

- *Sum*: If $\phi(x) = \phi_1(x) + \cdots + \phi_m(x)$, where $\phi_1, \ldots, \phi_m$ are convex, then any g of the form $g = g_1 + \cdots + g_m$ is in $\partial\phi(x)$, where $g_i \in \partial\phi_i(x)$.

- *Maximum*: Suppose that

  $$\phi(x) = \sup\left\{\phi_\alpha(x) \mid \alpha \in \mathcal{A}\right\},$$

  where each ϕ_α is convex, and $\mathcal{A}$ is any index set. Suppose that $\alpha_{\mathrm{ach}} \in \mathcal{A}$ is such that $\phi_{\alpha_{\mathrm{ach}}}(x) = \phi(x)$ (so that $\phi_{\alpha_{\mathrm{ach}}}(x)$ *achieves* the maximum). Then if $g \in \partial\phi_{\alpha_{\mathrm{ach}}}(x)$, we have $g \in \partial\phi(x)$. Of course there may be several different indices that achieve the maximum; we need only pick one.

  A special case is when ϕ is the maximum of the functionals $\phi_1, \ldots, \phi_n$, so that $\mathcal{A} = \{1, \ldots, n\}$. If $\phi(x) = \phi_i(x)$, then any subgradient g of $\phi_i(x)$ is also a subgradient of $\phi(x)$.

From these tools we can derive additional tools for determining a subgradient of a weighted sum or weighted maximum of convex functionals. Their use will become clear in the next section.

For quasiconvex functionals, we have the analogous tools:

- *Differentiable functional*: If ϕ is quasiconvex and differentiable at x, with nonzero derivative, then its derivative at x is a quasigradient of ϕ at x.

- *Scaling*: If $w \geq 0$ and ϕ is quasiconvex, then any quasigradient of ϕ at x is also a quasigradient of $w\phi$ at x.

- *Maximum*: Suppose that

  $$\phi(x) = \sup\left\{\phi_\alpha(x) \mid \alpha \in \mathcal{A}\right\},$$

  where each ϕ_α is quasiconvex, and $\mathcal{A}$ is any index set. Suppose that $\alpha_{\mathrm{ach}} \in \mathcal{A}$ is such that $\phi_{\alpha_{\mathrm{ach}}}(x) = \phi(x)$. Then if g is a quasigradient of $\phi_{\alpha_{\mathrm{ach}}}$ at x, then g is a quasigradient of ϕ at x.

- *Nested family*: Suppose that ϕ is defined in terms of a nested family of convex sets, *i.e.*, $\phi(x) = \inf\{\alpha \mid x \in \mathcal{C}^\alpha\}$, where $\mathcal{C}^\alpha \subseteq \mathcal{C}^\beta$ whenever $\alpha \leq \beta$ (see section 6.2.2). If $g^T(z - x) = 0$ defines a supporting hyperplane to $\mathcal{C}^{\phi(x)}$ at x, then g is a quasigradient of ϕ at x.

(The sum tool is not applicable because the sum of quasiconvex functionals need not be quasiconvex.)

13.4 Computing Subgradients

In this section we show how to compute subgradients of several of the convex functionals we have encountered in chapters 8–10. Since these are convex functionals on $\mathcal{H}$, an infinite-dimensional space, the subgradients we derive will be linear functionals on $\mathcal{H}$. In the next section we show how these can be used to calculate subgradients in $\mathbf{R}^n$ when a finite-dimensional approximation used in chapter 15 is made; the algorithms of the next chapter can then be used.

In general, the convex functionals we consider will be functionals of some particular entry (or block of entries) of the closed-loop transfer matrix H. To simplify notation, we will assume in each subsection that H consists of only the relevant entry or entries.

13.4.1 An RMS Response

We consider the weighted $\mathbf{H}_2$ norm,

$$\phi(H) = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} S_w(\omega) |H(j\omega)|^2 \, d\omega \right)^{1/2},$$

with SISO H for simplicity (and of course, $S_w(\omega) \geq 0$). We will determine a subgradient of ϕ at the transfer function H_0. If $\phi(H_0) = 0$, then the zero functional is a subgradient, so we now assume that $\phi(H_0) \neq 0$. In this case ϕ is differentiable at H_0, so our first rule above tells us that our only choice for a subgradient is the derivative of ϕ at H_0, which is the linear functional ϕ^{sg} given by

$$\phi^{\mathrm{sg}}(H) = \frac{1}{2\pi\phi(H_0)} \int_{-\infty}^{\infty} S_w(\omega) \Re \left(\overline{H_0(j\omega)} H(j\omega) \right) \, d\omega.$$

(The reader can verify that for small H, $\phi(H_0 + H) \approx \phi(H_0) + \phi^{\mathrm{sg}}(H)$; the Cauchy-Schwarz inequality can be used to directly verify that the subgradient inequality (13.3) holds.)

Using the subgradient for ϕ, we can find a supporting hyperplane to the maximum RMS response specification $\phi(H) \leq \alpha$.

There is an analogous expression for the case when H is a transfer matrix. For $\phi(H) = \|H\|_2$ and $H_0 \neq 0$, a subgradient of ϕ at H_0 is given by

$$\phi^{\mathrm{sg}}(H) = \frac{1}{2\pi\phi(H_0)} \int_{-\infty}^{\infty} \Re \, \mathbf{Tr} \left(H_0(j\omega)^* H(j\omega) \right) \, d\omega.$$

13.4.2 Step Response Overshoot

We consider the overshoot functional,

$$\phi(H) = \sup_{t \geq 0} s(t) - 1,$$

where s is the unit step response of the transfer function H. We will determine a subgradient at H_0. The unit step response of H_0 will be denoted s_0.

We will use the rule that involves a maximum of a family of convex functionals. For each $t \geq 0$, we define a functional $\phi^{\text{step},t}$ as follows: $\phi^{\text{step},t}(H) = s(t)$. The functional $\phi^{\text{step},t}$ evaluates the step response of its argument at the time t; it is a linear functional, since we can express it as

$$\phi^{\text{step},t}(H) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{e^{j\omega t}}{j\omega} H(j\omega)\, d\omega.$$

Note that we can express the overshoot functional ϕ as the maximum of the affine functionals $\phi^{\text{step},t} - 1$:

$$\phi(H) = \sup_{t\geq 0} \phi^{\text{step},t}(H) - 1.$$

Now we apply our last rule. Let $t_0 \geq 0$ denote *any* time such that the overshoot is achieved, that is, $\phi(H_0) = s_0(t_0) - 1$. There may be several instants at which the overshoot is achieved; t_0 can be any of them. (We ignore the pathological case where the overshoot is not achieved, but only approached as a limit, although it is possible to determine a subgradient in this case as well.)

Using our last rule, we find that any subgradient of the functional $\phi^{\text{step},t_0} - 1$ is a subgradient of ϕ at H_0. But the functional $\phi^{\text{step},t_0} - 1$ is affine; its derivative is just ϕ^{step,t_0}. Hence we have determined that the linear functional ϕ^{step,t_0} is a subgradient of ϕ at H_0.

Let us verify the basic subgradient inequality (13.3). It is

$$\phi(H) \geq \phi(H_0) + \phi^{\text{step},t_0}(H - H_0).$$

Using linearity of ϕ^{step,t_0} and the fact that $\phi(H_0) = s_0(t_0) - 1 = \phi^{\text{step},t_0}(H_0) - 1$, the subgradient inequality is

$$\phi(H) \geq s(t_0) - 1.$$

Of course, this is obvious: it states that for any transfer function, the overshoot is at least as large as the value of the unit step response at the particular time t_0, minus 1.

A subgradient of other functionals involving the maximum of a time domain quantity, *e.g.*, maximum envelope violation (see section 8.1.1), can be computed in a similar way.

13.4.3 Quasigradient for Settling Time

Suppose that ϕ is the settling-time functional, defined in section 8.1.1:

$$\phi(H) = \inf\{T \mid 0.95 \leq s(t) \leq 1.05 \text{ for } t \geq T\}$$

We now determine a quasigradient for ϕ at the transfer function H_0. Let $T_0 = \phi(H_0)$, the settling time of H_0. $s_0(T_0)$ is either 0.95 or 1.05. Suppose first that $s_0(T_0) = 1.05$. We now observe that any transfer function with unit step response at time T_0 greater than or equal to 1.05, must have a settling time greater than or equal to T_0, in other words,

$$\phi(H) \geq T_0 \quad \text{whenever} \quad s(T_0) \geq 1.05.$$

Using the step response evaluation functionals introduced above, we can express this observation as

$$\phi(H) \geq \phi(H_0) \quad \text{whenever} \quad \phi^{\text{step},T_0}(H - H_0) \geq 0.$$

But this shows that the nonzero linear functional ϕ^{step,T_0} is a quasigradient for ϕ at H_0.

In general we have the quasigradient ϕ^{qg} for ϕ at H_0, where

$$\phi^{\text{qg}} = \begin{cases} \phi^{\text{step},T_0} & \text{if } s_0(T_0) = 1.05, \\ -\phi^{\text{step},T_0} & \text{if } s_0(T_0) = 0.95, \end{cases}$$

and $T_0 = \phi(H_0)$.

13.4.4 Maximum Magnitude of a Transfer Function

We first consider the case of SISO H. Suppose that

$$\phi(H) = \|H\|_\infty = \sup_{\omega \in \mathbf{R}} |H(j\omega)|,$$

provided H is stable (see section (5.2.6)). (We leave to the reader the modification necessary if ϕ is a weighted $\mathbf{H}_\infty$ norm.) We will determine a subgradient of ϕ at the stable transfer function $H_0 \neq 0$.

For each $\omega \in \mathbf{R}$, consider the functional that evaluates the magnitude of its argument (a transfer function) at the frequency $j\omega$:

$$\phi^{\text{mag},\omega}(H) = |H(j\omega)|.$$

These functionals are convex, and we can express the maximum magnitude norm as

$$\phi(H) = \sup_{\omega \in \mathbf{R}} \phi^{\text{mag},\omega}(H).$$

Thus we can use our maximum tool to find a subgradient.

Suppose that $\omega_0 \in \mathbf{R}$ is any frequency such that $|H_0(j\omega_0)| = \phi(H_0)$. (We ignore the pathological case where the supremum is only approached as a limit. In this case it is still possible to determine a subgradient.) Then any subgradient of

$\phi^{\mathrm{mag},\omega_0}$ at H_0 is a subgradient of ϕ at H_0. But since $H_0(j\omega_0) \neq 0$, this functional is differentiable at H_0, with derivative

$$\phi^{\mathrm{sg}}(H) = \frac{1}{\phi(H_0)} \Re\left(\overline{H_0(j\omega_0)} H(j\omega_0)\right).$$

This linear functional is a subgradient of ϕ at H_0. The reader can directly verify that the subgradient inequality (13.3) holds.

13.4.5 $\mathbf{H}_\infty$ Norm of a Transfer Matrix

Now suppose that H is an $m \times p$ transfer matrix, and ϕ is the $\mathbf{H}_\infty$ norm:

$$\phi(H) = \|H\|_\infty.$$

We will express ϕ directly as the maximum of a set of linear functionals, as follows. For each $\omega \in \mathbf{R}$, $u \in \mathbf{C}^m$, and $v \in \mathbf{C}^p$, we define the linear functional

$$\phi^{u,v,\omega}(H) = \Re\left(u^* H(j\omega) v\right).$$

Then we have

$$\phi(H) = \sup\left\{\phi^{u,v,\omega}(H) \;\middle|\; \omega \in \mathbf{R},\ \|u\| = \|v\| = 1\right\},$$

using the fact that for any matrix $A \in \mathbf{C}^{m \times p}$,

$$\sigma_{\max}(A) = \sup\left\{\Re(u^* A v) \;\middle|\; \|u\| = \|v\| = 1\right\}.$$

Now we can determine a subgradient of ϕ at the transfer matrix H_0. We pick any frequency $\omega_0 \in \mathbf{R}$ at which the $\mathbf{H}_\infty$ norm of H_0 is achieved, *i.e.*

$$\sigma_{\max}(H_0(j\omega_0)) = \|H_0\|_\infty.$$

(Again, we ignore the case where there is no such ω_0, commenting that for rational H_0, there always is such a frequency, if we allow $\omega_0 = \infty$.) We now compute a singular value decomposition of $H_0(j\omega_0)$:

$$H_0(j\omega_0) = U\Sigma V^*.$$

Let u_0 be the first column of U, and let v_0 be the first column of V. A subgradient of ϕ at H_0 is given by the linear functional

$$\phi^{\mathrm{sg}} = \phi^{u_0,v_0,\omega_0}.$$

13.4.6 Peak Gain

We consider the peak gain functional

$$\phi(H) = \|H\|_{\mathrm{pk\_gn}} = \int_0^\infty |h(t)|\,dt.$$

In this case our functional is an integral of a family of convex functionals. We will *guess* a subgradient of ϕ at the transfer function H_0, reasoning by analogy with the sum rule above, and then verify that our guess is indeed a subgradient. The technique of the next section shows an alternate method by which we could derive a subgradient of the peak gain functional.

Let h_0 denote the impulse response of H_0. For each $t \geq 0$ we define the functional that gives the absolute value of the impulse response of the argument at time t:

$$\phi^{\mathrm{abs\_h},t}(H) = |h(t)|.$$

These functionals are convex, and we can express ϕ as

$$\phi(H) = \int_0^\infty \phi^{\mathrm{abs\_h},t}(H)\,dt.$$

If we think of this integral as a generalized sum, then from our sum rule we might suspect that the linear functional

$$\phi^{\mathrm{sg}}(H) = \int_0^\infty \phi^{\mathrm{sg},t}(H)\,dt$$

is a subgradient for ϕ, where for each t, $\phi^{\mathrm{sg},t}$ is a subgradient of $\phi^{\mathrm{abs\_h},t}$ at H_0. Now, these functionals are differentiable for those t such that $h_0(t) \neq 0$, and 0 is a subgradient of $\phi^{\mathrm{abs\_h},t}$ at H_0 for those t such that $h_0(t) = 0$. Hence a specific choice for our guess is

$$\phi^{\mathrm{sg}}(H) = \int_0^\infty \mathrm{sgn}(h_0(t))h(t)\,dt.$$

We will verify that this is a subgradient of ϕ at H_0.

For each t and any h we have $|h(t)| \geq \mathrm{sgn}(h_0(t))h(t)$; hence

$$\phi(H) = \int_0^\infty |h(t)|dt \geq \int_0^\infty \mathrm{sgn}(h_0(t))h(t)\,dt.$$

This can be rewritten as

$$\phi(H) \geq \int_0^\infty \left(|h_0(t)| + \mathrm{sgn}(h_0(t))(h(t) - h_0(t))\right)\,dt = \phi(H_0) + \phi^{\mathrm{sg}}(H - H_0).$$

This verifies that ϕ^{sg} is a subgradient of ϕ at H_0.

13.4.7 A Worst Case Norm

We consider the particular worst case norm described in section 5.1.3:

$$\phi(H) = \|H\|_{\rm wc} = \sup\left\{\|Hu\|_\infty \;\middle|\; \|u\|_\infty \le M_{\rm ampl}, \|\dot{u}\|_\infty \le M_{\rm slew}\right\}.$$

We first rewrite ϕ as

$$\phi(H) = \sup\left\{ \int_0^\infty v(t)h(t)\,dt \;\middle|\; \|v\|_\infty \le M_{\rm ampl}, \|\dot{v}\|_\infty \le M_{\rm slew} \right\}. \qquad (13.5)$$

Now for each signal v we define the linear functional

$$\phi^v(H) = \int_0^\infty v(t)h(t)\,dt.$$

We can express the worst case norm as a maximum of a set of these linear functionals:

$$\phi(H) = \sup\left\{\phi^v(H) \;\middle|\; \|v\|_\infty \le M_{\rm ampl}, \|\dot{v}\|_\infty \le M_{\rm slew}\right\}.$$

We proceed as follows to find a subgradient of ϕ at the transfer matrix H_0. Find a signal v_0 such that

$$\|v_0\|_\infty \le M_{\rm ampl}, \quad \|\dot{v}_0\|_\infty \le M_{\rm slew}, \quad \int_0^\infty v_0(t)h_0(t)\,dt = \phi(H_0).$$

(It can be shown that in this case there always is such a v_0; some methods for finding v_0 are described in the Notes and References for chapter 5.) Then a subgradient of ϕ at H_0 is given by

$$\phi^{\rm sg}(H) = \phi^{v_0}(H).$$

The same procedure works for *any* worst case norm: first, find a worst case input signal u_0 such that $\|H_0\|_{\rm wc} = \|H_0u_0\|_{\rm output}$. This task must usually be done to evaluate $\|H_0\|_{\rm wc}$ anyway. Now find any subgradient of the convex functional $\phi^{u_0}(H) = \|Hu_0\|_{\rm output}$; it will be a subgradient of $\|\cdot\|_{\rm wc}$ at H_0.

13.4.8 Subgradient for the Negative Dual Function

In this section we show how to find a subgradient for $-\psi$ at λ, where ψ is the dual function introduced in section 3.6.2 and discussed in section 6.6. Recall from (6.8) of section 6.6 that $-\psi$ can be expressed as the maximum of a family of linear functions of λ; we can therefore use the maximum tool to find a subgradient.

We start by finding any $H_{\rm ach}$ such that

$$\psi(\lambda) = \lambda_1\phi_1(H_{\rm ach}) + \cdots + \lambda_L\phi_L(H_{\rm ach}).$$

Then a subgradient of $-\psi$ at λ is given by

$$g = -\begin{bmatrix} \phi_1(H_{\rm ach}) \\ \vdots \\ \phi_L(H_{\rm ach}) \end{bmatrix}.$$

13.5 Subgradients on a Finite-Dimensional Subspace

In the previous section we determined subgradients for many of the convex functionals we encountered in chapters 8–10. These subgradients are linear functionals on the infinite-dimensional space of transfer matrices; most numerical computation will be done on finite-dimensional subspaces of transfer matrices (as we will see in chapter 15). In this section we show how the subgradients computed above can be used to calculate subgradients on finite-dimensional subspaces of transfer matrices.

Suppose that we have fixed transfer matrices $H_0, H_1, \ldots, H_N$, and ϕ is some convex functional on transfer matrices. We consider the convex function $\varphi : \mathbf{R}^N \to \mathbf{R}$ given by

$$\varphi(x) = \phi(H_0 + x_1 H_1 + \cdots + x_N H_N).$$

To determine some $g \in \partial\varphi(\tilde{x})$, we find a subgradient of ϕ at the transfer matrix $H_0 + \tilde{x}_1 H_1 + \cdots + \tilde{x}_N H_N$, say, ϕ^{sg}. Then

$$g = \begin{bmatrix} \phi^{\mathrm{sg}}(H_1) \\ \vdots \\ \phi^{\mathrm{sg}}(H_N) \end{bmatrix} \in \partial\varphi(\tilde{x}).$$

Let us give a specific example using our standard plant of section 2.4. Consider the weighted peak tracking error functional of section 11.1.2,

$$\varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta) = \left\| W\left(\alpha H_{13}^{(\mathrm{a})} + \beta H_{13}^{(\mathrm{b})} + (1-\alpha-\beta) H_{13}^{(\mathrm{c})} - 1\right)\right\|_{\mathrm{pk\_gn}},$$

where

$$W = \frac{0.5}{s+0.5},$$
$$H_{13}^{(\mathrm{a})} = \frac{-44.1s^3 + 334s^2 + 1034s + 390}{s^6 + 20s^5 + 155s^4 + 586s^3 + 1115s^2 + 1034s + 390},$$
$$H_{13}^{(\mathrm{b})} = \frac{-220s^3 + 222s^2 + 19015s + 7245}{s^6 + 29.1s^5 + 297s^4 + 1805s^3 + 9882s^2 + 19015s + 7245},$$
$$H_{13}^{(\mathrm{c})} = \frac{-95.1s^3 - 24.5s^2 + 9505s + 2449}{s^6 + 33.9s^5 + 425s^4 + 2588s^3 + 8224s^2 + 9505s + 2449}.$$

$\varphi_{\mathrm{pk\_trk}}$ has the form

$$\varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta) = \|H_0 + \alpha H_1 + \beta H_2\|_{\mathrm{pk\_gn}},$$

where

$$H_0 = W\left(H^{(\mathrm{c})} - 1\right),$$
$$H_1 = W\left(H^{(\mathrm{a})} - H^{(\mathrm{c})}\right),$$
$$H_2 = W\left(H^{(\mathrm{b})} - H^{(\mathrm{c})}\right).$$

The level curves of $\varphi_{\mathrm{pk\_trk}}$ are shown in figure 11.4. A subgradient $g \in \partial\varphi_{\mathrm{pk\_trk}}(\alpha, \beta)$ is given by

$$g = \begin{bmatrix} \phi^{\mathrm{sg}}(H_1) \\ \phi^{\mathrm{sg}}(H_2) \end{bmatrix} = \begin{bmatrix} \displaystyle\int_0^\infty \mathrm{sgn}(h(t))h_1(t)\,dt \\ \displaystyle\int_0^\infty \mathrm{sgn}(h(t))h_2(t)\,dt \end{bmatrix},$$

where h is the impulse response of $H_0 + \alpha H_1 + \beta H_2$, h_1 is the impulse response of H_1, and h_2 is the impulse response of H_2 (see section 13.4.6).

Consider the point $\alpha = 1$, $\beta = 0$, where $\varphi_{\mathrm{pk\_trk}}(1,\ 0) = 0.837$. A subgradient at this point is

$$g = \begin{bmatrix} 0.168 \\ -0.309 \end{bmatrix}. \tag{13.6}$$

In figure 13.6 the level curve $\varphi_{\mathrm{pk\_trk}}(\alpha, \beta) = 0.837$ is shown, together with the subgradient (13.6) at the point $[1\ \ 0]^T$. As expected, the subgradient determines a half-space that contains the convex set

$$\left\{ [\alpha\ \ \beta]^T \;\middle|\; \varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta) \le \varphi_{\mathrm{pk\_trk}}(1,\ 0) = 0.837 \right\}.$$

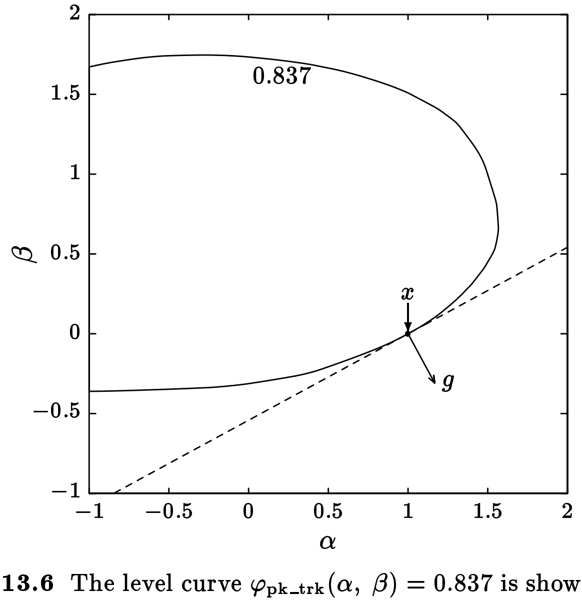


Figure 13.6 The level curve $\varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta) = 0.837$ is shown, together with the subgradient (13.6) at the point $[1\ \ 0]^T$.

Notes and References

Convex Analysis

Rockafellar's book [ROC70] covers convex analysis in detail. Other texts covering this material are Stoer and Witzgall [SW70], Barbu and Precupanu [BP78], Aubin and Vinter [AV82], and Demyanov and Vasilev [DV85]. The last three consider the infinite-dimensional (Banach space) case, carefully distinguishing between the many important different types of continuity, compactness, and so on, which we have not considered. Daniel's book [DAN71] also gives the precise infinite-dimensional formulation.

General nonsmooth analysis (which includes convex analysis) is covered in Clarke [CLA83], which gives the precise formulation of the concepts of this chapter in infinite-dimensional (Banach) spaces. Chapter 2 of Clarke's book contains a complete calculus for subgradients and quasigradients, including the subgradient computation tools of section 13.3 (and a lot more).

Subgradients of Closed-Loop Convex Functionals

Subgradients of an $\mathbf{H}_\infty$ norm are given in Polak and Wardi [PW82]; several of the other subgradients are derived in Polak and Salcudean [PS89] and Salcudean's Ph.D. thesis [SAL86].

Chapter 14

Special Algorithms for Convex Optimization

We describe several simple but powerful algorithms that are specifically designed for convex optimization. These algorithms require only the ability to compute the function value and any subgradient for each relevant function. A key feature of these algorithms is that they maintain converging upper and lower bounds on the quantity being computed, and thus can compute the quantity to a *guaranteed accuracy.* We demonstrate each on the two-parameter example of chapter 11; in chapter 15 they are applied to more substantial problems.

In this chapter we concentrate on the finite-dimensional case; these methods are extended to the infinite-dimensional case in the next chapter.

14.1 Notation and Problem Definitions

We will consider several specific forms of optimization problems.

The *unconstrained problem* is to compute the minimum value of ϕ,

$$\phi^* \stackrel{\Delta}{=} \min \phi(z), \tag{14.1}$$

and in addition to compute a minimizer x^*, which satisfies $\phi(x^*) = \phi^*$. We will use the notation

$$x^* = \arg\min \phi(z)$$

to mean that x^* is some minimizer of ϕ.

The *constrained optimization problem* is to compute the minimum value of ϕ, subject to the constraints $\psi_1(z) \leq 0, \ldots, \psi_m(z) \leq 0$, *i.e.*,

$$\phi^* = \min_{\psi_1(z)\leq 0,\ldots,\psi_m(z)\leq 0} \phi(z),$$

and in addition to compute a minimizer x^* that satisfies $\phi(x^*) = \phi^*$, $\psi_1(x^*) \leq 0$, $\ldots$, $\psi_m(x^*) \leq 0$. To simplify notation we define the *constraint function*

$$\psi(z) = \max_{1 \leq i \leq m} \psi_i(z),$$

so we can express the constraints as $\psi(z) \leq 0$:

$$\phi^* = \min_{\psi(z) \leq 0} \phi(z). \tag{14.2}$$

We will say that z is *feasible* if $\psi(z) \leq 0$.

The *feasibility* problem is:

$$\text{find } x \text{ such that } \psi(x) \leq 0, \text{ or determine that there is no such } x. \tag{14.3}$$

Algorithms that are designed for one form of the convex optimization problem can often be modified or adapted for others; we will see several examples of this. It is also possible to modify the algorithms we present to directly solve other problems that we do not consider, *e.g.*, to find Pareto optimal points or to do goal programming.

Throughout this chapter, unless otherwise stated, ϕ and $\psi_1, \ldots, \psi_m$ are convex functions from $\mathbf{R}^n$ into $\mathbf{R}$. The constraint function ψ is then also convex.

14.2 On Algorithms for Convex Optimization

Many algorithms for convex optimization have been devised, and we could not hope to survey them here. Instead, we give a more detailed description of two types of algorithms that are specifically designed for convex problems: *cutting-plane* and *ellipsoid* algorithms.

Let us briefly mention another large family of algorithms, the *descent methods*, contrasting them with the algorithms that we will describe. In these methods, successive iterations produce points that have decreasing objective values. General-purpose descent methods have been successfully applied to, and adapted for, nondifferentiable convex optimization; see the Notes and References at the end of this chapter. The cutting-plane and ellipsoid algorithms are *not* descent methods; objective values often *increase* after an iteration.

Possible advantages and disadvantages of some of the descent methods over cutting-plane and ellipsoid methods are:

- The cutting-plane and ellipsoid algorithms require only the evaluation of function values and *any one* (of possibly many) subgradients of functions. Most (but not all) descent methods require the computation of a *descent direction* or even steepest descent direction for the function at a point; this can be a difficult task in itself, and is always at least as difficult as computing a subgradient.

- Many (but not all) descent methods for convex optimization retain the heuristic stopping criteria used when they are applied to general (nonconvex) optimization problems. In contrast we will see that the cutting-plane and ellipsoid methods have simple stopping criteria that guarantee the optimum has been found to a known accuracy.

- Most descent methods for nondifferentiable optimization are substantially more complicated than the cutting-plane and ellipsoid algorithms. These methods often include parameters that need to be adjusted for the particular problem.

- For smooth problems, many of the descent methods offer substantially faster convergence, *e.g.*, quadratic.

Let us immediately qualify the foregoing remarks. Many descent methods have worked very well in practice: one famous example is the simplex method for linear programming. In addition, it is neither possible nor profitable to draw a sharp line between descent methods and non-descent methods. For example, the ellipsoid algorithm can be interpreted as a type of variable-metric descent algorithm. We refer the reader to the references at the end of this chapter.

14.3 Cutting-Plane Algorithms

14.3.1 Computing a Lower Bound on ϕ^*

We consider the unconstrained problem (14.1). Suppose we have computed function values and at least one subgradient at $x_1, \ldots, x_k$:

$$\phi(x_1), \ldots, \phi(x_k), \qquad g_1 \in \partial\phi(x_1), \ldots, g_k \in \partial\phi(x_k). \tag{14.4}$$

Each of these function values and subgradients yields an affine lower bound on ϕ:

$$\phi(z) \geq \phi(x_i) + g_i^T(z - x_i) \qquad \text{for all } z,\ 1 \leq i \leq k,$$

and hence

$$\phi(z) \geq \phi_k^{\mathrm{lb}}(z) \stackrel{\Delta}{=} \max_{1 \leq i \leq k} \left(\phi(x_i) + g_i^T(z - x_i)\right). \tag{14.5}$$

ϕ_k^{lb} is a piecewise linear convex function that is everywhere less than or equal to ϕ. Moreover, this global lower bound function is tight at the points $x_1, \ldots, x_k$, since $\phi(x_i) = \phi_k^{\mathrm{lb}}(x_i)$ for $1 \leq i \leq k$. In fact, ϕ_k^{lb} is the smallest convex function that has the function values and subgradients given in (14.4). An example is shown in figure 14.1.

It follows that

$$\phi^* \geq L_k \stackrel{\Delta}{=} \min_z \phi_k^{\mathrm{lb}}(z). \tag{14.6}$$

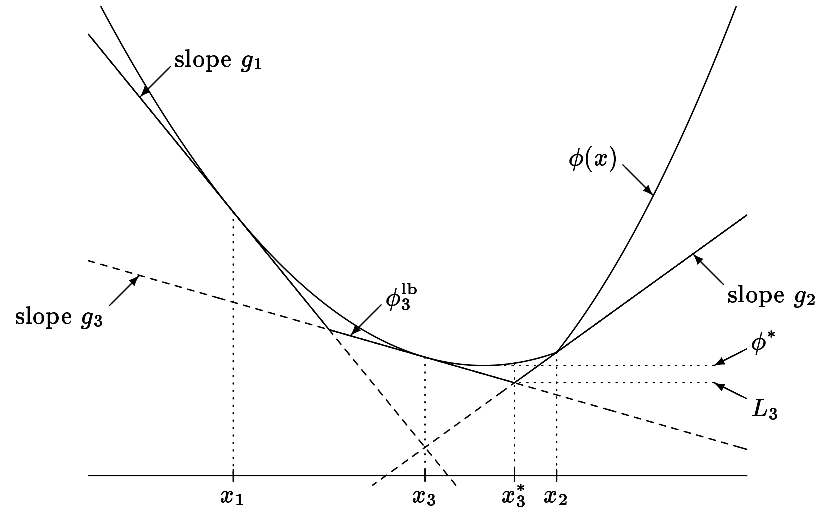


Figure 14.1 The maximum of the affine support functionals at x_1, x_2, and x_3 gives the global lower bound function ϕ_3^{lb}, which is tight at the points x_1, x_2, and x_3. L_3, the minimum of ϕ_3^{lb}, which occurs at x_3^*, is found by solving (14.7). L_3 is a lower bound on ϕ^*.

The minimization problem on the right-hand side is readily solved via linear programming. We can express (14.6) as

$$L_k = \min_{\substack{L,\ z \\ \phi(x_i) + g_i^T(z - x_i) \le L,\ 1 \le i \le k}} L,$$

which has the form of a linear program in the variable w:

$$L_k = \min_{Aw \le b} c^T w \tag{14.7}$$

where

$$w = \begin{bmatrix} z \\ L \end{bmatrix},\ c = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ A = \begin{bmatrix} g_1^T & -1 \\ \vdots & \vdots \\ g_k^T & -1 \end{bmatrix},\ b = \begin{bmatrix} g_1^T x_1 - \phi(x_1) \\ \vdots \\ g_k^T x_k - \phi(x_k) \end{bmatrix}. \tag{14.8}$$

In fact, this linear program determines not only L_k, but also a minimizer of ϕ_k^{lb}, which we denote x_k^* (shown in figure 14.1 as well).

The idea behind (14.6) is elementary, but it is very important. It shows that by knowing only a finite number of function values and subgradients of a convex function, we can deduce a lower bound on the minimum of the function. No such property holds for general nonconvex functions. This property will be useful in designing stopping criteria for optimization algorithms that can guarantee a maximum error.

The function ϕ_k^{lb} can be unbounded below, so that $L_k = -\infty$ (*e.g.* when $k = 1$ and $g_1 \neq 0$), which is not a useful bound. This can be avoided by explicitly specifying bounds on the variables, so that we consider

$$\phi^* = \min_{z_{\min} \le z \le z_{\max}} \phi(z).$$

In this case we have the lower bound

$$L_k = \min_{z_{\min} \le z \le z_{\max}} \phi_k^{\mathrm{lb}}(z)$$

which can be computed by adding the bound inequalities $z_{\min} \le z \le z_{\max}$ to the linear program (14.7).

Finally we mention that having computed ϕ at the points $x_1, \ldots, x_k$ we have the simple upper bound on ϕ^*:

$$U_k \stackrel{\Delta}{=} \min_{1 \le i \le k} \phi(x_i), \qquad (14.9)$$

which is the lowest objective value so far encountered. If an objective value and a subgradient are evaluated at another point x_{k+1}, the new lower and upper bounds L_{k+1} and U_{k+1} are improved:

$$L_k \le L_{k+1} \le \phi^* \le U_{k+1} \le U_k.$$

14.3.2 Kelley's Cutting-Plane Algorithm

Kelley's cutting-plane algorithm is a natural extension of the lower bound computation of the previous section. It is simply:

```
$x_1$, $z_{\min}$, $z_{\max}$ ← any initial box that contains minimizers;
$k$ ← 0;
repeat {
        $k$ ← $k + 1$;
        compute $\phi(x_k)$ and any $g_k \in \partial\phi(x_k)$;
        solve (14.7) to find $x_k^*$ and $L_k$;
        compute $U_k$ using (14.9);
        $x_{k+1}$ ← $x_k^*$;
} until ( $U_k - L_k \le \epsilon$ );
```

An example of the second iteration of the cutting-plane algorithm is shown in figure 14.2. The idea behind Kelley's algorithm is that at each iteration the lower bound function ϕ_k^{lb} is refined or improved (*i.e.*, made larger, so that $\phi_{k+1}^{\mathrm{lb}} \ge \phi_k^{\mathrm{lb}}$),

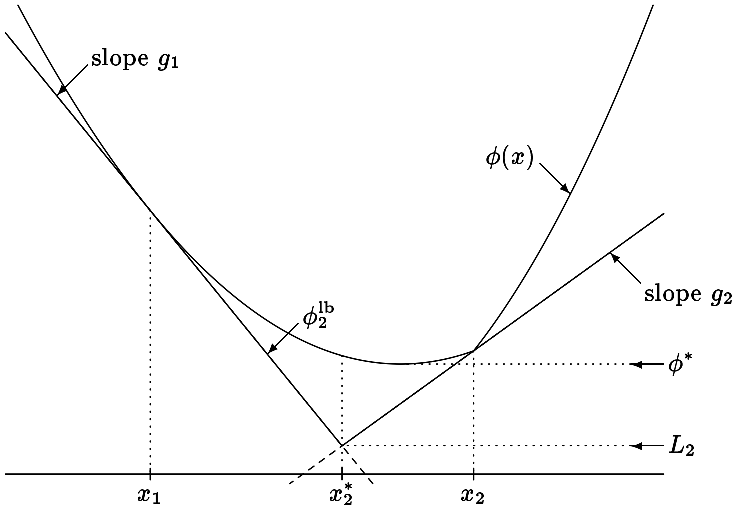


Figure 14.2 In the second iteration of the cutting-plane algorithm a subgradient g_2 at x_2 is found, giving the global lower bound function $\phi_2^{\rm lb}$. (14.7) is solved to give L_2 and $x_3 = x_2^*$. (The next iteration of the cutting-plane algorithm is shown in figure 14.1.)

since one more term is added to the maximum in (14.5). Moreover, since $\phi_k^{\rm lb}$ is *tight* at $x_1, \ldots, x_k$, it is a good approximation to ϕ near $x_1, \ldots, x_k$. In the next section, we will use this idea to show that the cutting-plane algorithm always terminates.

The cutting-plane algorithm maintains upper and lower bounds on the quantity being computed:

$$L_k \le \phi^* \le U_k,$$

which moreover converge as the algorithm proceeds:

$$U_k - L_k \to 0 \quad \text{as } k \to 0.$$

Thus we compute ϕ^* to a *guaranteed accuracy* of ϵ: on exit, we have a point with a low function value, and in addition we have a *proof* that there are no points with function value more than ϵ better than that of our point. Stopping criteria for general optimization algorithms (*e.g.*, descent methods) are often more heuristic—they cannot guarantee that on exit, ϕ^* has been computed to a given accuracy.

Provided $\phi^* \neq 0$, it is also possible to specify a maximum relative error (as opposed to absolute error), with the modified stopping criterion

until ($U_k - L_k \le \epsilon \min\{|L_k|,\ |U_k|\}$);

which guarantees a relative accuracy of at least ϵ on exit. These stopping criteria can offer a great advantage when the accuracy required is relatively low, for example, 10%. This relative accuracy can be achieved long before the objective values or iterates x_k appear to be converging; still, we can confidently halt the algorithm.

A valid criticism of the cutting-plane algorithm is that the number of constraints in the linear program (14.7) that must be solved at each iteration grows with the total number of elapsed iterations. In practice, if these linear programs are initialized at the previous point, they can be solved very rapidly, *e.g.*, in a few simplex iterations. Some cutting-plane algorithms developed since Kelley's drop constraints, so the size of the linear programs to be solved does not grow as the algorithm proceeds; see the Notes and References at end of this chapter.

While the cutting-plane algorithm makes use of all of the information ($\phi(x_i)$, g_i, $i = 1, \ldots, k$) that we have obtained about ϕ in previous iterations, the ellipsoid methods that we will describe later in this chapter maintain a data structure of constant size (at the cost of an approximation) that describes what we have learned about the function in past iterations.

14.3.3 Proof of Convergence

We noted above that when the cutting-plane algorithm terminates, we know that the optimum objective ϕ^* lies between the bounds L and U, which differ by less than ϵ. In this section we show that the cutting-plane algorithm does in fact always terminate.

Let B_{init} denote the initial box, and

$$G = \sup_{\substack{g \in \partial\phi(z) \\ z \in B_{\text{init}}}} \|g\|.$$

(G can be shown to be finite.) Suppose that for $k = 1, \ldots, K$ the algorithm has not terminated, *i.e.* $U_k - L_k > \epsilon$ for $k = 1, \ldots, K$. Since

$$L_k = \phi_k^{\text{lb}}(x_{k+1}) = \max_{j \le k} \left(\phi(x_j) + g_j^T(x_{k+1} - x_j)\right)$$

we have

$$L_k \ge \phi(x_j) + g_j^T(x_{k+1} - x_j), \qquad 1 \le j \le k \le K,$$

and hence using $\phi(x_j) \ge U_k$ and the Cauchy-Schwarz inequality

$$L_k \ge U_k - G\|x_{k+1} - x_j\|, \qquad 1 \le j \le k \le K.$$

From this and $U_k - L_k > \epsilon$ for $k \le K$ we conclude that

$$\|x_i - x_j\| > \frac{\epsilon}{G} \qquad i,\ j \le K,\ i \ne j, \tag{14.10}$$

in other words, the minimum distance between any two of the points $x_1, \ldots, x_k$ exceeds ϵ/G.

A volume argument can now be used to show that K cannot be too large. Around each x_i we place a ball B_i of diameter ϵ/G. By (14.10), these balls do not intersect, so their total volume is K times the volume of one ball. These balls are all contained in a box B which is the original box $B_{\rm init}$ enlarged in every dimension by $\epsilon/2G$. Hence the total volume of the balls must be less than the volume of B. We conclude that K is no larger than the volume of B divided by the volume of one of the balls.

If we take $K_{\max} = \mathbf{vol}(B)/\mathbf{vol}(B_1)$, then within $K_{\max}$ iterations, the cutting-plane algorithm terminates. (We comment that this upper bound is a very poor bound, vastly greater than the typical number of iterations required.)

14.3.4 Cutting-Plane Algorithm with Constraints

The cutting-plane algorithm of the previous section can be modified in many ways to handle the constrained optimization problem (14.2). We will show one simple method, which uses the same basic idea of forming a piecewise linear lower bound approximation of a convex function based on the function values and subgradients already evaluated.

Suppose we have computed function values and at least one subgradient at $x_1, \ldots, x_k$ for both the objective and the constraint function:

$$\begin{array}{ll} \phi(x_1), \ldots, \phi(x_k), & g_1 \in \partial\phi(x_1), \ldots, g_k \in \partial\phi(x_k), \\ \psi(x_1), \ldots, \psi(x_k), & h_1 \in \partial\psi(x_1), \ldots, h_k \in \partial\psi(x_k). \end{array}$$

These points x_i need not be feasible.

We form piecewise linear lower bound functions for both the objective and the constraint: $\phi_k^{\rm lb}$ in (14.5) and

$$\psi_k^{\rm lb}(z) \stackrel{\Delta}{=} \max_{1 \le i \le k} \left(\psi(x_i) + h_i^T (z - x_i) \right), \tag{14.11}$$

which satisfies $\psi_k^{\rm lb}(z) \le \psi(z)$ for all $z \in \mathbf{R}^n$.

The lower bound function $\psi_k^{\rm lb}$ yields a polyhedral *outer approximation* to the feasible set:

$$\{ z \mid \psi(z) \le 0 \} \subseteq \left\{ z \;\middle|\; \psi_k^{\rm lb}(z) \le 0 \right\}. \tag{14.12}$$

Thus we have the following lower bound on ϕ^*:

$$\phi^* \ge L_k \stackrel{\Delta}{=} \min \left\{ \phi_k^{\rm lb}(z) \;\middle|\; \psi_k^{\rm lb}(z) \le 0 \right\}. \tag{14.13}$$

As in section 14.3.1, the optimization problem (14.13) is equivalent to a linear program:

$$L_k = \min_{Aw \le b} \; c^T w \tag{14.14}$$

where

$$w=\left[\begin{array}{c} z \\ L \end{array}\right], \quad c=\left[\begin{array}{c} 0 \\ 1 \end{array}\right], \quad A=\left[\begin{array}{cc} g_1^T & -1 \\ \vdots & \vdots \\ g_k^T & -1 \\ h_1^T & 0 \\ \vdots & \vdots \\ h_k^T & 0 \end{array}\right], \quad b=\left[\begin{array}{c} g_1^T x_1-\phi(x_1) \\ \vdots \\ g_k^T x_k-\phi(x_k) \\ h_1^T x_1-\psi(x_1) \\ \vdots \\ h_k^T x_k-\psi(x_k) \end{array}\right].$$

We note again that the lower bound L_k in (14.13) can be computed no matter how the points and subgradients were chosen.

The modified cutting-plane algorithm is exactly the same as the cutting-plane algorithm, except that the linear program (14.14) is solved instead of (14.8), and the stopping criterion must be modified for reasons that we now consider.

If the feasible set is empty then eventually the linear program (14.14) will become infeasible. When this occurs, the cutting-plane algorithm terminates with the conclusion that the feasible set is empty. On the other hand, if the feasible set is not empty, and the optimum occurs on the boundary of the feasible set, which is often the case, then the iterates x_k are generally *infeasible*, since they lie in the outer approximation of the feasible set given by the right-hand side of (14.12). However, they *approach* feasibility (and optimality): it can be shown that

$$\lim_{k\to\infty}\phi(x_k)=\phi^*, \qquad \limsup_{k\to\infty}\psi(x_k)\le 0.$$

The second inequality means that given any $\epsilon_{\mathrm{feas}}>0$ we eventually have $\psi(x_k)<\epsilon_{\mathrm{feas}}$.

This has an important consequence for the upper bound and stopping criterion described in section 14.3.2. If x_k is not feasible, then $\phi(x_k)$ is not an upper bound on ϕ^*. Therefore we cannot use (14.9) to compute an upper bound on ϕ^*. A good stopping criterion for the modified cutting-plane algorithm is:

```
until ( $\psi(x_k)\le\epsilon_{\mathrm{feas}}$ and $\phi(x_k)-L_k\le\epsilon_{\mathrm{obj}}$ );
```

We interpret ϵ_{feas} as a *feasibility tolerance* and ϵ_{obj} as an *objective tolerance.* When the algorithm successfully terminates we are guaranteed to have found a point that is feasible and has objective value within ϵ_{obj} of optimal for the ϵ_{feas}-*relaxed problem*

$$\min\{\phi(z)\mid\psi(z)\le\epsilon_{\mathrm{feas}}\},$$

(but probably not feasible for the problem (14.2).)

Other modifications of the cutting-plane algorithm generate only feasible iterates; see the Notes and References at the end of this chapter.

14.3.5 Example

In this example we will use the cutting-plane algorithm to minimize a convex function on $\mathbf{R}^2$. We will use the standard example plant that we introduced in section 2.4.

The function to be minimized is

$$\phi\left(\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\right) = \varphi_{\rm wt\_max}(\alpha,\ \beta),$$

where the function

$$\varphi_{\rm wt\_max}(\alpha,\beta) = \max\left\{\varphi_{\rm pk\_trk}(\alpha,\beta),\ 0.5\varphi_{\rm max\_sens}(\alpha,\beta),\ 15\varphi_{\rm rms\_yp}(\alpha,\ \beta)\right\}$$

was defined in section 11.7. The level curves of $\varphi_{\rm wt\_max}$ are plotted in figure 11.23. The minimum value of $\varphi_{\rm wt\_max}$ is 0.737, which occurs at $x^* = [0.05\ \ 0.09]^T$.

The bounding box for the cutting-plane algorithm was $z_{\rm min} = [-0.9\ \ -0.9]^T$ and $z_{\rm max} = [1.9\ \ 1.9]^T$. This was an aesthetic choice; in a real problem the bounding box would be much larger. The starting point, $x_1 = [0.5\ \ 0.5]^T$, is the center of the bounding box. The upper and lower bounds versus iteration number are shown in figure 14.3. The maximum relative error, *i.e.*, $(U_k - L_k)/L_k$, is shown in figure 14.4. Level curves of $\phi_k^{\rm lb}$ are shown for $k = 1,\ 2,\ 3,\ 4$ in figures 14.5, 14.6, 14.7, and 14.8. The reader is encouraged to trace the execution of the algorithm through these figures, and compare the level curves of $\phi_k^{\rm lb}$ with those of $\varphi_{\rm wt\_max}$ in figure 11.23.

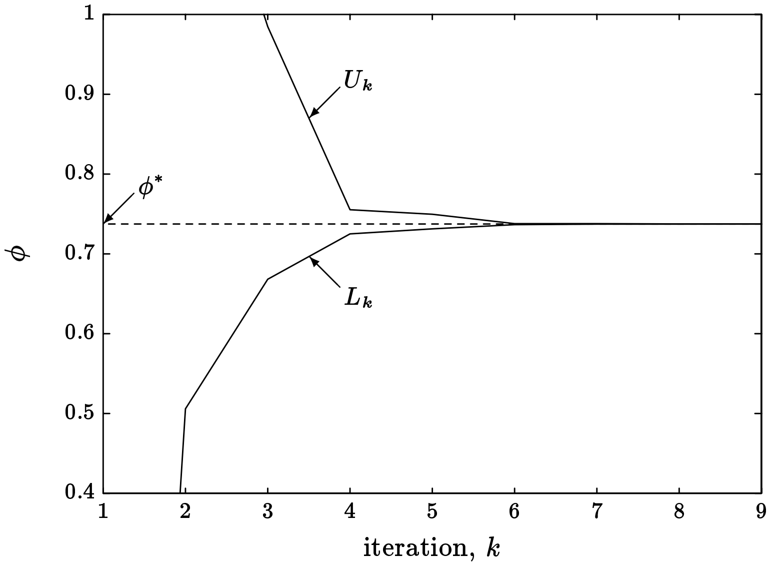


Figure 14.3 Upper and lower bounds on the solution ϕ^*, as a function of the iteration number k, are shown for the cutting-plane algorithm.

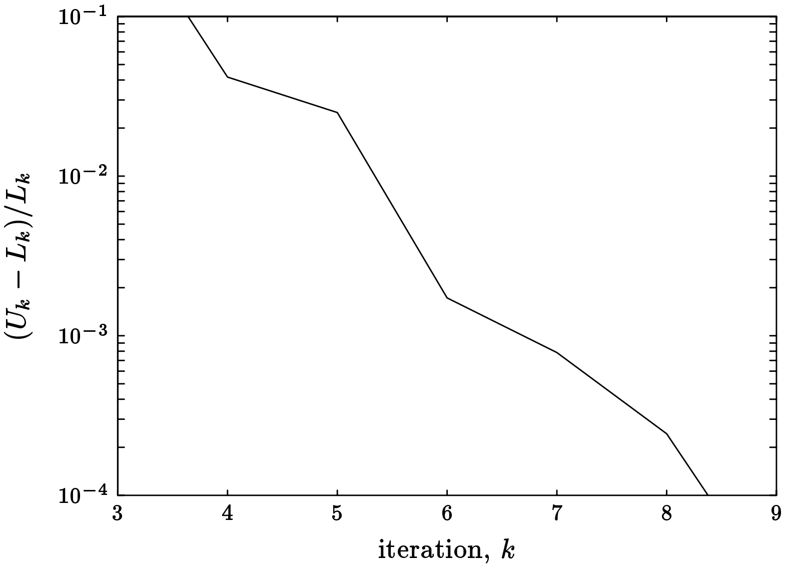


Figure 14.4 For the cutting-plane algorithm the maximum relative error, defined as $(U_k - L_k)/L_k$, falls below 0.01% by iteration 9.

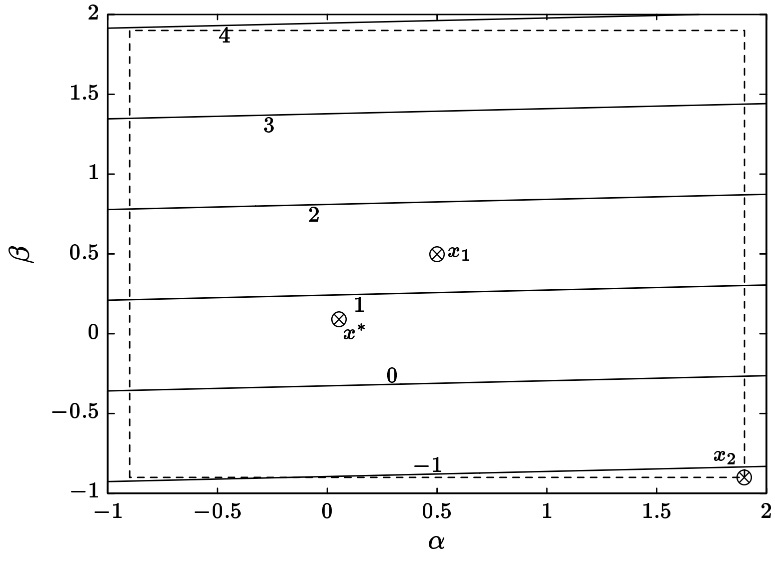


Figure 14.5 The cutting-plane algorithm is started at the point $x_1 = [0.5 \ \ 0.5]^T$. The subgradient g_1 at x_1 gives an affine global lower bound $\phi_1^{\rm lb}$ for ϕ. The level curves of $\phi_1^{\rm lb}$ are shown, together with the solution x_2 to (14.7). This point gives the lower bound $L_1 = -1.12$. (The dashed line shows the bounding box, and x^* is the minimizer of $\varphi_{\rm wt\_max}$.)

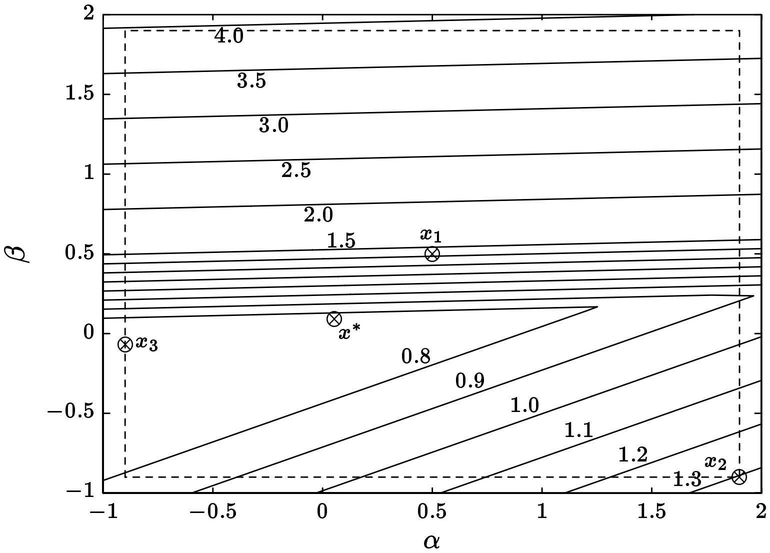


Figure 14.6 The level curves of ϕ_2^{lb} are shown, together with the solution x_3 to (14.7). This point gives the lower bound $L_2 = 0.506$.

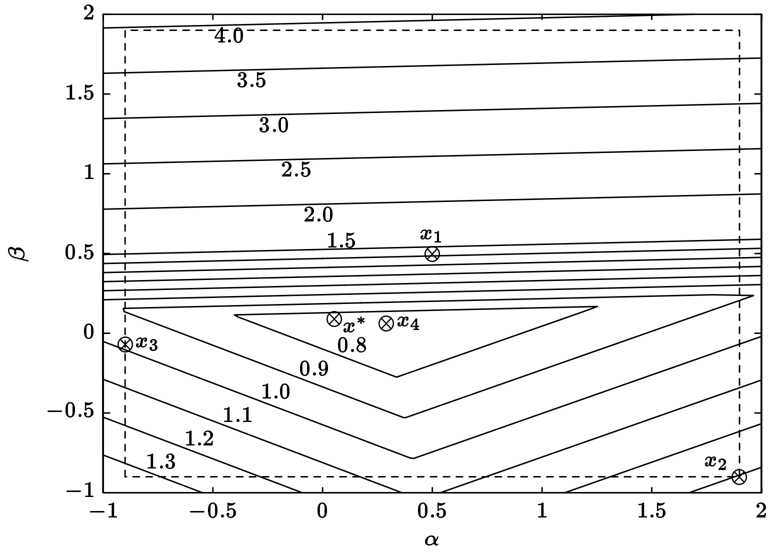


Figure 14.7 The level curves of ϕ_3^{lb} are shown, together with the solution x_4 to (14.7). This point gives the lower bound $L_3 = 0.668$.

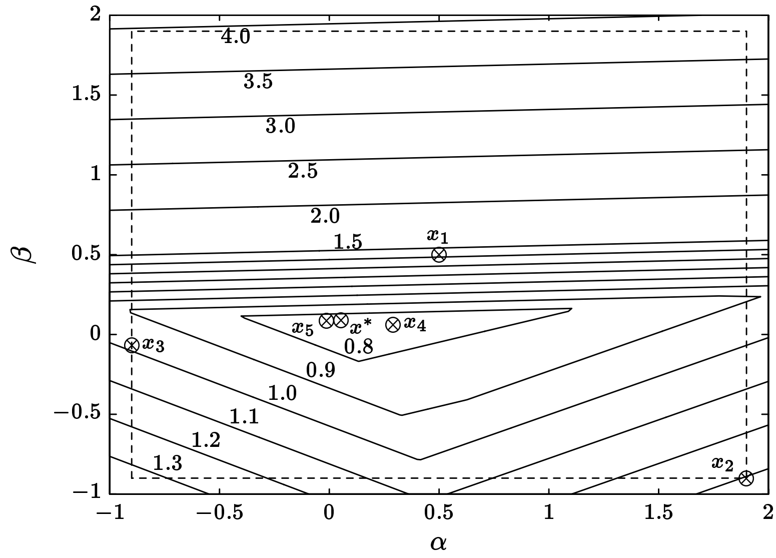


Figure 14.8 The level curves of ϕ_4^{lb} are shown, together with the solution x_5 to (14.7). This point gives the lower bound $L_4 = 0.725$.

14.4 Ellipsoid Algorithms

14.4.1 Basic Ellipsoid Algorithm

We first consider the unconstrained minimization problem (14.1). The ellipsoid algorithm generates a "decreasing" sequence of ellipsoids in $\mathbf{R}^n$ that are guaranteed to contain a minimizing point, using the idea that given a subgradient (or quasigradient) at a point, we can find a half-space containing the point that is guaranteed not to contain any minimizers of ϕ (see figure 13.2).

Suppose that we have an ellipsoid E_k that is guaranteed to contain a minimizer of ϕ. In the basic ellipsoid algorithm, we compute a subgradient g_k of ϕ at the center, x_k, of E_k. We then know that the "sliced" half ellipsoid

$$E_k \cap \left\{ z \;\middle|\; g_k^T(z - x_k) \le 0 \right\}$$

contains a minimizer of ϕ, as shown in figure 14.9.

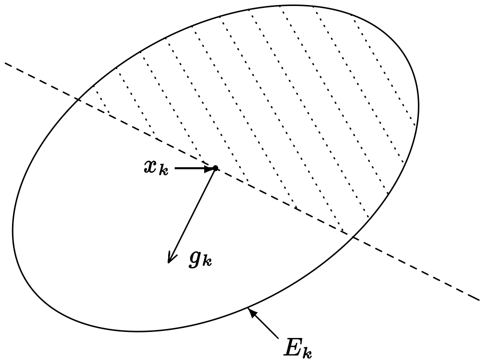


Figure 14.9 At the kth iteration of the ellipsoid algorithm a minimizer of ϕ is known to lie in the ellipsoid E_k centered at x_k. The subgradient g_k at x_k determines a half-space (below and to the left of the dashed line) in which $\phi(x)$ is at least $\phi(x_k)$. Therefore a minimizer of ϕ lies in the shaded region.

We compute the ellipsoid E_{k+1} of *minimum volume* that contains the sliced half ellipsoid; E_{k+1} is then guaranteed to contain a minimizer of ϕ, as shown in figure 14.10. The process is then repeated.

We now describe the algorithm more explicitly. An ellipsoid E can be described as

$$E = \left\{ z \;\middle|\; (z - a)^T A^{-1} (z - a) \le 1 \right\}$$

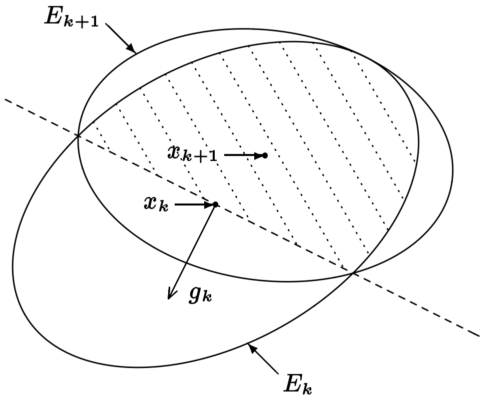


Figure 14.10 The shaded region in figure 14.9 is enclosed by the ellipsoid of smallest volume, denoted E_{k+1}, and centered at x_{k+1}. Any subgradient g_{k+1} at x_{k+1} is found and the next iteration of the ellipsoid algorithm follows. In $\mathbf{R}^2$ the area of E_{k+1} is always 77% of the area of E_k.

where $A = A^T > 0$. a is the center of the ellipsoid E and the matrix A gives the "size" and orientation of E: the square roots of the eigenvalues of A are the lengths of the semi-axes of E. The volume of E is given by

$$\mathbf{vol}(E) = \beta_n \sqrt{\det A},$$

where β_n is the volume of the unit sphere in $\mathbf{R}^n$; in fact,

$$\beta_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)},$$

but we will not need this result.

The minimum volume ellipsoid that contains the half ellipsoid

$$\left\{z \mid (z-a)^T A^{-1}(z-a) \le 1,\ g^T(z-a) \le 0\right\}$$

is given by

$$\tilde{E} = \left\{z \;\middle|\; (z-\tilde{a})^T \tilde{A}^{-1}(z-\tilde{a}) \le 1\right\},$$

where

$$\tilde{a} = a - \frac{A\tilde{g}}{n+1}, \tag{14.15}$$

$$\tilde{A} = \frac{n^2}{n^2-1}\left(A - \frac{2}{n+1}A\tilde{g}\tilde{g}^T A\right), \tag{14.16}$$

and

$$\tilde{g} = g \Big/ \sqrt{g^T A g}$$

is a normalized subgradient. (Note that the sliced ellipsoid depends only on the direction of g, and not its length.)

Thus, the basic ellipsoid algorithm is:

```
$x_1$, $A_1 \leftarrow$ any initial ellipsoid that contains minimizers;
$k \leftarrow 0$;
repeat {
        $k \leftarrow k+1$;
        evaluate $\phi(x_k)$ and compute any $g_k \in \partial\phi(x_k)$;
        $\tilde{g} \leftarrow g_k \Big/ \sqrt{g_k^T A_k g_k}$ ;
        $x_{k+1} \leftarrow x_k - A_k \tilde{g}/(n+1)$;
        $A_{k+1} \leftarrow \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1} A_k \tilde{g}\tilde{g}^T A_k\right)$;
} until ( stopping criterion );
```

From (14.16), we can think of E_{k+1} as slightly thinner than E_k in the direction g_k, and slightly enlarged over all. Even though E_{k+1} can be *larger* than E_k in the sense of maximum semi-axis ($\lambda_{\max}(A_{k+1}) > \lambda_{\max}(A_k)$ is possible), it turns out that its volume is less:

$$\begin{aligned} \mathbf{vol}(E_{k+1}) &= \left(\frac{n}{n+1}\right)^{(n+1)/2} \left(\frac{n}{n-1}\right)^{(n-1)/2} \mathbf{vol}(E_k) && (14.17) \\ &< e^{-\frac{1}{2n}}\, \mathbf{vol}(E_k), && (14.18) \end{aligned}$$

by a factor that only depends on the dimension n. We will use this fact in the next section to show that the ellipsoid algorithm converges, *i.e.*,

$$\lim_{k\to\infty} \phi(x_k) = \phi^*,$$

provided our original ellipsoid E_1 (which is often a large ball, *i.e.* $x_1 = 0$, $A_1 = R^2 I$) contains a minimizing point in its interior.

Since we always know that there is a minimizer $z^* \in E_k$, we have

$$\phi^* = \phi(z^*) \geq \phi(x_k) + g_k^T(z^* - x_k)$$

for some $z^* \in E_k$, and hence

$$\begin{aligned} \phi(x_k) - \phi^* &\leq -g_k^T(z^* - x_k) \\ &\leq \max_{z\in E_k} -g_k^T(z - x_k) \\ &= \sqrt{g_k^T A_k g_k}. \end{aligned}$$

Thus the simple stopping criterion

until $\left(\sqrt{g_k^T A_k g_k} \le \epsilon \right)$;

guarantees that on exit, $\phi(x_k)$ is within ϵ of ϕ^*. A more sophisticated stopping criterion is

until ($U_k - L_k \le \epsilon$);

where

$$U_k = \min_{1 \le i \le k} \phi(x_i), \quad L_k = \max_{1 \le i \le k} \left(\phi(x_i) - \sqrt{g_i^T A_i g_i} \right). \tag{14.19}$$

While the ellipsoid algorithm works for quasiconvex functions (with the g_k's quasigradients), this stopping criterion does not.

14.4.2 Proof of Convergence

In this section we show that the ellipsoid algorithm converges. We suppose that $z^* \in E_1$ and that for $k = 1, \ldots, K$, $\phi(x_k) > \phi^* + \epsilon$, where $\epsilon > 0$. Then every point z excluded in iterations $1, \ldots, K$ has $\phi(z) > \phi^* + \epsilon$, since at iteration k the function values in each excluded half-space exceed $\phi(x_k)$. If

$$G = \max_{\substack{g \in \partial\phi(x) \\ x \in E_1}} \|g\|$$

is the maximum length of the subgradients over the initial ellipsoid, then we find that in the ball

$$B = \{z \mid \|z - z^*\| \le \epsilon/G\}$$

we have $\phi(z) \le \phi^* + \epsilon$ (we assume without loss of generality that $B \subseteq E_1$), and consequently no point of B was excluded in iterations $1, \ldots, K$, so that in fact

$$B \subseteq E_k.$$

Thus, $\mathbf{vol}(E_k) \ge \mathbf{vol}(B)$, so using (14.17–14.18),

$$e^{-\frac{K}{2n}} \mathbf{vol}(E_1) \ge (\epsilon/G)^n \beta_n.$$

For $E_1 = \{z \mid \|z\| \le R\}$ we have $\mathbf{vol}(E_1) = R^n \beta_n$, so, taking logs,

$$-\frac{K}{2n} + n \log R \ge n \log \frac{\epsilon}{G},$$

and therefore

$$K \le 2n^2 \log \frac{RG}{\epsilon}.$$

Thus to compute ϕ^* with error at most ϵ, it takes no more than $2n^2 \log RG/\epsilon$ iterations of the ellipsoid algorithm; this number grows slowly with both dimension n and accuracy ϵ. We will return to this important point in section 14.6.

14.4.3 Ellipsoid Algorithm with Constraints

The basic ellipsoid algorithm is readily modified to solve the constrained problem (14.2). In this section we describe one such modification.

Once again, we generate a sequence of ellipsoids of decreasing volume, each of which is guaranteed to contain a feasible minimizer. If x_k is feasible ($\psi(x_k) \leq 0$) then we form E_{k+1} exactly as in the basic ellipsoid algorithm; we call this an *objective* iteration. If x_k is infeasible ($\psi(x_k) > 0$) then we form E_{k+1} as in the basic ellipsoid algorithm, but using a subgradient of the constraint instead of the objective. We call this a *constraint* iteration.

The algorithm is thus:

```
$x_1,\ A_1 \leftarrow$ an ellipsoid that contains feasible minimizers (if there are any);
$k \leftarrow 0$;
repeat {
        $k \leftarrow k+1$;
        compute $\psi(x_k)$;
        if ( $\psi(x_k) > 0$ ) {
                /* $x_k$ is infeasible */
                compute any $h_k \in \partial\psi(x_k)$;
                $\tilde{g} \leftarrow h_k \Big/ \sqrt{h_k^T A_k h_k}$ ;
                if $\left( \psi(x_k) - \sqrt{h_k^T A_k h_k} > 0 \right)$ {
                        quit because the feasible set is empty;
                }
        } else {
                /* $x_k$ is feasible */
                compute $\phi(x_k)$ and any $g_k \in \partial\phi(x_k)$;
                $\tilde{g} \leftarrow g_k \Big/ \sqrt{g_k^T A_k g_k}$;
        }
        $x_{k+1} \leftarrow x_k - A_k\tilde{g}/(n+1)$;
        $A_{k+1} \leftarrow \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1}A_k\tilde{g}\tilde{g}^T A_k\right)$;
} until $\left( \psi(x_k) \leq 0 \text{ and } \sqrt{g_k^T A_k g_k} \leq \epsilon \right)$;
```

In a constraint iteration, the points we discard are all infeasible. In an objective iteration, the points we discard all have objective value greater than or equal to the current, feasible point. Thus in each case, we do not discard any minimizers, so that the ellipsoids will always contain any minimizers that are in the initial ellipsoid.

The same proof as for the basic ellipsoid algorithm shows that this modified algorithm works provided the set of points that are feasible and have nearly optimal objective value has positive volume. More sophisticated variations work in other cases (*e.g.*, equality constraints). Alternatively, if we allow slightly violated constraints, we can use the stopping criterion

$$\textit{until}\ \left(\ \psi(x_k) \le \epsilon_{\rm feas}\ \textit{and}\ \sqrt{g_k^T A_k g_k} \le \epsilon_{\rm obj}\ \right);$$

With this stopping criterion, the modified algorithm will work even when the set of feasible points with nearly optimal objective value does not have positive volume, *e.g.*, with equality constraints. In this case the modified algorithm produces nearly optimal points for the ϵ-relaxed problem, just as in the modified cutting-plane algorithm.

14.4.4 Deep-Cut Ellipsoid Algorithm for Inequality Specifications

A simple variation of the ellipsoid algorithm can be used to solve the feasibility problem (14.3). This modified algorithm often performs better than the ellipsoid algorithm applied to the constraint function ψ.

The idea is based on figure 13.3: suppose we are given an x that is not feasible, so that $\psi(x) > 0$. Given $h \in \partial\psi(x)$, we have for all z

$$\psi(z) \ge \psi(x) + h^T(z - x)$$

so for every feasible point $z_{\rm feas}$ we have

$$h^T(z_{\rm feas} - x) \le -\psi(x).$$

We can therefore exclude from consideration the half-space

$$\left\{ z \ \middle|\ h^T(z - x) > -\psi(x) \right\}$$

which is bigger ($\psi(x) > 0$) than the half-space $\{z \mid h^T(z - x) > 0\}$ excluded in the ellipsoid algorithm, as shown in figure 14.11.

In the modified ellipsoid algorithm, we maintain ellipsoids guaranteed to contain a feasible point (if there is one), as shown in figures 14.11 and 14.12. If x_k is not feasible ($\psi(x_k) > 0$), we let E_{k+1} be the minimum volume ellipsoid that contains the set

$$S_k = E_k \cap \left\{ z \ \middle|\ h_k^T(z - x_k) \le -\psi(x_k) \right\}.$$

If $S_k = \emptyset$, we know that there are no feasible points. This happens if and only if

$$\sqrt{h_k^T A_k h_k} < \psi(x_k).$$

Otherwise, E_{k+1} is given by

$$x_{k+1} = x_k - \frac{1 + n\alpha}{n + 1} A_k \tilde{h}_k \tag{14.20}$$

$$A_{k+1} = \frac{n^2}{n^2 - 1}(1 - \alpha^2)\left(A_k - \frac{2(1 + n\alpha)}{(n + 1)(1 + \alpha)} A_k \tilde{h}_k \tilde{h}_k^T A_k \right), \tag{14.21}$$

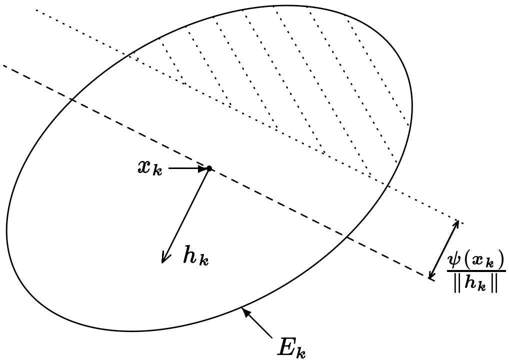


Figure 14.11 At the kth iteration of the deep-cut ellipsoid algorithm any feasible point must lie in the ellipsoid E_k centered at x_k. If x_k is infeasible (meaning $\psi(x_k) > 0$) any subgradient h_k of ψ at x_k determines a half-space (below and to the left of the dotted line) in which $\psi(x)$ is known to be positive. Thus any feasible point is now known to lie in the shaded region.

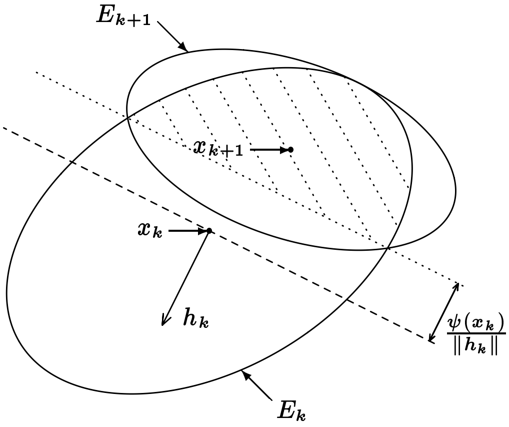


Figure 14.12 The shaded region in figure 14.11 is enclosed by the ellipsoid of smallest volume, denoted E_{k+1}, and centered at x_{k+1}. If the point x_{k+1} is feasible the algorithm terminates. Otherwise, any subgradient h_{k+1} of ψ at x_{k+1} is found. The algorithm then proceeds as shown in figure 14.11.

where

$$\alpha = \frac{\psi(x_k)}{\sqrt{h_k^T A_k h_k}},$$

$$\tilde{h}_k = \frac{h_k}{\sqrt{h_k^T A_k h_k}}.$$

In forming E_{k+1}, we cut out more of the ellipsoid E_k than in the basic ellipsoid method, so this algorithm is called a *deep-cut* ellipsoid method. The deep-cut ellipsoid algorithm is thus:

```
$x_1$, $A_1 \leftarrow$ any ellipsoid that contains feasible points (if there are any);
$k \leftarrow 1$;
repeat {
        compute $\psi(x_k)$ and any $h_k \in \partial\psi(x_k)$;
        if ( $\psi(x_k) \le 0$ ) {
                done: $x_k$ is feasible;
        }
        if $\left( \psi(x_k) > \sqrt{h_k^T A_k h_k} \right)$ {
                quit: the feasible set is empty;
        }
        $\alpha \leftarrow \psi(x_k) \Big/ \sqrt{h_k^T A_k h_k}$;
        $\tilde{h}_k \leftarrow h_k \Big/ \sqrt{h_k^T A_k h_k}$;
        $x_{k+1} \leftarrow x_k - \frac{1+n\alpha}{n+1} A_k \tilde{h}_k$;
        $A_{k+1} \leftarrow \frac{n^2}{n^2-1}(1-\alpha^2)\left(A_k - \frac{2(1+n\alpha)}{(n+1)(1+\alpha)} A_k \tilde{h}_k \tilde{h}_k^T A_k\right)$;
        $k \leftarrow k+1$;
}
```

Deep-cuts can be used for the constraint iterations in the modified ellipsoid algorithm for the constrained problem; they can also be used for the objective iterations in the ellipsoid algorithm for the unconstrained or constrained problems, as follows. At iteration k, it is known that the optimum function value does not exceed U_k, so we can cut out the half-space in which the function must exceed U_k, which is often larger (if $U_k < \phi(x_k)$) than the half-space in which the function must exceed $\phi(x_k)$.

14.4.5 Example

We use the same function, $\varphi_{\mathrm{wt\_max}}$, that we used in the cutting-plane example in section 14.3.5. The algorithm was started with E_1 set to a circle of radius 3/2 about $x_1 = [0.5 \;\; 0.5]^T$. Ellipsoid volume is shown in figure 14.13. Upper and lower bounds versus iteration number are shown in figure 14.14. The worst case relative error is shown in figure 14.15. The ellipsoid E_k is shown at various iterations in figure 14.16.

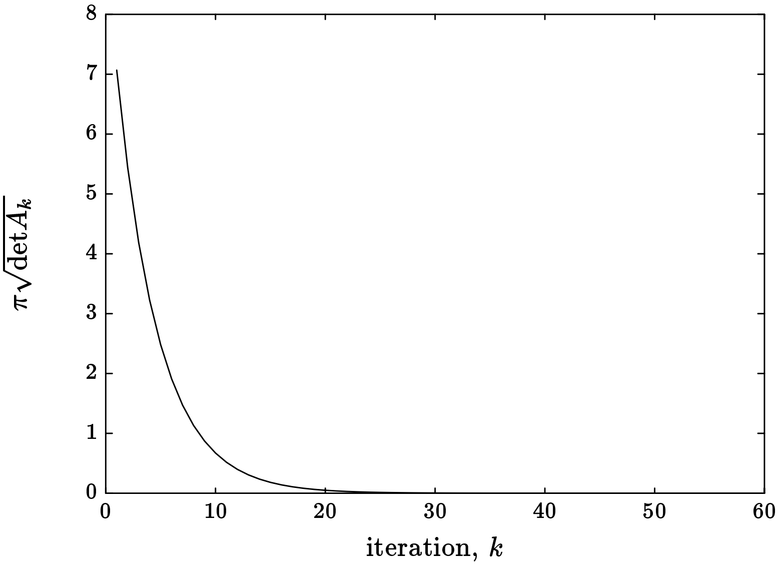


Figure 14.13 The volume of the ellipsoid E_k decreases exponentially with the iteration number k.

For the deep-cut algorithm we consider the inequality specifications

$$\varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta) \le 0.75 \qquad \varphi_{\mathrm{max\_sens}}(\alpha,\ \beta) \le 1.5 \qquad \varphi_{\mathrm{rms\_yp}}(\alpha,\ \beta) \le 0.05.$$

using the constraint function

$$\psi\left(\left[\begin{array}{c}\alpha\\ \beta\end{array}\right]\right) \stackrel{\Delta}{=} \max\left\{\frac{\varphi_{\mathrm{pk\_trk}}(\alpha,\ \beta)-0.75}{0.75},\ \frac{\varphi_{\mathrm{max\_sens}}(\alpha,\ \beta)-1.5}{1.5},\right.$$
$$\left.\frac{\varphi_{\mathrm{rms\_yp}}(\alpha,\ \beta)-0.05}{0.05}\right\}. \qquad (14.22)$$

The deep-cut ellipsoid algorithm finds a feasible point in 8 iterations. The execution of the algorithm is traced in table 14.1 and figure 14.17. Figure 14.17 also shows the set of feasible points, which coincides with the set where $\varphi_{\mathrm{wt\_max}}(\alpha,\beta) \le 0.75$.

14.5 Example: LQG Weight Selection via Duality

In this section we demonstrate some of the methods described in this chapter on the problem of weight selection for an LQG controller design. We consider the multicriterion LQG problem formulated in section 12.2.1, and will use the notation

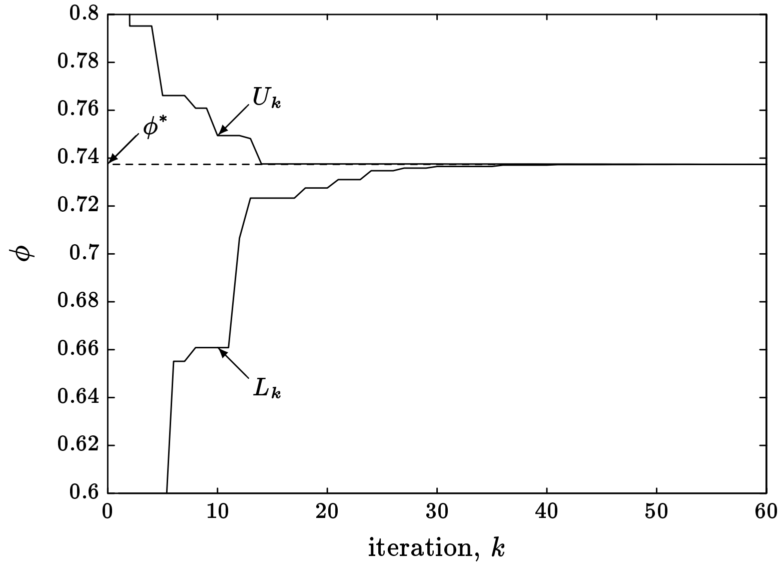


Figure 14.14 Upper and lower bounds on the solution ϕ^*, as a function of the iteration number k, for the ellipsoid algorithm.

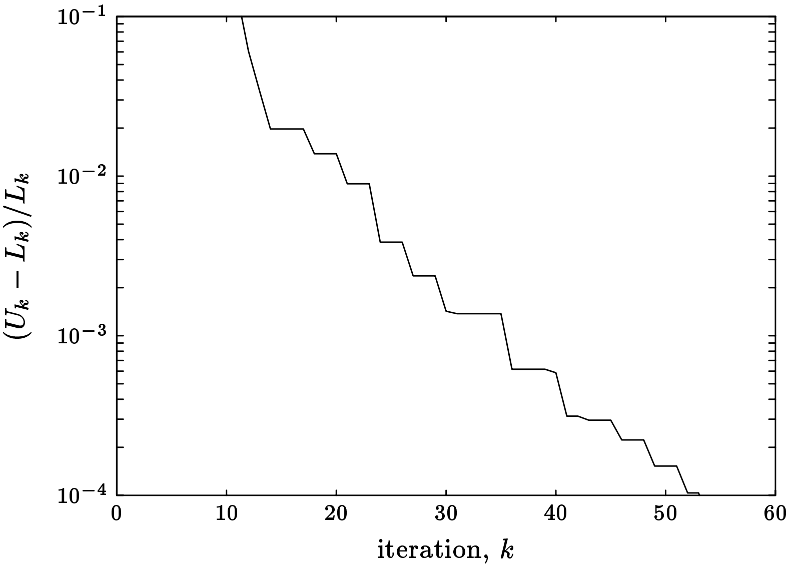


Figure 14.15 For the ellipsoid algorithm the maximum relative error, defined as $(U_k - L_k)/L_k$, falls below 0.01% by iteration 54.

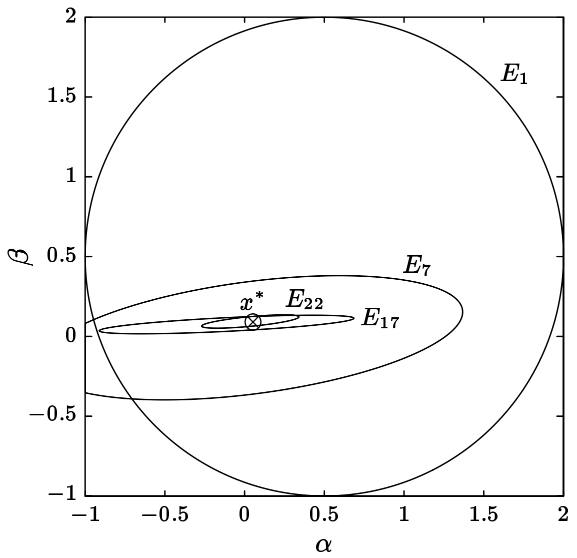


Figure 14.16 The initial ellipsoid, E_1, is a circle of radius 1.5 centered at $[0.5 \;\; 0.5]^T$. The ellipsoids at the 7th, 17th, and 22nd iterations are shown, together with the optimum x^*.

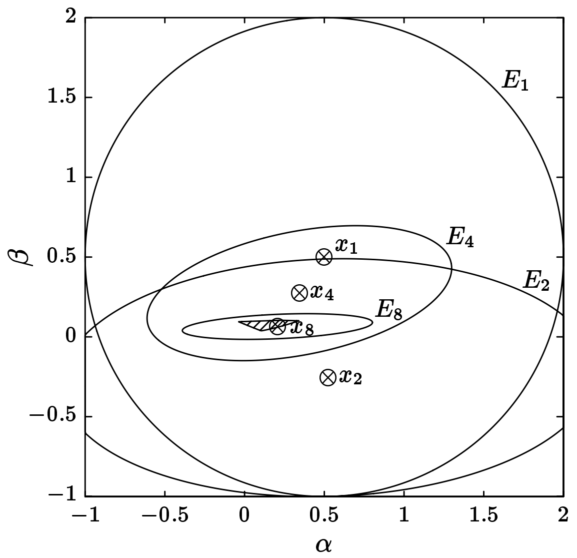


Figure 14.17 The initial ellipsoid, E_1 is a circle of radius 1.5 centered at $x_1 = [0.5 \;\; 0.5]^T$. After eight iterations of the deep-cut ellipsoid algorithm a feasible point x_8 is found for the constraint function (14.22). The set of feasible points for these specifications is shaded. Note that each ellipsoid contains the entire feasible set. See also table 14.1.

k	x_k	$\psi(x_k)$	$\phi_{\mathrm{pk\_trk}}$ ≤ 0.75?	$\phi_{\mathrm{max\_sens}}$ ≤ 1.5?	$\phi_{\mathrm{rms\_yp}}$ ≤ 0.05?	action
1	0.500 0.500	0.902	0.643 yes	2.853 no	0.0372 yes	cut $\phi_{\mathrm{max\_sens}}$
2	0.524 -0.256	0.160	0.858 no	1.739 no	0.0543 no	cut $\phi_{\mathrm{max\_sens}}$
3	-0.160 0.050	0.051	0.692 yes	1.489 yes	0.0526 no	cut $\phi_{\mathrm{rms\_yp}}$
4	0.344 0.274	0.384	0.661 yes	2.075 no	0.0422 yes	cut $\phi_{\mathrm{max\_sens}}$
5	0.169 0.021	0.009	0.719 yes	1.514 no	0.0498 yes	cut $\phi_{\mathrm{max\_sens}}$
6	-0.129 0.051	0.043	0.692 yes	1.488 yes	0.0522 no	cut $\phi_{\mathrm{rms\_yp}}$
7	0.236 0.122	0.041	0.693 yes	1.561 no	0.0466 yes	cut $\phi_{\mathrm{max\_sens}}$
8	0.206 0.065	-0.001	0.708 yes	1.498 yes	0.0483 yes	done

Table 14.1 The steps performed by the deep-cut ellipsoid algorithm for the constraint function (14.22). At each iteration a deep-cut was done using the function that had the largest normalized constraint violation. See also figure 14.17.

defined there. The specifications are realizability and limits on the RMS values of the components of z, *i.e.*,

$$\mathrm{RMS}(z_1) \leq \sqrt{a_1}, \ldots, \mathrm{RMS}(z_L) \leq \sqrt{a_L}, \tag{14.23}$$

(with the particular power spectral density matrix for w described in section 12.2.1). Each z_i is either an actuator signal or some linear combination of the system state. $\mathcal{A}$ will denote the set of $a \in \mathbf{R}_+^L$ that corresponds to achievable specifications of the form (14.23).

We will describe an algorithm that solves the feasibility problem for this family of specifications, *i.e.*, determines whether or not $a \in \mathcal{A}$, and if so, finds a controller that achieves the given specifications. This problem can usually be solved by a skilled designer using ad hoc weight adjustment and LQG design (see section 3.7.2), but we will describe an organized algorithm that cannot fail.

By the convex duality principle (see section 6.6; the technical condition holds in this case), we know that

$$a \in \mathcal{A} \iff \text{there is no } \lambda \geq 0 \text{ with } \psi(\lambda) > a^T\lambda.$$

Since this condition on λ is not affected by positive scaling, we may assume that the sum of the weights is one. We can thus pose our problem in terms of the following *convex* optimization problem:

$$\alpha = \min_{\substack{\lambda \geq 0 \\ \lambda_1 + \cdots + \lambda_L = 1}} a^T\lambda - \psi(\lambda). \tag{14.24}$$

Let λ^* denote a minimizer (which in fact is unique) for the problem (14.24). Then we have:

$$a \in \mathcal{A} \Longleftrightarrow \alpha \geq 0, \tag{14.25}$$

and moreover if $\alpha \geq 0$, then the LQG design that corresponds to weights λ^* achieves the specification (see section 12.2.1). So finding α and λ^* will completely solve our problem.

The equivalence (14.25) can be given a simple interpretation. $a^T\lambda$ is the LQG cost, using the weights λ, that a design corresponding to a would achieve, if it were feasible. $\psi(\lambda)$ is simply the smallest achievable LQG cost, using weights λ. It follows that if $a^T\lambda < \psi(\lambda)$, then the specification corresponding to a is not achievable, since it would beat the optimal LQG design. Thus, we can interpret (14.24) as the problem of finding the LQG weights such that the LQG cost of a design that just meets the specification a most favorably compares to the LQG-optimal cost.

We can evaluate a subgradient of $a^T\lambda - \psi(\lambda)$ (in fact, it is differentiable) using the formula given in section 13.4.8:

$$a - \begin{bmatrix} \phi_1(H_{\mathrm{lqg},\lambda}) \\ \vdots \\ \phi_L(H_{\mathrm{lqg},\lambda}) \end{bmatrix} \in \partial\left(a^T\lambda - \psi(\lambda)\right), \tag{14.26}$$

where $H_{\mathrm{lqg},\lambda}$ is the LQG-optimal design for the weights λ. We can therefore solve (14.24) using any of the algorithms described in this chapter. We will give some of the details for the ellipsoid algorithm.

We can handle the equality constraint $\lambda_1 + \cdots + \lambda_L = 1$ by letting

$$x = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{L-1} \end{bmatrix}$$

be our optimization variables, and setting

$$\lambda_L = 1 - \lambda_1 - \cdots - \lambda_{L-1}.$$

The inequality constraint on x is then

$$\max\{-x_1, \ldots, -x_{L-1}, x_1 + \cdots + x_{L-1} - 1\} \leq 0. \tag{14.27}$$

A subgradient $g \in \mathbf{R}^{L-1}$ for the objective is readily derived from (14.26):

$$g = \begin{bmatrix} a_1 - \phi_1(H_{\mathrm{lqg},\lambda}) - a_L + \phi_L(H_{\mathrm{lqg},\lambda}) \\ \vdots \\ a_{L-1} - \phi_{L-1}(H_{\mathrm{lqg},\lambda}) - a_L + \phi_L(H_{\mathrm{lqg},\lambda}) \end{bmatrix},$$

where $H_{\mathrm{lqg},\lambda}$ is optimal for the current weights.

As initial ellipsoid we take the ball of radius one centered at the weight vector

$$x^{(1)} = \begin{bmatrix} 1/L \\ \vdots \\ 1/L \end{bmatrix}.$$

Since this ball contains the feasible set given by (14.27), we are guaranteed that every feasible minimizer of (14.24) is inside our initial ellipsoid.

We can now directly apply the algorithm of section 14.4.3. We note a few simplifications to the stopping criterion. First, the feasible set is clearly not empty, so the algorithm will not terminate during a feasibility cut. Second, the stopping criterion can be:

$$\textit{until}\ \left(\ a^T\lambda < \psi(\lambda)\ \textit{or}\ \phi_i(H_{\mathrm{lqg},\lambda}) \le a_i\ \textit{for}\ i = 1, \ldots, L\ \right);$$

The algorithm will terminate either when a is known to be infeasible $(a^T\lambda < \psi(\lambda))$, or when a feasible design has been found.

This ellipsoid algorithm is guaranteed to terminate in a solution to our problem, except for the case when the specification a is Pareto optimal, *i.e.*, is itself an LQG-optimal specification for some weight vector. In this exceptional case, $H_{\mathrm{lqg},\lambda}$ and λ will converge to the unique design and associated weights that meet the specification a. This exceptional case can be ruled out by adding a small tolerance to either of the inequalities in the stopping criterion above.

14.5.1 Some Numerical Examples

We now demonstrate this algorithm on the standard example plant from section 2.4, with the process and sensor noise power spectral densities described in section 11.2.3 and the objectives

$$\phi_1(H) = \mathbf{E}\, u^2, \qquad \phi_2(H) = \mathbf{E}\, y_{\mathrm{p}}^2, \qquad \phi_3(H) = \mathbf{E}\, \dot{y}_{\mathrm{p}}^2,$$

which are the variance of the actuator signal, the system output signal y_{p}, and its derivative. The optimization variables are the weights λ_1 and λ_2 associated with the actuator and system output variance, respectively. The weight for the system output rate is given by $1 - \lambda_1 - \lambda_2$. The initial ellipsoid E_1 is a circle centered at $[1/3\ \ 1/3]^T$ of radius $\sqrt{5}/3$; it includes the triangle of feasible weights λ (see figure 14.19).

specification	RMS(u)	RMS($y_{\rm p}$)	RMS($\dot{y}_{\rm p}$)	achievable?
1	≤ 0.1	≤ 0.13	≤ 0.04	yes
2	≤ 0.1	≤ 0.07	≤ 0.04	no
3	≤ 0.15	≤ 0.07	≤ 0.04	yes
4	≤ 0.149	≤ 0.0698	≤ 0.04	no

Table 14.2 Four different specifications on the RMS values of u, $y_{\rm p}$, and $\dot{y}_{\rm p}$.

For specifications that are either deep inside or far outside the feasible set $\mathcal{A}$, the ellipsoid algorithm rapidly finds a set of weights for an LQG design that either achieves the specification or proves that the specification is infeasible. To get more than a few iterations, specifications close to Pareto optimal must be chosen. We will consider the four sets of specifications shown in table 14.2. Specifications 1 and 3 are achievable (the latter just barely), while 2 and 4 are unachievable (the latter only barely). The results of running the ellipsoid algorithm on each of these specifications is shown in table 14.3.

These results can be verified by checking whether each specification is above or below the tradeoff surface. Since each specification has the same limit on RMS($\dot{y}_{\rm p}$), we can plot a tradeoff curve between RMS($y_{\rm p}$) and RMS(u) when RMS($\dot{y}_{\rm p}$) is required to be below 0.04. Such a tradeoff curve is shown in figure 14.18. As expected, specification 1 is achievable and specification 2 is not. Specifications 3 and 4 are very close to the tradeoff boundary, but on opposite sides.

The execution of the ellipsoid algorithm is shown in

- figure 14.19 for specification 1,
- table 14.4 for specification 2,
- figure 14.20 for specification 3,
- figure 14.21 for specification 4.

quantity	spec. 1	spec. 2	spec. 3	spec. 4
$\sqrt{a}$	0.1 0.13 0.04	0.1 0.07 0.04	0.15 0.07 0.04	0.149 0.0698 0.04
iterations	23	9	59	34
ψ evaluations	12	6	47	22
λ	0.00955 0.00799 0.98246	0.01951 0.09072 0.88977	0.00226 0.02745 0.97029	0.00232 0.02952 0.96816
$\sqrt{J}$	0.09817 0.09795 0.03986	0.09726 0.05888 0.04202	0.14922 0.06997 0.03999	0.14896 0.06887 0.04005
$\psi(\lambda)$	0.001729	0.002070	0.001737	0.001745
$a^T\lambda$	0.001802	0.002063	0.001738	0.001744
exit condition	$J < a$	$\psi(\lambda) > a^T\lambda$	$J < a$	$\psi(\lambda) > a^T\lambda$
achievable?	yes	no	yes	no

Table 14.3 Result of running the ellipsoid algorithm on each of the four specifications from table 14.2.

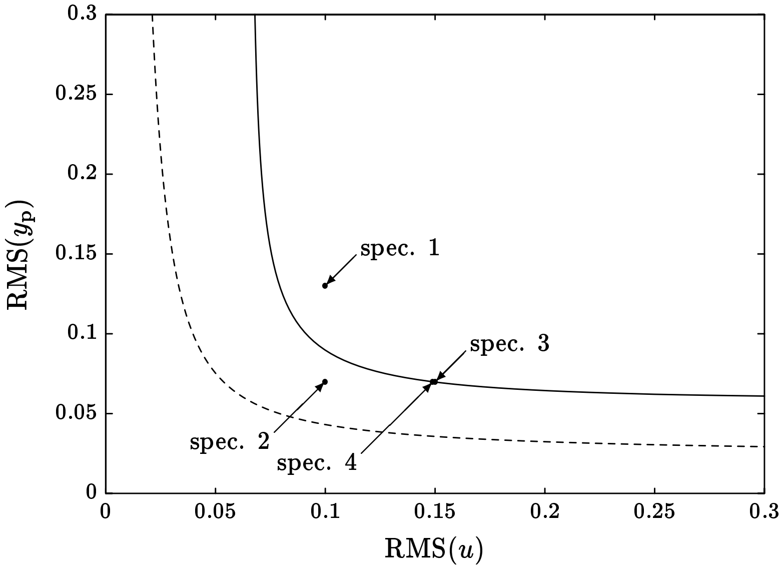


Figure 14.18 The tradeoff curve between $\mathrm{RMS}(y_\mathrm{p})$ and $\mathrm{RMS}(u)$, with the specification $\mathrm{RMS}(\dot{y}_\mathrm{p}) \leq 0.04$, is shown, together with the four specifications from table 14.2. Since these four specifications all require that $\mathrm{RMS}(\dot{y}_\mathrm{p}) \leq 0.04$, each specification will be achievable if it lies on or above the tradeoff curve. For comparison, the tradeoff curve between $\mathrm{RMS}(y_\mathrm{p})$ and $\mathrm{RMS}(u)$ with no specification on $\mathrm{RMS}(\dot{y}_\mathrm{p})$ is shown with a dashed line.

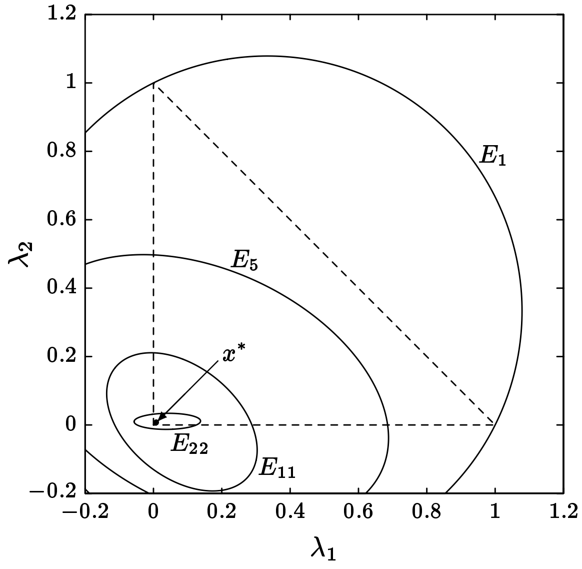


Figure 14.19 The progression of the ellipsoid algorithm is shown for specification 1 in table 14.2. The initial ellipsoid, E_1, is a circle of radius $\sqrt{5}/3$ centered at $x^{(1)} = [1/3 \ \ 1/3]^T$ that includes the triangular set of feasible weights. The ellipsoid algorithm terminates at the point x^*, which corresponds to weights for which the LQG-optimal controller satisfies specification 1.

iter.	λ	$\sqrt{J}$	$\psi(\lambda)$	$a^T\lambda$	action
1	0.33333 0.33333 0.33333	0.06065 0.06265 0.04789	0.00330	0.00550	cut ψ
2	0.09163 0.27584 0.63254	0.07837 0.05383 0.04533	0.00266	0.00328	cut ψ
3	−0.02217 0.11561 0.90656	—	—	—	cut λ_1
4	0.14809 −0.07633 0.92824	—	—	—	cut λ_2
5	0.09606 0.20281 0.70113	0.07392 0.05758 0.04474	0.00260	0.00308	cut ψ
6	−0.00277 0.14218 0.86059	—	—	—	cut λ_1
7	0.10218 0.00942 0.88840	0.05459 0.11386 0.04218	0.00201	0.00249	cut ψ
8	0.05750 0.20722 0.73528	0.08348 0.05416 0.04421	0.00245	0.00277	cut ψ
9	0.01951 0.09072 0.88977	0.09726 0.05888 0.04202	0.00207	0.00206	done

Table 14.4 Tracing the execution of the ellipsoid algorithm with specification 2 from table 14.2. After 9 iterations $\psi(\lambda) > a^T\lambda$, so specification 2 has been proven to be unachievable.

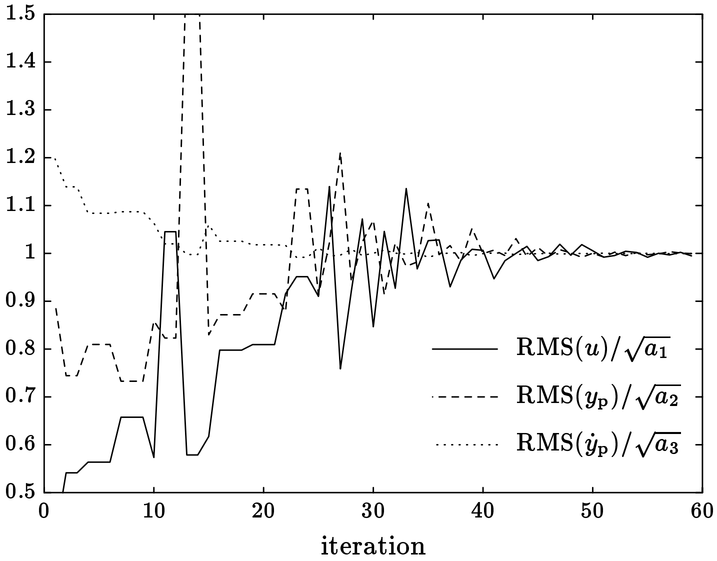


Figure 14.20 The normalized values of $\mathrm{RMS}(u)$, $\mathrm{RMS}(y_\mathrm{p})$, and $\mathrm{RMS}(\dot{y}_\mathrm{p})$ are plotted versus iteration number for specification 3 in table 14.2. These values are those achieved by the LQG-optimal regulator with the current weights λ. At iteration 59 all three curves are simultaneously at or below 1.0, so the algorithm terminates; it has found weights for which the LQG-optimal regulator satisfies the specifications.

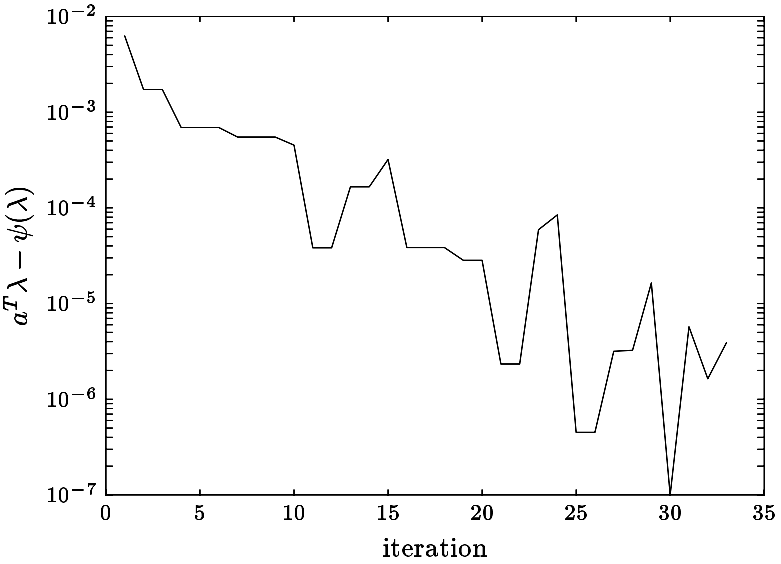


Figure 14.21 The difference between $a^T\lambda$ and $\psi(\lambda)$ is plotted for each iteration of the ellipsoid algorithm for specification 4 in table 14.2. At iteration 34, $\psi(\lambda) > a^T\lambda$ (this point is not plotted), and the algorithm terminates; it has proven that the specifications are unachievable. Note that at iteration 30, the weights almost proved that the specification is unachievable.

14.6 Complexity of Convex Optimization

We have seen simple algorithms that can be used to compute the global minimum of a convex or quasiconvex function. In fact, these optimization problems are not only solvable, they are intrinsically *tractable*: roughly speaking, they can be solved with a "reasonable" amount of computation. For more general optimization problems, while we can often compute a *local* minimum with reasonable computation, the computation of the *global* minimum usually requires vastly more computation, and so is tractable only for small problems.

In this section we informally discuss the *complexity* of different types of (global) optimization problems, which involves the following ideas:

- a class of problems,
- a notion of the size of a problem and required accuracy of solution,
- a measure of computation effort.

The complexity describes how the minimum computation effort required depends on the problem size and required accuracy.

The four classes of optimization problems that we will consider are:

- *Convex quadratic program* (QP): compute

  $$\min\left\{x^TAx \;\middle|\; Cx \le B\right\},$$

  where $x \in \mathbf{R}^n$, $B \in \mathbf{R}^k$, and $A \ge 0$.

- *General QP*: compute

  $$\min\left\{x^TAx \;\middle|\; Cx \le B\right\},$$

  where $x \in \mathbf{R}^n$ and $B \in \mathbf{R}^k$.

- *Convex program*: compute

  $$\min_{x\in\mathcal{K}} f(x)$$

  where $f : \mathbf{R}^n \to \mathbf{R}$ is convex, and $\mathcal{K} \subseteq \mathbf{R}^n$ is a convex set.

- *General program*: compute

  $$\min_{x\in\mathcal{K}} f(x)$$

  where $f : \mathbf{R}^n \to \mathbf{R}$ and $\mathcal{K} \subseteq \mathbf{R}^n$.

In the following sections we describe some measures of the relative complexities of these problems.

14.6.1 Bit Complexity of Quadratic Programs

The problem size is measured by the number of bits required to describe the matrices A and C, and the vector B. The computational effort is measured by the number of bit operations required to find the (exact) solution.

Convex QP's have polynomial bit complexity: they can be solved in a number of bit operations that grows no faster than a polynomial function of the problem size. This celebrated result was obtained by the Soviet mathematician Khachiyan in 1979 using a variation of the ellipsoid algorithm.

In contrast, it is known that general QP's have a bit complexity that is the same as many other "hard" problems. The only known algorithms that solve these "hard" problems require a number of bit operations that grows exponentially with the problem size. Moreover, it is generally believed that there are no algorithms that solve these problems using a number of bit operations that grows only as fast a polynomial in the problem size. See the Notes and References.

In conclusion, in the sense of bit complexity, convex QP's are "tractable", whereas no non-combinatorial algorithms for solving general QP's are known, and it is widely believed that none exist.

14.6.2 Information Based Complexity

Information based complexity is measured by the number of function and gradient (or subgradient) evaluations of f that are needed to compute the minimum to a guaranteed accuracy; information about f is only known through these function and gradient evaluations. The problem size is measured by the dimension n and required accuracy ϵ.

Roughly speaking, bit complexity counts all operations, and assumes that the function to be minimized is completely specified, whereas information based complexity counts only the number of calls to a subroutine that evaluates the function and a gradient.

Many sharp bounds are known for the information based complexity of convex and general programs. For example, convex programs can be solved to an accuracy of ϵ with no more than $p(n)\log(1/\epsilon)$ function and subgradient evaluations, where p is a specific polynomial. It is also known that the minimum number of function and gradient evaluations required to solve a general program grows like $(1/\epsilon)^n$, *i.e.*, exponentially.

Therefore, the information based complexity of general programs is enormously larger than convex programs.

14.6.3 Some Caveats

We warn the reader that the results described above treat idealized versions of complexity, which may or may not describe the practical tractability.

For example, the simplex method for solving convex QP's works very well in practice, even though it is not a polynomial algorithm. The ellipsoid algorithm for solving convex QP's, on the other hand, is polynomial, but so far has not performed better in practice than the simplex method.

As another example, we note that the information based complexity of general QP's and convex QP's is the same, and no bigger than a polynomial in $n + k$ (because once we know what QP we must solve, no more subroutine calls to the function evaluator are required). In contrast, the bit complexity of convex and general QP's is generally considered to be enormously different, which is consistent with practical experience. The low information based complexity of general QP's is a result, roughly speaking, of local computations being "free", with "charges" incurred only for function evaluations.

14.6.4 Local Versus Global Optimization

General programs are often "solved" in practice using local, often descent, methods. These algorithms often converge rapidly to a *local* minimum of the function f. In many cases, this local minimum is the global minimum. But *verifying* that a given local minimum is actually global has a high computational cost.

Local optimization methods often work well for design tasks: they rapidly find local minima, which in many cases are actually global, but in return, they give up the certainty of determining the global minimum. Often, an acceptable design can be found using a local optimization method.

However, local methods cannot determine a limit of performance: global optimization is needed to confidently know that a design *cannot* be achieved, *i.e.*, lies in the unshaded region described in section 1.4.1 of chapter 1. So the value of convexity is that not only can we find designs that we know are "good" (*i.e.*, nearly Pareto optimal), we can also, with reasonable computation, determine performance limits.

Notes and References

Descent Methods

General descent methods are described in, *e.g.*, Luenberger [LUE84]. General methods for nondifferentiable optimization are surveyed in Polak [POL87]. Kiwiel's monograph [KIW85] gives a detailed description of many descent algorithms for nondifferentiable optimization; see also Fukushima [FUK84] and the references cited there. In the Soviet literature, descent methods are sometimes called *relaxation processes*; see for example the survey by Lyubich and Maistrovskii [LM70].

Specializations of general-purpose algorithms to nondifferentiable convex optimization include Wolfe's conjugate subgradients algorithm [WOL75] and Lemarechal's Davidon method for nondifferentiable objectives [LEM75].

Nondifferentiable Convex Optimization

A good general reference on convex optimization is Levitin and Polyak [LP66], which treats the infinite-dimensional case, but considers mostly smooth problems. Good general references on nondifferentiable convex optimization, which describe some methods presented in this chapter are Akgül [AKG84], Evtushenko [EVT85], and Demyanov and Vasilev [DV85]. In [SHO85], Shor describes the *subgradient algorithm*, a precursor of the ellipsoid method, and a variable-metric subgradient algorithm, which is the ellipsoid method as he originally developed it.

Rockafellar's books [ROC81, ROC82] describe subgradients and convex optimization, but not the algorithms we have presented in this chapter.

Cutting-Plane Methods

The cutting-plane algorithm described in section 14.3.2 is generally attributed to Kelley [KEL60], although it is related to earlier algorithms, *e.g.*, Cheney and Goldstein's method [CG59], which they call Newton's method for convex programming. In fact, Kelley's description of the algorithm is for the constrained case: he shows how to convert the unconstrained problem into one with constraints and linear objective.

Convergence proofs are given in, *e.g.*, Kelley [KEL60], Levitin and Polyak [LP66, §10], Demyanov and Vasilev [DV85, §3.8], and Luenberger [LUE84, P419–420]. The cutting-plane algorithm for constrained optimization that we presented in section 14.3.4, and a proof of its convergence, can be found in Demyanov and Vasilev [DV85, P420–423].

Elzinga and Moore [EM75] give a cutting-plane algorithm for the constrained problem that has two possible advantages over the one described in section 14.3.4. Their algorithm generates feasible iterates, so we can use the simple stopping criteria used in the algorithm for the unconstrained problem, and we do not have to accept "slightly violated constraints". Their algorithm also drops old constraints, so that the linear programs solved at each iteration do not grow in size as the algorithm proceeds. Demyanov and Vasilev [DV85, §3.9–10] describe the *extremum basis method*, a cutting-plane algorithm that maintains a fixed number of constraints in the linear program. See also Gonzaga and Polak [GP79].

Ellipsoid Methods

Two early articles that give algorithms for minimizing a quasiconvex function using the idea that a quasigradient evaluated at a point rules out a half-space containing the point are Newman [NEW65] and Levin [LEV65]. In these precursors of the ellipsoid algorithm, the complexity of the set which is known to contain a minimizer increases as the algorithm proceeds, so that the computation per iteration grows, as in cutting-plane methods.

The ellipsoid algorithm was developed in the 1970's in the Soviet Union by Shor, Yudin, and Nemirovsky. A detailed history of its development, including English and Russian references, appears in chapter 3 of Akgül [AKG84]. It was used in 1979 by Khachiyan in his famous proof that linear programs can be solved in polynomial time; an English translation appears in [KHA79] (see also Gács and Lovázs [GL81]).

The 1981 survey by Bland, Goldfarb, and Todd [BGT81] contains a very clear description of the method and has extensive references on its development and early history, concentrating on its application to linear programming. Our exposition follows Chapter 3 of the book by Grötschel, Lovász, and Schrijver [GLS88]. In [GOF83, GOF84], Goffin gives an interpretation of the ellipsoid algorithm as a variable-metric descent algorithm.

Ecker and Kupferschmid [EK83, EK85] describe the results of extensive numerical tests on a large number of benchmark optimization problems, comparing an ellipsoid method to other optimization algorithms. Some of these problems are not convex, but the ellipsoid method seems to have done very well (*i.e.*, found feasible points with low function values) even though it was not designed for nonconvex optimization. In the paper [KME85], Kupferschmid et al. describe an application of an ellipsoid algorithm to the nonconvex, but important, problem of feedback gain optimization.

Initializing the Cutting-Plane and Ellipsoid Algorithms

In many cases we can determine, a priori, a box or ellipsoid that is guaranteed to contain a minimizer; we saw an example in the LQG weight selection problem.

For the cutting-plane algorithm, if we do not know an initial box that is guaranteed to contain a minimizer, we can *guess* an initial box; if x_k stays on the box boundary for too many iterations, then we increase the box size and continue. (By continue, we mean that all of the information gathered in previous function and subgradient evaluations can be kept.) Roughly speaking, once x_k lands inside the box, we can be certain that our initial box was large enough (provided the Lagrange multipliers in the linear program are positive at the iterate inside the box).

In many cases the ellipsoid algorithm converges to a minimizer even if no minimizers were inside the initial ellipsoid, although of course there is no guarantee that this will happen. Of course, this can only occur because at each iteration we include in our new ellipsoid some points that were not in the previous ellipsoid. These new points normally represent "wasted" ellipsoid volume, since we are including points that are known not to include a minimizer. But in the case where the minimizer lies outside the initial ellipsoid, these new points allow the ellipsoid to twist around so as to include a minimizer.

If an iterate x_k falls outside the initial ellipsoid, it is good practice to restart the algorithm with a larger ellipsoid. We have found a very slow rise in the total number of ellipsoid algorithm iterations required as the size of the initial ellipsoid is increased, despite the fact that the volume of the initial ellipsoid increases very rapidly with its semi-axes.

LQG Weight Selection

The informal method of adjusting weights is used extensively in LQG design. In the LQG problem, the maximum value rule-of-thumb for initial weights is often referred to as *Bryson's rule* [BH75].

In [HS78], Harvey and Stein state,

> To use LQG methods, the designer must reduce all of his varied performance requirements to a single criterion which is constrained to be quadratic in state and controls. So little is known about the relationship between such specific criteria and more general control design specifications that the designer must invariably resort to trial and error iterations.

In Stein [STE79], we find

> Probably the most important area about which better understanding [of LQG weight selection] is needed is the relationship between the weighting parameters selected for the basic scalar performance index and the resulting regulator properties. Practitioners have wrestled with this relationship for nearly two decades, trying various intuitive ways to select a "good set of weights" to satisfy various design specifications.

We have observed that the mapping from LQG weights into the resulting optimal LQG cost, *i.e.*, the dual function ψ, is concave. We exploited this property to design an algorithm that solves the LQG multicriterion feasibility problem.

We were originally unaware of any published algorithm that solves this feasibility problem, but recently discovered an article by Toivonen and Mäkilä [TM89], which uses a quadratically convergent method to minimize $a^T\lambda - \psi(\lambda)$ over $\lambda \geq 0$.

The deep-cut algorithm of section 14.4.4 can be used with the inequality specification $a^T\lambda - \psi(\lambda) \geq 0$. Such an algorithm will determine whether the RMS specifications are achievable or not, but, unlike the algorithm we described, might not find a design meeting the specification when the specifications are achievable.

Complexity of Convex Optimization

Bit complexity is described in, for example, the book by Garey and Johnson [GJ79]. The "hard" problems we referred to in section 14.6.1 are called NP-complete. The results on bit complexity of quadratic programs are described in Pardalos and Rosen [PR87]. Pardalos and Rosen also describe some methods for solving general QP's; these methods require large computation times on supercomputers, whereas very large convex QP's are readily solved on much smaller computers.

The results on the information based complexity of optimization problems are due to Nemirovsky and Yudin, and described in the clear and well written book [NY83] and article [YN77]. Our description of the problem is not complete: the problems considered must also have some known bound on the size of a subgradient over $\mathcal{K}$ (which we used in our proofs of convergence).

Chapter 15

Solving the Controller Design Problem

In this chapter we describe methods for forming and solving finite-dimensional approximations to the controller design problem. A method based on the parametrization described in chapter 7 yields an *inner* approximation of the region of achievable specifications in performance space. For some problems, an *outer* approximation of this region can be found by considering a dual problem. By forming both approximations, the controller design problem can be solved to an arbitrary, and guaranteed, accuracy.

In chapter 3 we argued that many approaches to controller design could be described in terms of a family of design specifications that is parametrized by a *performance vector* $a \in \mathbf{R}^L$,

$$H \text{ satisfies } \mathcal{D}_{\mathrm{hard}}, \quad \phi_1(H) \le a_1, \ldots, \phi_L(H) \le a_L. \tag{15.1}$$

Some of these specifications are unachievable; the designer must choose among the specifications that are achievable. In terms of the performance vectors, the designer must choose an $a \in \mathcal{A}$, where $\mathcal{A}$ denotes the set of performance vectors that correspond to achievable specifications of the form (15.1). We noted in chapter 3 that the actual controller design problem can take several specific forms, *e.g.*, a constrained optimization problem with weighted-sum or weighted-max objective, or a simple feasibility problem.

In chapters 7–11 we found that in many controller design problems, the hard constraint $\mathcal{D}_{\mathrm{hard}}$ is convex (or even affine) and the functionals $\phi_1, \ldots, \phi_L$ are convex; we refer to these as *convex controller design problems.* We refer to a controller design problem in which one or more of the functionals is quasiconvex but not convex as a *quasiconvex controller design problem.* These controller design problems can be considered convex (or quasiconvex) optimization problems over $\mathcal{H}$; since $\mathcal{H}$

has infinite dimension, the algorithms described in the previous chapter cannot be directly applied.

15.1 Ritz Approximations

The Ritz method for solving infinite-dimensional optimization problems consists of solving the problem over larger and larger finite-dimensional subsets. For the controller design problem, the Ritz approximation method is determined by a sequence of $n_z \times n_w$ transfer matrices

$$R_0,\ R_1,\ R_2, \ldots \in \mathcal{H}. \tag{15.2}$$

We let

$$\mathcal{H}_N \stackrel{\Delta}{=} \left\{ R_0 + \sum_{1 \le i \le N} x_i R_i \;\middle|\; x_i \in \mathbf{R},\ 1 \le i \le N \right\}$$

denote the finite-dimensional affine subset of $\mathcal{H}$ that is determined by R_0 and the next N transfer matrices in the sequence. The Nth Ritz approximation to the family of design specifications (15.1) is then

$$H \text{ satisfies } \mathcal{D}_{\text{hard}}, \quad \phi_1(H) \le a_1, \ldots, \phi_L(H) \le a_L, \quad H \in \mathcal{H}_N. \tag{15.3}$$

The Ritz approximation yields a convex (or quasiconvex) controller design problem, if the original controller problem is convex (or quasiconvex), since it is the original problem with the affine specification $H \in \mathcal{H}_N$ adjoined.

The Nth Ritz approximation to the controller design problem can be considered a finite-dimensional optimization problem, so the algorithms described in chapter 14 can be applied. With each $x \in \mathbf{R}^N$ we associate the transfer matrix

$$H_N(x) \stackrel{\Delta}{=} R_0 + \sum_{1 \le i \le N} x_i R_i, \tag{15.4}$$

with each functional ϕ_i we associate the function $\phi_i^{(N)} : \mathbf{R}^N \to \mathbf{R}$ given by

$$\phi_i^{(N)}(x) \stackrel{\Delta}{=} \phi_i(H_N(x)), \tag{15.5}$$

and we define

$$\mathcal{D}^{(N)} \stackrel{\Delta}{=} \{x \mid H_N(x) \text{ satisfies } \mathcal{D}_{\text{hard}}\}. \tag{15.6}$$

Since the mapping from $x \in \mathbf{R}^N$ into $\mathcal{H}$ given by (15.4) is affine, the functions $\phi_i^{(N)}$ given by (15.5) are convex (or quasiconvex) if the functionals ϕ_i are; similarly the subsets $\mathcal{D}^{(N)} \subseteq \mathbf{R}^N$ are convex (or affine) if the hard constraint $\mathcal{D}_{\text{hard}}$ is. In section 13.5 we showed how to compute subgradients of the functions $\phi_i^{(N)}$, given subgradients of the functionals ϕ_i.

Let $\mathcal{A}_N$ denote the set of performance vectors that correspond to achievable specifications for the Nth Ritz approximation (15.3). Then we have

$$\mathcal{A}_1 \subseteq \cdots \subseteq \mathcal{A}_N \subseteq \cdots \subseteq \mathcal{A},$$

i.e., the Ritz approximations yield *inner* or *conservative* approximations of the region of achievable specifications in performance space.

If the sequence (15.2) is chosen well, and the family of specifications (15.1) is well behaved, then the approximations $\mathcal{A}_N$ should in some sense converge to $\mathcal{A}$ as $N \to \infty$. There are many conditions known that guarantee this convergence; see the Notes and References at the end of this chapter.

We note that the specification $\mathcal{H}_{\text{slice}}$ of chapter 11 corresponds to the $N = 2$ Ritz approximation:

$$R_0 = H^{(\mathrm{c})}, \qquad R_1 = H^{(\mathrm{a})} - H^{(\mathrm{c})}, \qquad R_2 = H^{(\mathrm{b})} - H^{(\mathrm{c})}.$$

15.1.1 A Specific Ritz Approximation Method

A specific method for forming Ritz approximations is based on the parametrization of closed-loop transfer matrices achievable by stabilizing controllers (see section 7.2.6):

$$\mathcal{H}_{\text{stable}} = \{T_1 + T_2QT_3 \mid Q \text{ stable}\}. \tag{15.7}$$

We choose a sequence of stable $n_u \times n_y$ transfer matrices $Q_1, Q_2, \ldots$ and form

$$R_0 = T_1, \qquad R_k = T_2Q_kT_3, \quad k = 1, 2, \ldots \tag{15.8}$$

as our Ritz sequence. Then we have $\mathcal{H}_N \subseteq \mathcal{H}_{\text{stable}}$, *i.e.*, we have automatically taken care of the specification $\mathcal{H}_{\text{stable}}$.

To each $x \in \mathbf{R}^N$ there corresponds the controller $K_N(x)$ that achieves the closed-loop transfer matrix $H_N(x) \in \mathcal{H}_N$: in the Nth Ritz approximation, we search over a set of controllers that is parametrized by $x \in \mathbf{R}^N$, in the same way that the family of PID controllers is parametrized by the vector of gains, which is in $\mathbf{R}^3$. But the parametrization $K_N(x)$ has a very special property: *it preserves the geometry of the underlying controller design problem.* If a design specification or functional is closed-loop convex or affine, so is the resulting constraint on or function of $x \in \mathbf{R}^N$. This is not true of more general parametrizations of controllers, *e.g.*, the PID controllers. The controller architecture that corresponds to the parametrization $K_N(x)$ is shown in figure 15.1.

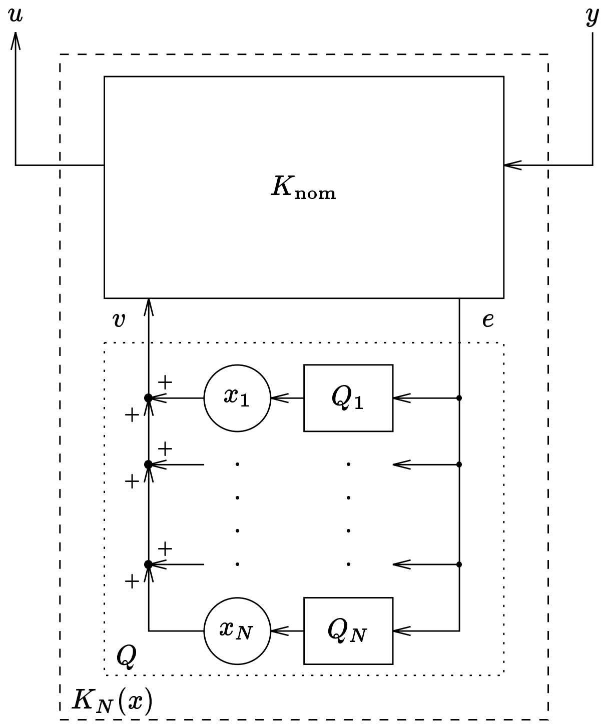


Figure 15.1 The Ritz approximation (15.7–15.8) corresponds to a parametrized controller $K_N(x)$ that consists of two parts: a nominal controller K_{nom}, and a stable transfer matrix Q that is a linear combination of the fixed transfer matrices $Q_1, \ldots, Q_N$. See also section 7.3 and figure 7.5.

15.2 An Example with an Analytic Solution

In this section we demonstrate the Ritz method on a problem that has an analytic solution. This allows us to see how closely the solutions of the approximations agree with the exact, known solution.

15.2.1 The Problem and Solution

The example we will study is the standard plant from section 2.4. We consider the RMS actuator effort and RMS regulation functionals described in sections 11.2.3 and 11.3.2:

$$\begin{aligned}\mathrm{RMS}(y_{\mathrm{p}}) &\overset{\Delta}{=} \phi_{\mathrm{rms\_yp}}(H) = \left(\|H_{12}W_{\mathrm{sensor}}\|_2^2 + \|H_{13}W_{\mathrm{proc}}\|_2^2\right)^{1/2} \\ \mathrm{RMS}(u) &\overset{\Delta}{=} \phi_{\mathrm{rms\_u}}(H) = \left(\|H_{22}W_{\mathrm{sensor}}\|_2^2 + \|H_{23}W_{\mathrm{proc}}\|_2^2\right)^{1/2}.\end{aligned}$$

We consider the specific problem:

$$\min_{\phi_{\mathrm{rms\_yp}}(H) \le 0.1} \quad \phi_{\mathrm{rms\_u}}(H). \tag{15.9}$$

The solution, $\phi^*_{\mathrm{rms\_u}} = 0.0397$, can be found by solving an LQG problem with weights determined by the algorithm given in section 14.5.

15.2.2 Four Ritz Approximations

We will demonstrate four different Ritz approximations, by considering two different parametrizations (*i.e.*, T_1, T_2, and T_3 in (15.7)) and two different sequences of stable Q's.

The parametrizations are given by the formulas in section 7.4 using the two estimated-state-feedback controllers $K^{(\mathrm{a})}$ and $K^{(\mathrm{d})}$ from section 2.4 (see the Notes and References for more details). The sequences of stable transfer matrices we consider are

$$Q_i = \left(\frac{1}{s+1}\right)^i, \qquad \tilde{Q}_i = \left(\frac{4}{s+4}\right)^i, \qquad i = 1, \ldots \tag{15.10}$$

We will denote the four resulting Ritz approximations as

$$(K^{(\mathrm{a})}, Q), \qquad (K^{(\mathrm{a})}, \tilde{Q}), \qquad (K^{(\mathrm{d})}, Q), \qquad (K^{(\mathrm{d})}, \tilde{Q}). \tag{15.11}$$

The resulting finite-dimensional Ritz approximations of the problem (15.9) turn out to have a simple form: both the objective and the constraint function are convex quadratic (with linear and constant terms) in x. These problems were solved exactly using a special algorithm for such problems; see the Notes and References at the end of this chapter. The performance of these four approximations is plotted in figure 15.2 along with a dotted line that shows the exact optimum, 0.0397. Figure 15.3 shows the same data on a more detailed scale.

15.3 An Example with no Analytic Solution

We now consider a simple modification to the problem (15.9) considered in the previous section: we add a constraint on the overshoot of the step response from the reference input r to the plant output y_{p}, *i.e.*,

$$\min_{\substack{\phi_{\mathrm{rms\_yp}}(H) \le 0.1 \\ \phi_{\mathrm{os}}(H_{13}) \le 0.1}} \quad \phi_{\mathrm{rms\_u}}(H). \tag{15.12}$$

Unlike (15.9), no analytic solution to (15.12) is known. For comparison, the optimal design for the problem (15.9) has a step response overshoot of 39.7%.

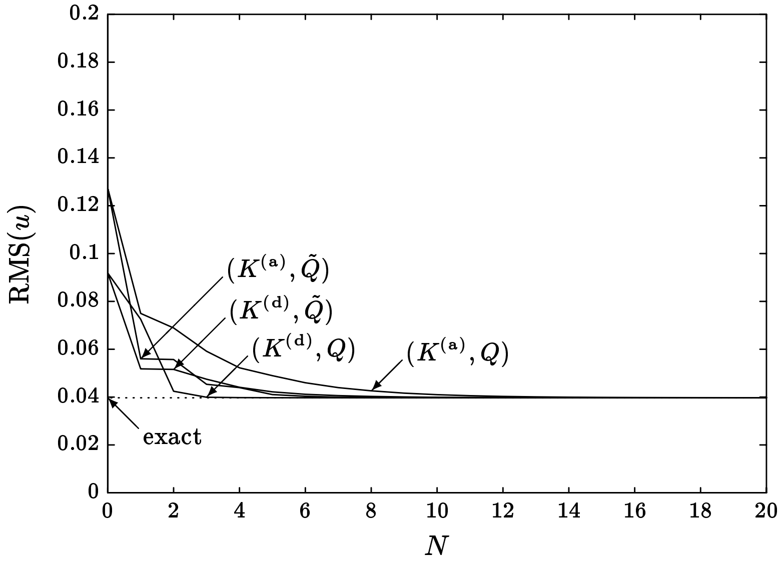


Figure 15.2 The optimum value of the finite-dimensional inner approximation of the optimization problem (15.9) versus the number of terms N for the four different Ritz approximations (15.11). The dotted line shows the exact solution, RMS(u) = 0.0397.

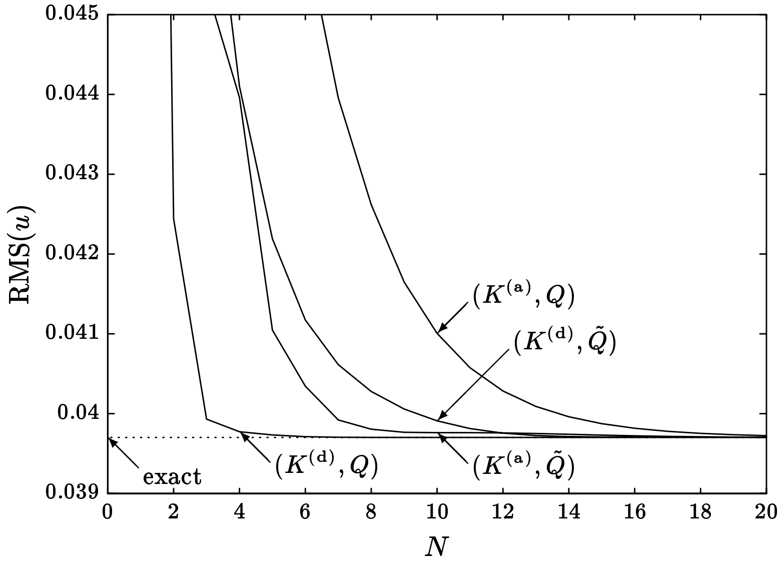


Figure 15.3 Figure 15.2 is re-plotted to show the convergence of the finite-dimensional inner approximations to the exact solution, RMS(u) = 0.0397.

The same four Ritz approximations (15.11) were formed for the problem (15.12), and the ellipsoid algorithm was used to solve them. The performance of the approximations is plotted in figure 15.4, which the reader should compare to figure 15.2. The minimum objective values for the Ritz approximations appear to be converging to 0.058, whereas without the step response overshoot specification, the minimum objective is 0.0397. We can interpret the difference between these two numbers as the cost of reducing the step response overshoot from 39.7% to 10%.

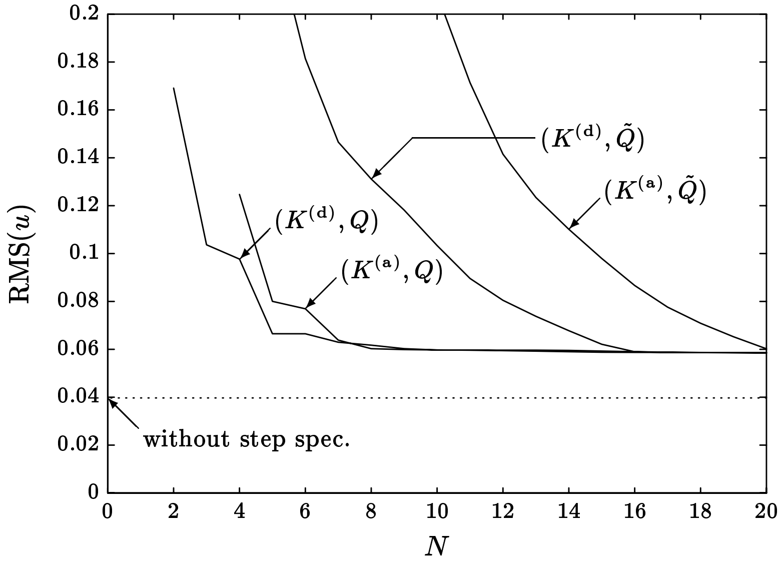


Figure 15.4 The optimum value of the finite-dimensional inner approximation of the optimization problem (15.12) versus the number of terms N for the four different Ritz approximations (15.11). No analytic solution to this problem is known. The dotted line shows the optimum value of $\mathrm{RMS}(u)$ without the step response overshoot specification.

15.3.1 Ellipsoid Algorithm Performance

The finite-dimensional optimization problems produced by the Ritz approximations of (15.12) are much more substantial than any of the numerical example problems we encountered in chapter 14, which were limited to two variables (so we could plot the progress of the algorithms). It is therefore worthwhile to briefly describe how the ellipsoid algorithms performed on the $N = 20$ $(K^{(\mathrm{a})}, Q)$ Ritz approximation, as an example of a demanding numerical optimization problem.

The basic ellipsoid algorithm was initialized with $A_1 = 5000I$, so the initial ellipsoid was a sphere with radius 70.7. All iterates were well inside this initial ellipsoid. Moreover, increasing the radius had no effect on the final solution (and, indeed, only a small effect on the total computation time).

The algorithm took 34 iterations to find a feasible point, and 4259 iterations for the maximum relative error to fall below 0.1%. The maximum and actual relative errors versus iteration number are shown in figure 15.5. The relative constraint violation and the normalized objective function value versus iteration number are shown in figure 15.6. From these figures it can be seen that the ellipsoid algorithm produces designs that are within a few percent of optimal within about 1500 iterations.

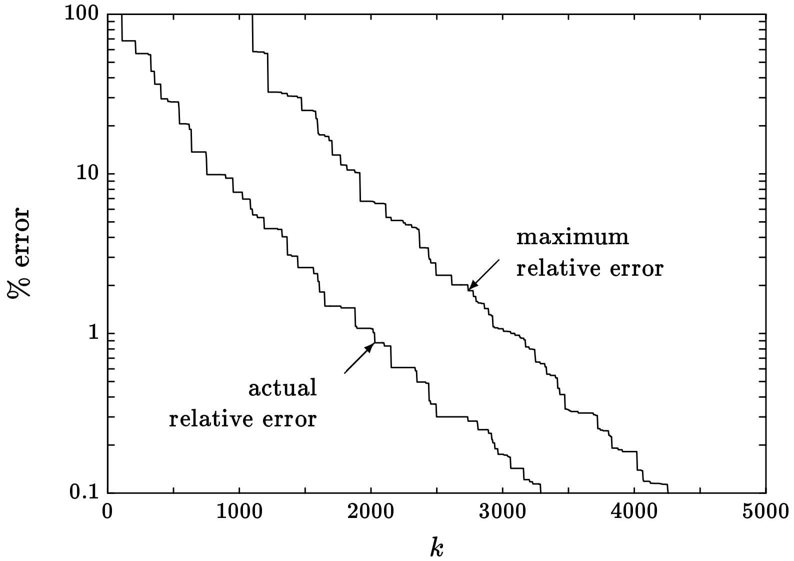


Figure 15.5 The ellipsoid algorithm maximum and actual relative errors versus iteration number, k, for the solution of the $N = 20$ $(K^{(\mathrm{a})}, Q)$ Ritz approximation of (15.12). After 34 iterations a feasible point has been found, and after 4259 iterations the objective has been computed to a maximum relative error of 0.1%. Note the similarity to figure 14.15, which shows a similar plot for a much simpler, two variable, problem.

For comparison, we solved the same problem using the ellipsoid algorithm with deep-cuts for both objective and constraint cuts, with the same initial ellipsoid. A few more iterations were required to find a feasible point (53), and somewhat fewer iterations were required to find the optimum to within 0.1% (2620). Its performance is shown in figure 15.7. The objective and constraint function values versus iteration number for the deep-cut ellipsoid algorithm were similar to the basic ellipsoid algorithm.

The number of iterations required to find the optimum to within a guaranteed maximum relative error of 0.1% for each N (for the $(K^{(\mathrm{a})}, Q)$ Ritz approximation) is shown in figure 15.8 for both the regular and deep-cut ellipsoid algorithms. (For $N < 4$ the step response overshoot constraint was infeasible in the $(K^{(\mathrm{a})}, Q)$ Ritz approximation.)

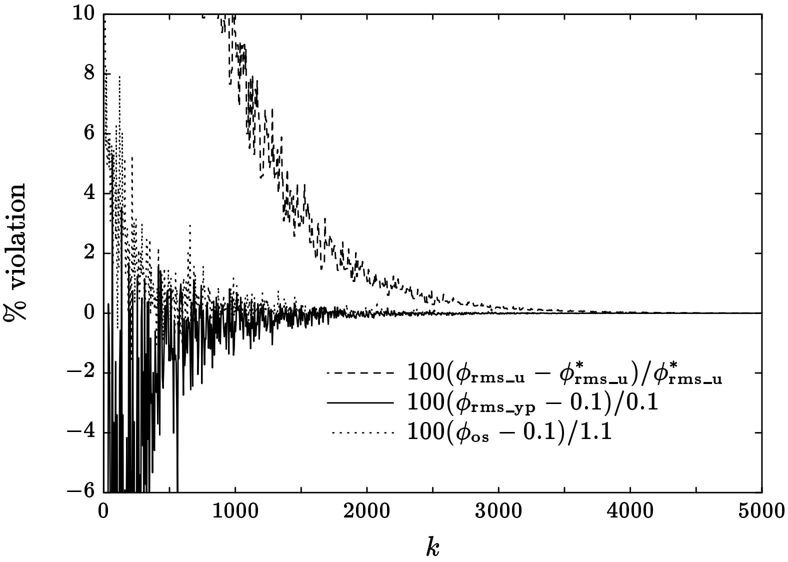


Figure 15.6 The ellipsoid algorithm constraint and objective functionals versus iteration number, k, for the solution of the $N = 20$ $(K^{(\rm a)}, Q)$ Ritz approximation of (15.12). For the constraints, the percentage violation is plotted. For the objective, $\phi_{\rm rms\_u}$, the percentage difference between the current and final objective value, $\phi^*_{\rm rms\_u}$, is plotted. It is hard to distinguish the plots for the constraints, but the important point here is the "steady, stochastic" nature of the convergence. Note that within 1500 iterations, designs were obtained with objective values and contraints within a few percent of optimal and feasible, repsectively.

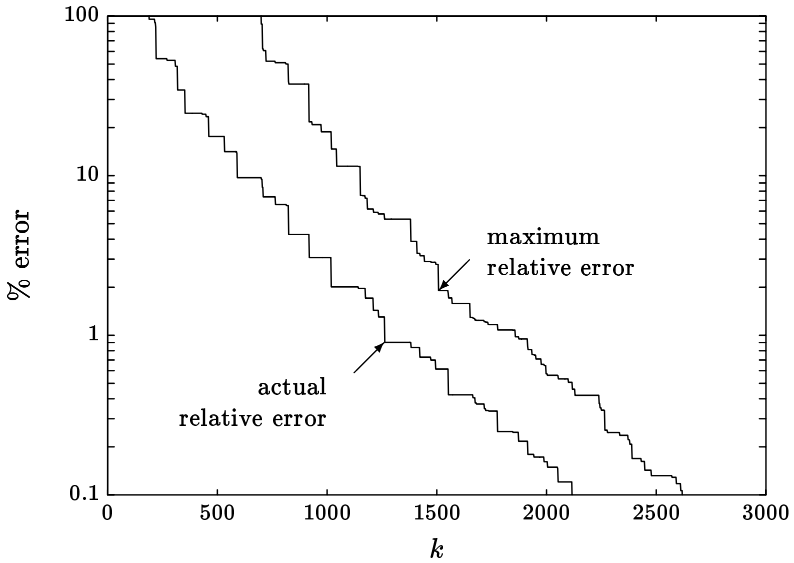


Figure 15.7 The deep-cut ellipsoid algorithm maximum and actual relative errors versus iteration number, k, for the solution of the $N = 20$ $(K^{(\mathrm{a})}, Q)$ Ritz approximation of (15.12). After 53 iterations a feasible point has been found, and after 2620 iterations the objective has been computed to a maximum relative error of 0.1%. Note the similarity to figure 15.5.

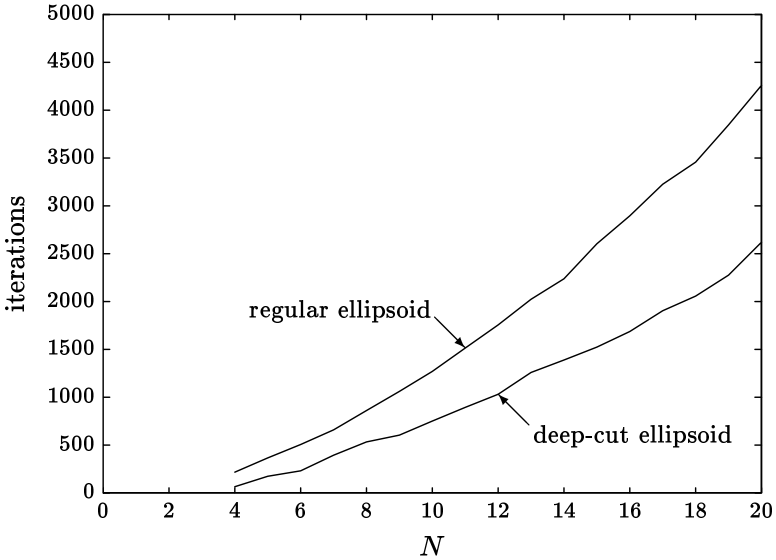


Figure 15.8 The number of ellipsoid iterations required to compute upper bounds of $\mathrm{RMS}(u)$ to within 0.1%, versus the number of terms N, is shown for the Ritz approximation $(K^{(\mathrm{a})}, Q)$. The upper curve shows the iterations for the regular ellipsoid algorithm. The lower curve shows the iterations for the deep-cut ellipsoid algorithm, using deep-cuts for both objective and constraint cuts.

15.4 An Outer Approximation via Duality

Figure 15.4 suggests that the minimum value of the problem (15.12) is close to 0.058, since several different Ritz approximations appear to be converging to this value, at least over the small range of N plotted. The value 0.058 could reasonably be accepted on this basis alone.

To further strengthen the plausibility of this conclusion, we could appeal to some convergence theorem (one is given in the Notes and References). But even knowing that each of the four curves shown in figure 15.4 converges to the exact value of the problem (15.12), we can only assert with certainty that the optimum value lies between 0.0397 (the true optimum of (15.9)) and 0.0585 (the lowest objective value computed with a Ritz approximation).

This is a problem of *stopping criterion*, which we discussed in chapter 14 in the context of optimization algorithms; accepting 0.058 as the optimum value of (15.12) corresponds to a (quite reasonable) heuristic stopping criterion. Just as in chapter 14, however, a stopping criterion that is based on a known lower bound may be worth the extra computation involved.

In this section we describe a method for computing lower bounds on the value of (15.12), by forming an appropriate dual problem. Unlike the stopping criteria for the algorithms in chapter 14, which involve little or no additional computation, the lower bound computations that we will describe require the solution of an auxiliary minimum $\mathbf{H}_2$ norm problem.

The dual function introduced in section 3.6.2 produces lower bounds on the solution of (15.12), but is not useful here since we cannot exactly evaluate the dual function, except by using the same approximations that we use to approximately solve (15.12). To form a dual problem that we can solve requires some manipulation of the problem (15.12) and a generalization of the dual function described in section 3.6.2. The generalization is easily described informally, but a careful treatment is beyond the scope of this book; see the Notes and References at the end of this chapter.

We replace the RMS actuator effort objective and the RMS regulation constraint in (15.12) with the corresponding variance objective and constraint, *i.e.*, we square the objective and constraint functionals $\phi_{\mathrm{rms\_u}}$ and $\phi_{\mathrm{rms\_yp}}$. Instead of considering the step response overshoot constraint in (15.12) as a single functional inequality, we will view it as a family of constraints on the step response: one constraint for each $t \geq 0$, that requires the step response at time t not exceed 1.1:

$$\begin{array}{cl} \min & \phi_{\mathrm{rms\_u}}(H)^2 \\ \phi_{\mathrm{rms\_yp}}(H)^2 \leq 0.1^2 & \\ \phi^{\mathrm{step},t}(H) \leq 1.1, \quad t \geq 0 & \end{array} \tag{15.13}$$

where $\phi^{\mathrm{step},t}$ is the affine functional that evaluates the step response of the 1,3 entry at time t:

$$\phi^{\mathrm{step},t}(H) = s_{13}(t).$$

In this transformed optimization problem, the objective is quadratic and the constraints consist of one quadratic constraint along with a family of affine constraints that is parametrized by $t \in \mathbf{R}_+$.

This suggests the following generalized *dual functional* for the problem (15.13):

$$\psi(\lambda_u, \lambda_y, \lambda_s) \stackrel{\Delta}{=} \min_{H \in \mathcal{H}} \left(\lambda_u \phi_{\rm rms\_u}(H)^2 + \lambda_y \phi_{\rm rms\_yp}(H)^2 + \int_0^\infty \lambda_s(t) \phi^{{\rm step},t}(H)\, dt \right), \quad (15.14)$$

where the "weights" now consist of the positive numbers λ_u and λ_y, along with the *function* $\lambda_s : \mathbf{R}_+ \to \mathbf{R}_+$. So in equation (3.10), we have replaced the weighted sum of (a finite number of) constraint functionals with a weighted integral over the family of constraint functionals that appears in (15.13).

It is easily established that $-\psi$ is a convex functional of $(\lambda_u, \lambda_y, \lambda_s)$ and that whenever $\lambda_y \geq 0$ and $\lambda_s(t) \geq 0$ for all $t \geq 0$, we have

$$\psi(1, \lambda_y, \lambda_s) - 0.1^2 \lambda_y - \int_0^\infty 1.1 \lambda_s(t)\, dt \leq \alpha_{\rm pri}$$

where $\alpha_{\rm pri}$ is the optimum value of (15.12). Thus, computing $\psi(1, \lambda_y, \lambda_s)$ yields a lower bound on the optimum of (15.12).

The convex duality principle (equations (6.11–6.12) of section 6.6) suggests that we actually have

$$\alpha_{\rm pri} = \max_{\lambda_y \geq 0,\ \lambda_s(t) \geq 0} \left(\psi(1, \lambda_y, \lambda_s) - 0.1^2 \lambda_y - \int_0^\infty 1.1 \lambda_s(t)\, dt \right), \quad (15.15)$$

which in fact is true. So the optimization problem on the right-hand side of (15.15) can be considered a dual of (15.12).

We can compute $\psi(1, \lambda_y, \lambda_s)$, provided we have a state-space realization with impulse response $\lambda_s(t)$. The objective in (15.14) is an LQG objective, with the addition of the integral term, which is an *affine* functional of H. By completing the square, it can be recast as an $\mathbf{H}_2$-optimal controller problem, and solved by (an extension of) the method described in section 12.2. Since we can find a minimizer H_λ^*, for the problem (15.14), we can evaluate a subgradient for $-\psi$:

$$\phi^{\rm sg}(\lambda_u, \lambda_y, \lambda_s) = -\lambda_u \phi_{\rm rms\_u}(H_\lambda^*)^2 - \lambda_y \phi_{\rm rms\_yp}(H_\lambda^*)^2 - \int_0^\infty \lambda_s(t) \phi^{{\rm step},t}(H_\lambda^*)\, dt$$

(*c.f.* section 13.4.8).

By applying a Ritz approximation to the infinite-dimensional optimization problem (15.15), we obtain *lower bounds* on the right-hand, and hence left-hand sides of (15.15).

To demonstrate this, we use two Ritz sequences for λ_s given by the transfer functions

$$\Lambda_0 = 0, \quad \Lambda_i(s) = \left(\frac{2}{s+2}\right)^i, \quad \tilde{\Lambda}_i(s) = \left(\frac{4}{s+4}\right)^i, \quad i = 1, \ldots. \qquad (15.16)$$

We shall denote these two Ritz approximations Λ, $\tilde{\Lambda}$. The solution of the finite-dimensional inner approximation of the dual problem (15.15) is shown in figure 15.9 for various values of N. Each curve gives lower bounds on the solution of (15.12).

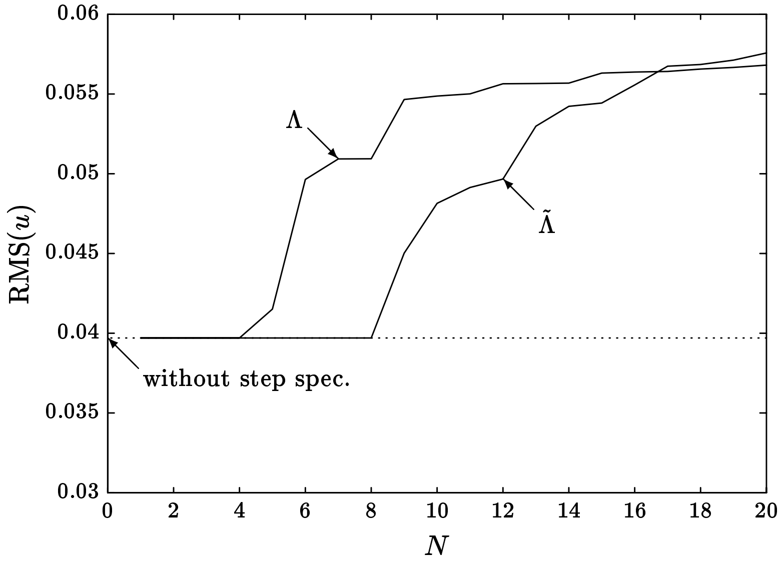


Figure 15.9 The optimum value of the finite-dimensional inner approximation of the dual problem (15.15) versus the number of terms N for the two different Ritz approximations (15.16). The dotted line shows the optimum value of RMS(u) without the step response overshoot specification.

The upper bounds from figure 15.4 and lower bounds from figure 15.9 are shown together in figure 15.10. The exact solution of (15.12) is known to lie between the dashed lines; this band is shown in figure 15.11 on a larger scale for clarity.

The best lower bound on the value of (15.12) is 0.0576, corresponding to the $N = 20$ $\{\tilde{\Lambda}\}$ Ritz approximation of (15.15). The best upper bound on the value of (15.12) is 0.0585, corresponding to the $N = 20$ $(K^{(a)}, Q)$ Ritz approximation of (15.12). Thus we can state with certainty that

$$.0576 \leq \min_{\substack{\phi_{\rm rms\_yp}(H) \leq 0.1 \\ \phi_{\rm os}(H_{13}) \leq 0.1}} \phi_{\rm rms\_u}(H) \leq .0585.$$

We now *know* that 0.058 is within 0.0005 (*i.e.*, 1%) of the minimum value of the problem (15.12); the plots in figure 15.4 only strongly hint that this is so.

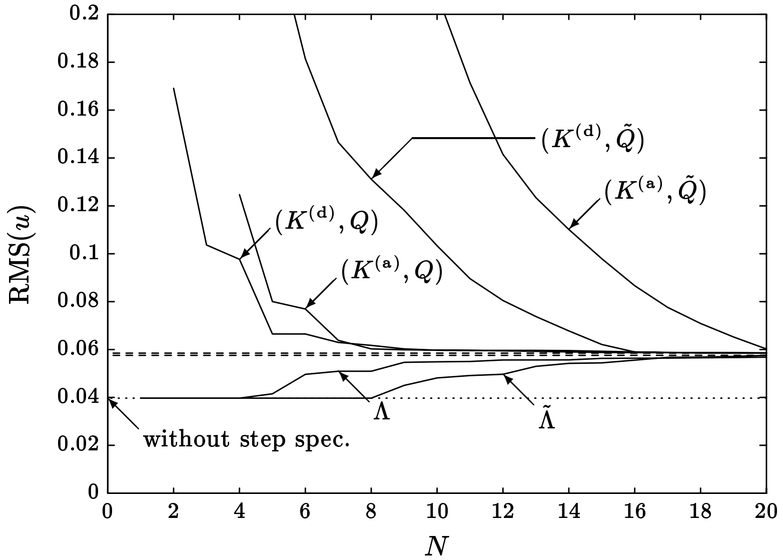


Figure 15.10 The curves from figures 15.4 and 15.9 are shown. The solution of each finite-dimensional inner approximation of (15.12) gives an *upper bound* of the exact solution. The solution of each finite-dimensional inner approximation of the dual problem (15.15) gives a *lower bound* of the exact solution. The exact solution is therefore known to lie between the dashed lines.

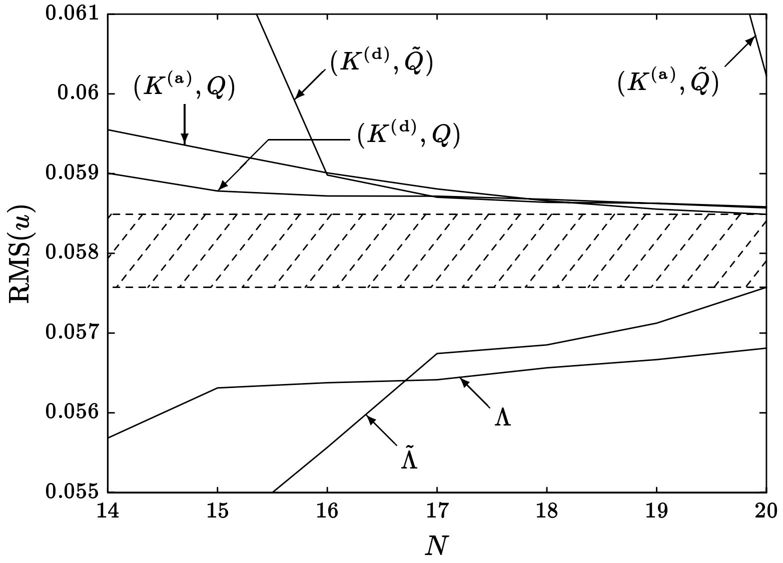


Figure 15.11 Figure 15.10 is shown in greater detail. The exact solution of (15.12) is known to lie inside the shaded band.

We should warn the reader that we do not know how to form a solvable dual problem for the most general convex controller design problem; see the Notes and References.

15.5 Some Tradeoff Curves

In the previous three sections we studied two specific optimization problems, and the performance of several approximate solution methods. In this section we consider some related two-parameter families of design specifications and the same approximate solution methods, concentrating on the effect of the approximations on the computed region of achievable specifications in performance space.

15.5.1 Tradeoff for Example with Analytic Solution

We consider the family of design specifications given by

$$\phi_{\mathrm{rms\_yp}}(H) \le \alpha, \qquad \phi_{\mathrm{rms\_u}}(H) \le \beta, \tag{15.17}$$

for the same example that we have been studying.

Figure 15.12 shows the tradeoff curves for the $(K^{(\mathrm{a})}, Q)$ Ritz approximations of (15.17), for three values of N. The regions above these curves are thus $\mathcal{A}_N$; $\mathcal{A}$ is the region above the solid curve, which is the exact tradeoff curve. This figure makes clear the nomenclature “inner approximation”.

15.5.2 Tradeoff for Example with no Analytic Solution

We now consider the family of design specifications given by

$$\mathcal{D}^{(\alpha,\beta)} \quad : \quad \phi_{\mathrm{rms\_yp}}(H) \le \alpha, \quad \phi_{\mathrm{rms\_u}}(H) \le \beta, \quad \phi_{\mathrm{os}}(H_{13}) \le 0.1. \tag{15.18}$$

Inner and outer approximations for the tradeoff curve for (15.18) can be computed using Ritz approximations to the primal and dual problems. For example, an $N = 6$ $(K^{(\mathrm{d})}, Q)$ Ritz approximation to (15.18) shows that the specifications in the top right region in figure 15.13 are achievable. On the other hand, an $N = 5$ Λ Ritz approximation to the dual of (15.18) shows that the specifications in the bottom left region in figure 15.13 are unachievable. Therefore the exact tradeoff curve is known to lie in the shaded region in figure 15.14.

The inner and outer approximations shown in figure 15.14 can be improved by solving larger finite-dimensional approximations to (15.18) and its dual, as shown in figure 15.15.

Figure 15.15 shows the tradeoff curve for (15.17), for comparison. The gap between this curve and the shaded region shows the cost of the additional step response specification. (The reader should compare these curves with those of figure 14.18 in section 14.5, which shows the cost of the additional specification $\mathrm{RMS}(\dot{y}_{\mathrm{p}}) \le 0.04$ on the family of design specifications (15.17).)

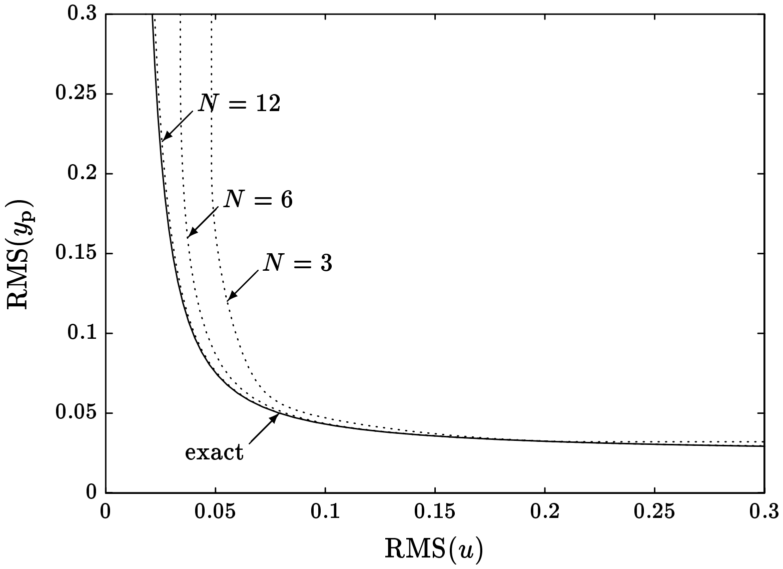


Figure 15.12 The exact tradeoff between RMS(y_p) and RMS(u) for the problem (15.17) can be computed using LQG theory. The tradeoff curves computed using three $(K^{(a)}, Q)$ Ritz inner approximations are also shown.

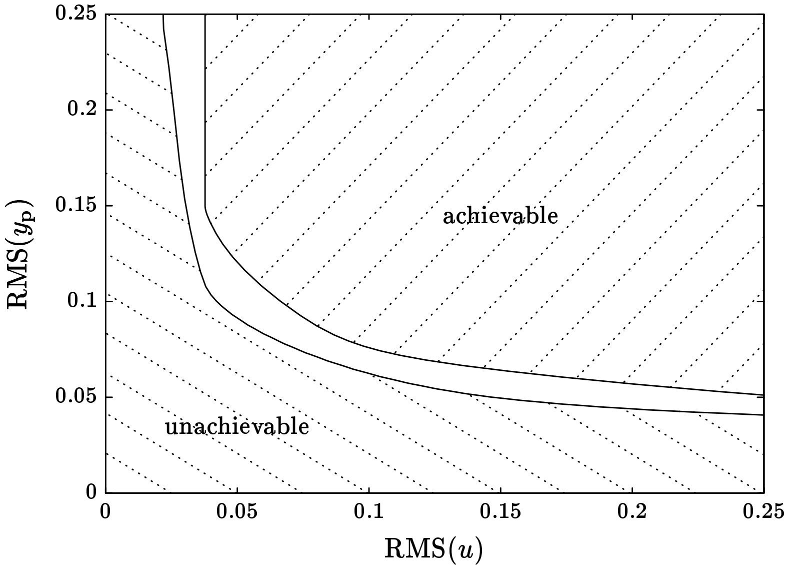


Figure 15.13 With the specification $\phi_{os}(H_{13}) \leq 0.1$, specifications on ϕ_{rms_yp} and ϕ_{rms_u} in the upper shaded region are shown to be achievable by solving an $N = 6$ $(K^{(d)}, Q)$ finite-dimensional approximation of (15.18). Specifications in the lower shaded region are shown to be unachievable by solving an $N = 5$ Λ finite-dimensional approximation of the dual of (15.18).

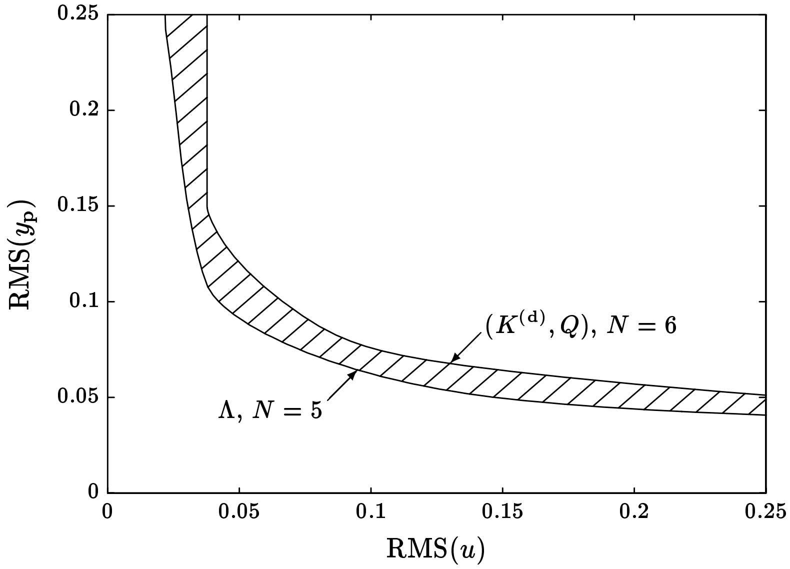


Figure 15.14 From figure 15.13 we can conclude that the exact tradeoff curve lies in the shaded region.

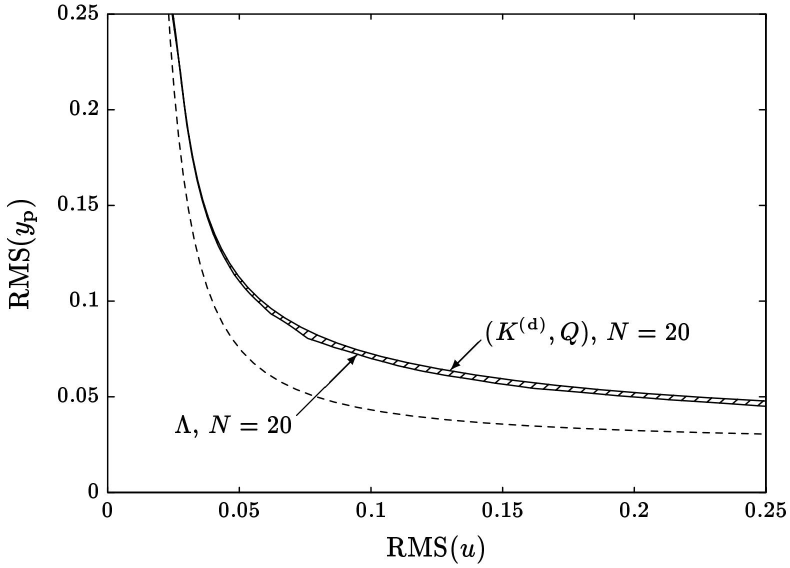


Figure 15.15 With larger finite-dimensional approximations to (15.18) and its dual, the bounds on the achievable limit of performance are substantially improved. The dashed curve is the boundary of the region of achievable performance without the step response overshoot constraint.

Notes and References

Ritz Approximations

The term comes from Rayliegh-Ritz approximations to infinite-dimensional eigenvalue problems; see, *e.g.*, Courant and Hilbert [CH53, P175]. The topic is treated in, *e.g.*, section 3.7 of Daniel [DAN71]. A very clear discussion of Ritz methods, and their convergence properties, appears in sections 8 and 9 of the paper by Levitin and Polyak [LP66].

Proof that the Example Ritz Approximations Converge

It is often possible to *prove* that a Ritz approximation "works", *i.e.*, that $\mathcal{A}_N \to \mathcal{A}$ (in some sense) as $N \to \infty$. As an example, we give a complete proof that for the four Ritz approximations (15.11) of the controller design problem with objectives RMS actuator effort, RMS regulation, and step response overshoot, we have

$$\overline{\bigcup_N \mathcal{A}_N} \supseteq \mathcal{A}. \tag{15.19}$$

This means that every achievable specification that is not on the boundary (*i.e.*, not Pareto optimal) can be achieved by a Ritz approximation with large enough N.

We first express the problem in terms of the parameter Q in the free parameter representation of $\mathcal{D}_{\mathrm{stable}}$:

$$\begin{aligned}
\psi_1(Q) &\stackrel{\Delta}{=} \phi_{\mathrm{rms\_yp}}(T_1 + T_2QT_3) = \|\tilde{G}_1 + G_1Q\|_2,\\
\psi_2(Q) &\stackrel{\Delta}{=} \phi_{\mathrm{rms\_u}}(T_1 + T_2QT_3) = \|\tilde{G}_2 + G_2Q\|_2,\\
\psi_3(Q) &\stackrel{\Delta}{=} \phi_{\mathrm{os}}([1\ 0](T_1 + T_2QT_3)[0\ 0\ 1]^T) = \|\tilde{G}_3 + G_3Q\|_{\mathrm{pk\_step}} - 1,
\end{aligned}$$

where the $\tilde{G}_i$ depend on submatrices of T_1, and the G_i depend on the appropriate submatrices of T_2 and T_3. (These transfer matrices incorporate the constant power spectral densities of the sensor and process noises, and combine the T_2 and T_3 parts since Q is scalar.) These transfer matrices are stable and rational. We also have $G_3(0) = 0$, since P_0 has a pole at $s = 0$.

We have

$$\begin{aligned}
\mathcal{A} &= \{a \mid \psi_i(Q) \le a_i,\ i = 1,2,3,\ \text{for some } Q \in \mathbf{H}_2\},\\
\mathcal{A}_N &= \left\{ a \;\middle|\; \psi_i(Q) \le a_i,\ i = 1,2,3,\ \text{for some } Q = \sum_{i=1}^{N} x_iQ_i \right\}
\end{aligned}$$

($\mathbf{H}_2$ is the Hilbert space of Laplace transforms of square integrable functions from $\mathbf{R}_+$ into $\mathbf{R}$).

We now observe that ψ_1, ψ_2, and ψ_3 are continuous functionals on $\mathbf{H}_2$; in fact, there is an $M < \infty$ such that

$$\left|\psi_i(Q) - \psi_i(\tilde{Q})\right| \le M\|Q - \tilde{Q}\|_2, \qquad i = 1,2,3. \tag{15.20}$$

This follows from the inequalities

$$\begin{aligned}
\left|\psi_1(Q) - \psi_1(\tilde{Q})\right| &\le \|G_1\|_\infty \|Q - \tilde{Q}\|_2,\\
\left|\psi_2(Q) - \psi_2(\tilde{Q})\right| &\le \|G_2\|_\infty \|Q - \tilde{Q}\|_2,\\
\left|\psi_3(Q) - \psi_3(\tilde{Q})\right| &\le \|G_3/s\|_2 \|Q - \tilde{Q}\|_2.
\end{aligned}$$

(The first two are obvious; the last uses the general fact that $\|AB\|_{\mathrm{pk\_step}} \le \|A/s\|_2 \|B\|_2$, which follows from the Cauchy-Schwarz inequality.)

We now observe that the sequence $(s+\alpha)^{-i}$, $i = 1, 2, \ldots$, where $\alpha > 0$, is complete, *i.e.*, has dense span in $\mathbf{H}_2$. In fact, if this sequence is orthonormalized we have the Laplace transforms of the *Laguerre functions* on $\mathbf{R}_+$.

We can now prove (15.19). Suppose $a \in \mathcal{A}$ and $\epsilon > 0$. Since $a \in \mathcal{A}$, there is a $Q^* \in \mathbf{H}_2$ such that $\psi_i(Q^*) \le a_i$, $i = 1, 2, 3$. Using completeness of the Q_i's, find N and $x^* \in \mathbf{R}^N$ such that

$$\|Q^* - Q_N^*\|_2 \le \epsilon/M,$$

where

$$Q_N^* \overset{\Delta}{=} \sum_{i=1}^{N} x_i^* Q_i.$$

By (15.20), $\psi_i(Q_N^*) \le a_i + \epsilon$, $i = 1, 2, 3$. This proves (15.19).

The Example Ritz Approximations

We used the state-space parametrization in section 7.4, with

$$P_0^{\mathrm{std}}(s) = C(sI - A)^{-1}B,$$

where

$$A = \begin{bmatrix} -10 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & -1 & 10 \end{bmatrix}.$$

The controllers $K^{(\mathrm{a})}$ and $K^{(\mathrm{b})}$ are estimated-state-feedback controllers with

$$L_{\mathrm{est}}^{(\mathrm{a})} = \begin{bmatrix} 5.20000 \\ -2.08000 \\ -0.80800 \end{bmatrix}, \qquad K_{\mathrm{sfb}}^{(\mathrm{a})} = \begin{bmatrix} -6.00000 & 5.25000 & 2.50000 \end{bmatrix},$$

$$L_{\mathrm{est}}^{(\mathrm{d})} = \begin{bmatrix} 0.00000 \\ -3.16228 \\ -1.11150 \end{bmatrix}, \qquad K_{\mathrm{sfb}}^{(\mathrm{d})} = \begin{bmatrix} 1.42758 & 10.29483 & 2.44949 \end{bmatrix}.$$

Problem with Analytic Solution

With the Ritz approximations, the problem has a single convex quadratic constraint, and a convex quadratic objective, which are found by solving appropriate Lyapunov equations. The resulting optimization problems are solved using a standard method that is described in, *e.g.*, Golub and Van Loan [GL89, P564–566]. Of course an ellipsoid or cutting-plane algorithm could also be used.

Dual Outer Approximation for Linear Controller Design

General duality in infinite-dimensional optimization problems is treated in the book by Rockafellar [ROC74], which also has a complete reference list of other sources covering this material in detail. See also the book by Anderson and Nash [AN87] and the paper by Reiland [REI80].

As far as we know, the idea of forming finite-dimensional outer approximations to a convex linear controller design problem, by Ritz approximation of an appropriate dual problem, is new.

The method can be applied when the functionals are quadratic (*e.g.*, weighted $\mathbf{H}_2$ norms of submatrices of H), or involve envelope constraints on time domain responses. We do not know how to form a solvable dual problem of a general convex controller design problem.

Chapter 16

Discussion and Conclusions

We summarize the main points that we have tried to make, and we discuss some applications and extensions of the methods described in this book, as well as some history of the main ideas.

16.1 The Main Points

- *An explicit framework.* A sensible formulation of the controller design problem is possible only by considering simultaneously *all* of the closed-loop transfer functions of interest, *i.e.*, the closed-loop transfer matrix H, which should include every closed-loop transfer function necessary to evaluate a candidate design.
- *Convexity of many specifications.* The set of transfer matrices that meet a design specification often has simple geometry—affine or convex. In many other cases it is possible to form convex inner (conservative) approximations.
- *Effectiveness of convex optimization.* Many controller design problems can be cast as convex optimization problems, and therefore can be "efficiently" solved.
- *Numerical methods for performance limits.* The methods described in this book can be used both to design controllers (via the primal problem) and to find the limits of performance (via the dual problem).

16.2 Control Engineering Revisited

In this section we return to the broader topic of control engineering. Some of the major tasks of control engineering are shown in figure 16.1:

- The system to be controlled, along with its sensors and actuators, is modeled as the plant P.
- Vague goals for the behavior of the closed-loop system are formulated as a set of design specifications (chapters 8–10).
- If the plant is LTI and the specifications are closed-loop convex, the resulting feasibility problem can be solved (chapters 13–15).
- If the specifications are achievable, the designer will check that the design is satisfactory, perhaps by extensive simulation with a detailed (probably nonlinear) model of the system.

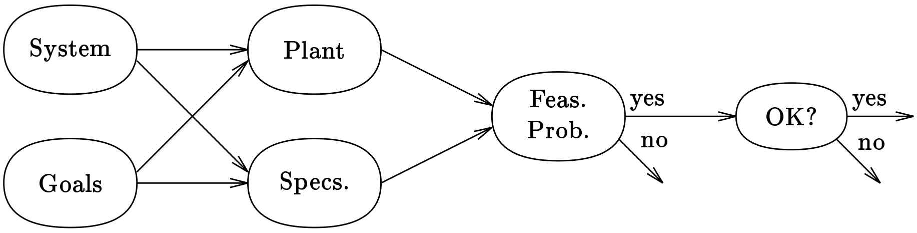


Figure 16.1 A partial flowchart of the control engineer's tasks.

One design will involve many iterations of the steps shown in figure 16.1. We now discuss some possible design iterations.

Modifying the Specifications

The specifications are weakened if they are infeasible, and possibly tightened if they are feasible, as shown in figure 16.2. This iteration may take the form of a search over Pareto optimal designs (chapter 3).

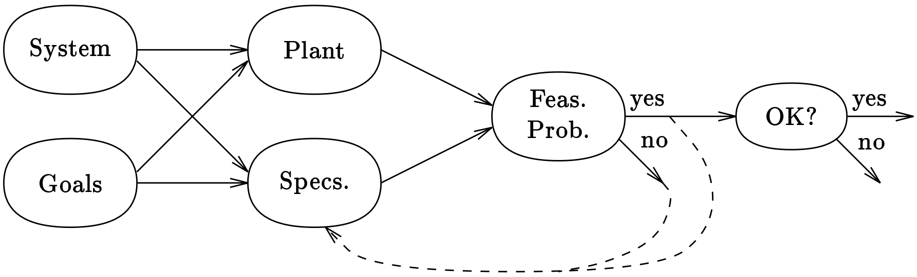


Figure 16.2 Based on the outcome of the feasibility problem, the designer may decide to modify (*e.g.*, tighten or weaken) some of the specifications.

Modifying the Control Configuration

Based on the outcome of the feasibility problem, the designer may modify the choice and placement of the sensors and actuators, as shown in figure 16.3. If the specifications are feasible, the designer might remove actuators and sensors to see if the specifications are still feasible; if the specifications are infeasible, the designer may add or relocate actuators and sensors until the specifications become achievable. The value of *knowing* that a given set of design specifications cannot be achieved with a given configuration should be clear.

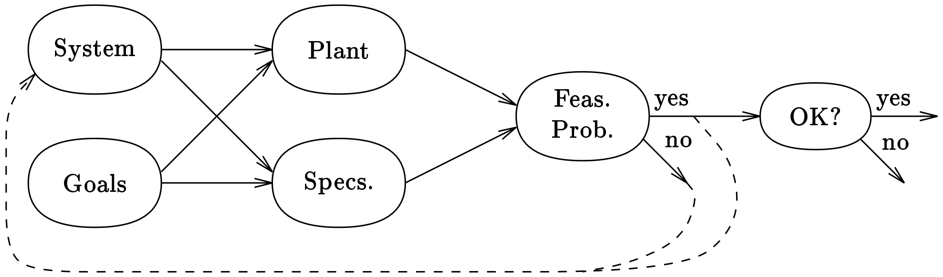


Figure 16.3 Based on the outcome of the feasibility problem, the designer may decide to add or remove sensors or actuators.

These iterations can take a form that is analogous to the iteration described above, in which the specifications are modified. We consider a fixed set of specifications, and a *family* (which is usually finite) of candidate control configurations. Figure 16.4 shows fourteen possible control configurations, each of which consists of some selection among the two potential actuators A_1 and A_2 and the three sensors S_1, S_2, and S_3. (These are the configurations that use at least one sensor, and one, but not both, actuators. A_1 and A_2 might represent two candidate motors for a system that can only accommodate one.) These control configurations are partially ordered by inclusion; for example, A_1S_1 consists of deleting the sensor S_3 from the configuration $A_1S_1S_3$.

These different control configurations correspond to different plants, and therefore different feasibility problems, some of which may be feasible, and others infeasible. One possible outcome is shown in figure 16.5: nine of the configurations result in the specifications being feasible, and five of the configurations result in the specifications being infeasible. In the iteration described above, the designer could choose among the achievable specifications; here, the designer can choose among the control configurations that result in the design specification being feasible. Continuing the analogy, we might say that $A_1S_1S_2$ is a Pareto optimal control configuration, on the boundary between feasibility and infeasibility.

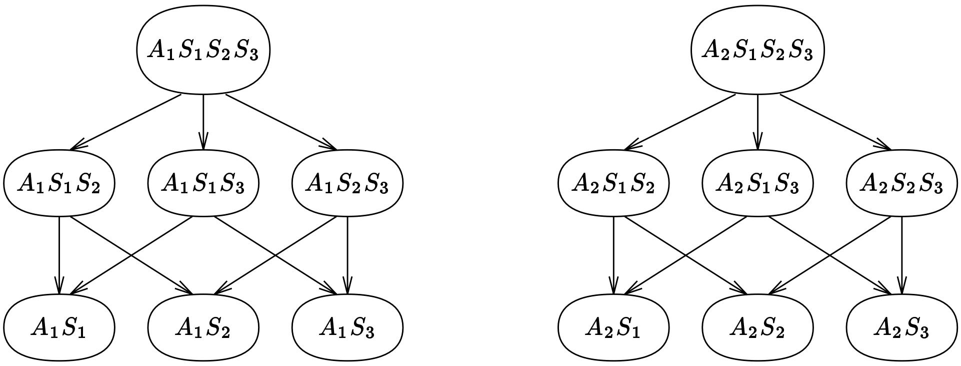


Figure 16.4 The possible actuator and sensor configurations, with the partial ordering induced by achievability of a set of specifications.

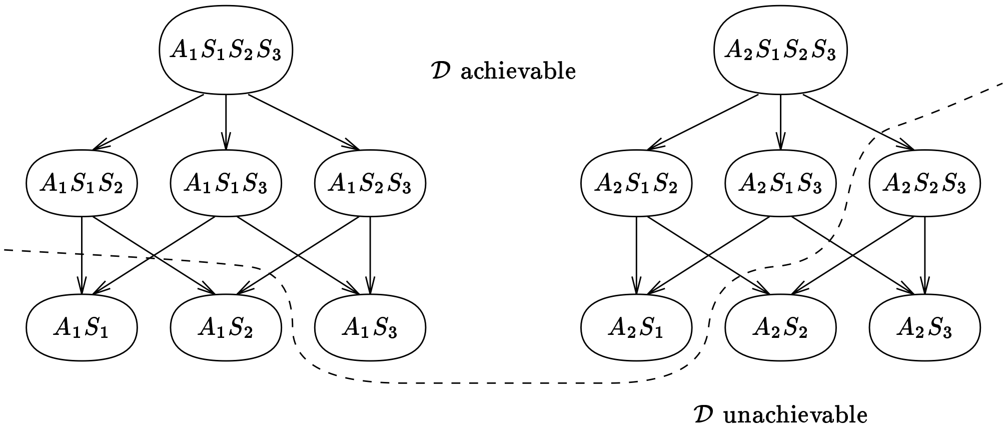


Figure 16.5 The actuator and sensor configurations that can meet the specification $\mathcal{D}$.

Modifying the Plant Model and Specifications

After choosing an achievable set of design specifications, the design is verified: does the controller, designed on the basis of the LTI model P and the design specifications $\mathcal{D}$, achieve the original goals when connected in the real closed-loop system? If the answer is no, the plant P and design specifications $\mathcal{D}$ have failed to accurately represent the original system and goals, and must be modified, as shown in figure 16.6.

Perhaps some unstated goals were not included in the design specifications. For example, if some critical signal is too big in the closed-loop system, it should be added to the regulated variables signal, and suitable specifications added to $\mathcal{D}$, to constrain the its size.

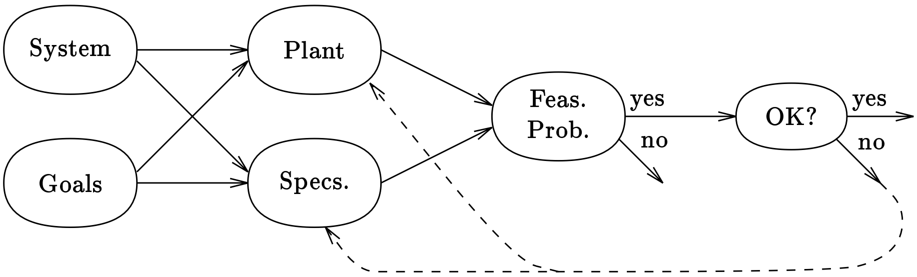


Figure 16.6 If the outcome of the feasibility problem is inconsistent with designer's criteria then the plant and specifications must be modified to capture the designer's intent.

As a specific example, the controller designed in the rise time versus undershoot tradeoff example in section 12.4, would probably be unsatisfactory, since our design specifications did not constrain actuator effort. This unsatisfactory aspect of the design would not be apparent from the specifications—indeed, our design cannot be greatly improved in terms of rise time or undershoot. The excessive actuator effort would become apparent during design verification, however. The solution, of course, is to add an appropriate specification that limits actuator effort.

An achievable design might also be unsatisfactory because the LTI plant P is not a sufficiently good model of the system to be controlled. Constraining various signals to be smaller may improve the accuracy with which the system can be modeled by an LTI P; adding appropriate robustness specifications (chapter 10) may also help.

16.3 Some History of the Main Ideas

16.3.1 Truxal's Closed-Loop Design Method

The idea of first designing the closed-loop system and then determining the controller required to achieve this closed-loop system is at least forty years old. An explicit presentation of a such a method appears in Truxal's 1950 Ph.D. thesis [TRU50], and chapter 5 of Truxal's 1955 book, *Automatic Feedback Control System Synthesis*, in which we find [TRU55, P279]:

> Guillemin in 1947 proposed that the synthesis of feedback control systems take the form ...
>
> 1. The closed-loop transfer function is determined from the specifications.
> 2. The corresponding open-loop transfer function is found.
> 3. The appropriate compensation networks are synthesized.

Truxal cites his Ph.D. thesis and a 1951 article by Aaron [AAR51].

On the difference between classical controller synthesis and the method he proposes, he states ([TRU55, P278–279]):

> The word *synthesis* rigorously implies a logical procedure for the transition from specifications to system. In pure synthesis, the designer is able to take the specifications and in a straightforward path proceed to the final system. In this sense, neither the conventional methods of servo design nor the root locus method is pure synthesis, for in each case the designer attempts to modify and to build up the open-loop system until he has reached a point where the system, after the loop is closed, will be satisfactory.
>
> ... [The closed-loop design] approach to the synthesis of closed-loop systems represents a complete change in basic thinking. No longer is the designer working inside the loop and trying to splice things up so that the overall system will do the job required. On the contrary, he is now saying, "I have a certain job that has to be done. I will force the system to do it."

So Truxal views his closed-loop controller design method as a "more logical synthesis pattern" (P278) than classical methods. He does not extensively justify this view, except to point out the simple relation between the classical error constants and the closed-loop transfer function (P281). (In chapter 8 we saw that the classical error constants are affine functionals of the closed-loop transfer matrix.)

The closed-loop design method is described in the books [NGK57], [RF58, CH7], [HOR63, §5.12], and [FPW90, §5.7] (see also the Notes and References from chapter 7 on the interpolation conditions).

16.3.2 Fegley's Linear and Quadratic Programming Approach

The observation that some controller design problems can be solved by numerical optimization that involves closed-loop transfer functions is made in a series of papers starting in 1964 by Fegley and colleagues. In [FEG64] and [FH65], Fegley applies linear programming to the closed-loop controller design approach, incorporating such specifications as asymptotic tracking of a specific command signal and an overshoot limit. This method is extended to use quadratic programming in [PF66, BF68]. In [CF68] and [MF71], specifications on RMS values of signals are included. A summary of most of the results of Fegley and his colleagues appears in [FBB71], which includes examples such as a minimum variance design with a step response envelope constraint. This paper has the summary:

> Linear and quadratic programming are applicable ... to the design of control systems. The use of linear and quadratic programming frequently represents the easiest approach to an optimal solution and often

> makes it possible to impose constraints that could not be imposed in other methods of solution.

So, several important ideas in this book appear in this series of papers by Fegley and colleagues: designing the closed-loop system directly, noting the restrictions placed by the plant on the achievable closed-loop system; expressing performance specifications as closed-loop convex constraints; and using numerical optimization to solve problems that do not have an analytical solution (see the quote above).

Several other important ideas, however, do not appear in this series of papers. Convexity is never mentioned as the property of the problems that makes effective solution possible; linear and quadratic programming are treated as useful "tools" which they "apply" to the controller design problem. The casual reader might conclude that an extension of the method to indefinite (nonconvex) quadratic programming is straightforward, and might allow the designer to incorporate some other useful specifications. This is *not* the case: numerically solving nonconvex QP's is vastly more difficult than solving convex QP's (see section 14.6.1).

Another important idea that does not appear in the early literature on the closed-loop design method is that it can potentially search over *all* possible LTI controllers, whereas a classical design method (or indeed, a modern state-space method) searches over a restricted (but often adequate) set of LTI controllers. Finally, this early form of the closed-loop design method is restricted to the design of *one* closed-loop transfer function, for example, from command input to system output.

16.3.3 Q-Parameter Design

The closed-loop design method was first extended to MAMS control systems (*i.e.*, by considering closed-loop transfer *matrices* instead of a particular transfer function), in a series of papers by Desoer and Chen [DC81A, DC81B, CD82B, CD83] and Gustafson and Desoer [GD83, DG84B, DG84A, GD85]. These papers emphasize the design of controllers, and not the determination that a set of design specifications cannot be achieved by any controller.

In his 1986 Ph. D. thesis, Salcudean [SAL86] uses the parametrization of achievable closed-loop transfer matrices described in chapter 7 to formulate the controller design problem as a constrained convex optimization problem. He describes many of the closed-loop convex specifications we encountered in chapters 8–10, and discusses the importance of convexity. See also the article by Polak and Salcudean [PS89].

The authors of this book and colleagues have developed a program called QDES, which is described in the article [BBB88]. The program accepts input written in a *control specification language* that allows the user to describe a discrete-time controller design problem in terms of many of the closed-loop convex specifications presented in this book. A simple method is used to approximate the controller design problem as a finite-dimensional linear or quadratic programming problem, which is then solved. The simple organization and approximations made in QDES make it practical only for small problems. The paper by Oakley and Barratt [OB90] describes the use of QDES to design a controller for a flexible mechanical structure.

16.3.4 FIR Filter Design via Convex Optimization

A relevant parallel development took place in the area of digital signal processing. In about 1969, several researchers observed that many *finite impulse response* (FIR) filter design problems could be cast as linear programs; see, for example, the articles [CRR69, RAB72] or the books by Oppenheim and Schaefer [OS70, §5.6] and Rabiner and Gold [RG75, CH.3]. In [RG75, §3.39] we even find designs subject to both time and frequency domain specifications:

> Quite often one would like to impose simultaneous restrictions on both the time and frequency response of the filter. For example, in the design of lowpass filters, one would often like to limit the step response overshoot or ripple, at the same time maintaining some reasonable control over the frequency response of the filter. Since the step response is a linear function of the impulse response coefficients, a linear program is capable of setting up constraints of the type discussed above.

A recent article on this topic is [OKU88].

Like the early work on the closed-loop design method, convexity is not recognized as the property of the FIR filter design problem that allows efficient solution. Nor is it noted that the method actually computes the global optimum, *i.e.*, if the method fails to design an FIR filter that meets some set of convex specifications, then the specifications cannot be achieved by any FIR filter (of that order).

16.4 Some Extensions

16.4.1 Discrete-Time Plants

Essentially all of the material in this book applies to single-rate discrete-time plants and controllers, provided the obvious changes are made (*e.g.*, redefining stability to mean no poles on or outside the unit disk). For a discrete-time development, there is a natural choice of stable transfer matrices that can be used to form the (analog of the) Ritz sequence (15.8) described in section 15.1:

$$Q_{ijk}(z) = E_{ij} z^{-(k-1)}, \qquad 1 \le i \le n_u,\ 1 \le j \le n_y,\ k = 1, 2, \ldots,$$

(E_{ij} is the matrix with a unit i, j entry, and all other entries zero), which corresponds to a delay of $k-1$ time steps, from the jth input of Q to its ith output. Thus in the Ritz approximation, the entries of the transfer matrix Q are polynomials in z^{-1}, *i.e.*, FIR filters. This approach is taken in the program QDES [BBB88].

Many of the results can be extended to *multi-rate* plant and controllers, *i.e.*, a plant in which different sensor signals are sampled at different rates, or different actuator signals are updated at different rates. A parametrization of stabilizing multi-rate controllers has recently been developed by Meyer [MEY90]; this parametrization uses a transfer matrix $Q(z)$ that ranges over all stable transfer matrices that satisfy some additional convex constraints.

16.4.2 Nonlinear Plants

There are several heuristic methods for designing a nonlinear controller for a nonlinear plant, based on the design of an LTI controller for an LTI plant (or a family of LTI controllers for a family of LTI plants); see the Notes and References for chapter 2. In the Notes and References for chapter 10, we saw a method of designing a nonlinear controller for a plant that has saturating actuators. These methods often work well in practice, but do not qualify as extensions of the methods and ideas described in this book, since they do not consider all possible closed-loop systems that can be achieved. In a few cases, however, stronger results have been obtained.

In [DL82], Desoer and Liu have shown that for stable nonlinear plants, there is a parametrization of stabilizing controllers that is similar to the one described in section 7.2.4, provided a technical condition on P holds (incremental stability).

For unstable nonlinear plants, however, only partial results have been obtained. In [DL83] and [AD84], it is shown how a family of stabilizing controllers can be obtained by first finding one stabilizing controller, and then applying the results of Desoer and Liu mentioned above. But even in the case of an LTI plant and controller, this "two-step compensation" approach can fail to yield all controllers that stabilize the plant. This approach is discussed further in the articles [DL84A, DL84B, DL85].

In a series of papers, Hammer has investigated an extension of the stable factorization theory (see the Notes and References for chapter 7) to nonlinear systems; see [HAM88] and the references therein. Stable factorizations of nonlinear systems are also discussed in Verma [VER88].

Notation and Symbols

Basic Notation

Notation	Meaning
$\{\ \cdots\ \}$	Delimiters for sets, and for statement grouping in algorithms in chapter 14.
$(\ \cdots\ )$	Delimiters for expressions.
$f : X \to Y$	A function from the set X into the set Y.
$\emptyset$	The empty set.
$\wedge$	Conjunction of predicates; "and".
$\|\cdot\|$	A norm; see page 69. A particular norm is indicated with a mnemonic subscript.
$\partial\phi(x)$	The subdifferential of the functional ϕ at the point x; see page 293.
$\triangleq$	Equals by definition.
$\simeq$	Equals to first order.
$\approx$	Approximately equal to (used in vague discussions).
$\lesssim$	The inequality holds to first order.
δX	A first order change in X.
$\arg\min$	A minimizer of the argument. See page 58.
$\mathbf{C}$	The complex numbers.
$\mathbf{C}^n$	The vector space of n-component complex vectors.
$\mathbf{C}^{m\times n}$	The vector space of $m \times n$ complex matrices.
$\mathbf{E}\,X$	The expected value of the random variable X.
$\Im(z)$	The imaginary part of a complex number z.

inf	The infimum of a function or set. The reader unfamiliar with the notation inf can substitute min without ill effect.
j	A square root of -1.
lim sup	The asymptotic supremum of a function; see page 72.
$\mathbf{Prob}(Z)$	The probability of the event Z.
$\Re(z)$	The real part of a complex number z.
$\mathbf{R}$	The real numbers.
$\mathbf{R}_+$	The nonnegative real numbers.
$\mathbf{R}^n$	The vector space of n-component real vectors.
$\mathbf{R}^{m\times n}$	The vector space of $m \times n$ real matrices.
$\sigma_i(M)$	The ith singular value of a matrix M: the square root of the ith largest eigenvalue of M^*M.
$\sigma_{\max}(M)$	The maximum singular value of a matrix M: the square root of the largest eigenvalue of M^*M.
sup	The supremum of a function or set. The reader unfamiliar with the notation sup can substitute max without ill effect.
$\mathbf{Tr}\, M$	The trace of a matrix M: the sum of its entries on the diagonal.
$M \geq 0$	The $n \times n$ complex matrix M is positive semidefinite, *i.e.*, $z^*Mz \geq 0$ for all $z \in \mathbf{C}^n$.
$M > 0$	The $n \times n$ complex matrix M is positive definite, *i.e.*, $z^*Mz > 0$ for all nonzero $z \in \mathbf{C}^n$.
$\lambda \geq 0$	The n-component real-valued vector λ has nonnegative entries, *i.e.*, $\lambda \in \mathbf{R}^n_+$.
M^T	The transpose of a matrix or transfer matrix M.
M^*	The complex conjugate transpose of a matrix or transfer matrix M.
$M^{1/2}$	A symmetric square root of a matrix $M = M^* \geq 0$, *i.e.*, $M^{1/2}M^{1/2} = M$.

Global Symbols

Symbol	Meaning	Page
ϕ	A function from the space of transfer matrices to real numbers, *i.e.*, a functional on $\mathcal{H}$. A particular function is indicated with a mnemonic subscript.	53
φ	The restriction of a functional ϕ to a finite-dimensional domain, *i.e.*, a function from $\mathbf{R}^n$ to $\mathbf{R}$. A particular function is indicated with a mnemonic subscript.	252
$\mathcal{D}$	A design specification: a predicate or boolean function on $\mathcal{H}$. A particular design specification is indicated with a mnemonic subscript.	47
H	The closed-loop transfer matrix from w to z.	33
H_{ab}	The closed-loop transfer matrix from the signal b to the signal a.	32
$\mathcal{H}$	The set of all $n_z \times n_w$ transfer matrices. A particular subset of $\mathcal{H}$ (*i.e.*, a design specification) is indicated with a mnemonic subscript.	48
K	The transfer matrix of the controller.	32
L	The classical loop gain, $L = P_0K$.	36
n_w	The number of exogenous inputs, *i.e.*, the size of w.	26
n_u	The number of actuator inputs, *i.e.*, the size of u.	26
n_z	The number of regulated variables, *i.e.*, the size of z.	26
n_y	The number of sensed outputs, *i.e.*, the size of y.	26
P	The transfer matrix of the plant.	31
P_0	The transfer matrix of a classical plant, which is usually one part of the plant model P.	34
S	The classical sensitivity transfer function or matrix.	$36, 41$
T	The classical I/O transfer function or matrix.	$36, 41$
w	Exogenous input signal vector.	25
u	Actuator input signal vector.	25
z	Regulated output signal vector.	26
y	Sensed output signal vector.	26

Other Symbols

Symbol	Meaning	Page
Δ	A feedback perturbation.	221
$\boldsymbol{\Delta}$	A set of feedback perturbations.	221
$\lVert x\rVert$	The Euclidean norm of a vector $x \in \mathbf{R}^n$ or $x \in \mathbf{C}^n$, *i.e.*, $\sqrt{x^* x}$.	70
$\lVert u\rVert_1$	The $\mathbf{L}_1$ norm of the signal u.	81
$\lVert u\rVert_2$	The $\mathbf{L}_2$ norm of the signal u.	80
$\lVert u\rVert_\infty$	The peak magnitude norm of the signal u.	70
$\lVert u\rVert_{\mathrm{aa}}$	The average-absolute value norm of the signal u.	74
$\lVert u\rVert_{\mathrm{rms}}$	The RMS norm of the signal u.	72
$\lVert u\rVert_{\mathrm{ss}\infty}$	The steady-state peak magnitude norm of the signal u.	72
$\lVert H\rVert_2$	The $\mathbf{H}_2$ norm of the transfer function H.	$96, 110$
$\lVert H\rVert_\infty$	The $\mathbf{H}_\infty$ norm of the transfer function H.	$112, 112$
$\lVert H\rVert_{\infty,a}$	The a-shifted $\mathbf{H}_\infty$ norm of the transfer function H.	100
$\lVert H\rVert_{\mathrm{hankel}}$	The Hankel norm of the transfer function H.	103
$\lVert H\rVert_{\mathrm{pk\_step}}$	The peak of the step response of the transfer function H.	95
$\lVert H\rVert_{\mathrm{pk\_gn}}$	The peak gain of the transfer function H.	$97, 111$
$\lVert H\rVert_{\mathrm{rms},w}$	The RMS response of the transfer function H when driven by the stochastic signal w.	96
$\lVert H\rVert_{\mathrm{rms\_gn}}$	The RMS gain of the transfer function H, equal to its $\mathbf{H}_\infty$ norm.	$99, 112$
$\lVert H\rVert_{\mathrm{wc}}$	A worst case norm of the transfer function H.	97
$\mathcal{A}$	The region of achievable specifications in performance space.	139
$A_P, B_w, B_u,$ $C_z, C_y, D_{zw},$ D_{zu}, D_{yw}, D_{yu}	The matrices in a state-space representation of the plant.	43
$\mathbf{CF}(u)$	The crest factor of a signal u.	91
$\mathbf{Dz}(\cdot)$	The dead-zone function.	230
$I_\gamma(H)$	The γ-entropy of the transfer function H.	113
n_{proc}	A process noise, often actuator-referred.	35
n_{sensor}	A sensor noise.	35

$\mathbf{F}_u(a)$	The amplitude distribution function of the signal u.	76
$h(t)$	The impulse response of the transfer function H at time t.	96
$H^{(\mathrm{a})}$, $H^{(\mathrm{b})}$, $H^{(\mathrm{c})}$, $H^{(\mathrm{d})}$	The closed-loop transfer matrices from w to z achieved by the four controllers $K^{(\mathrm{a})}$, $K^{(\mathrm{b})}$, $K^{(\mathrm{c})}$, $K^{(\mathrm{d})}$ in our standard example system.	42
$K^{(\mathrm{a})}$, $K^{(\mathrm{b})}$, $K^{(\mathrm{c})}$, $K^{(\mathrm{d})}$	The four controllers in our standard example system.	42
$\mathcal{P}$	A perturbed plant set.	211
P_0^{std}	The transfer function of our standard example classical plant.	41
p, q	Auxiliary inputs and outputs used in the perturbation feedback form.	221
s	Used for *both* complex frequency, $s = \sigma + j\omega$, *and* the step response of a transfer function or matrix (although not usually in the same equation).	
$\mathbf{Sat}(\cdot)$	The saturation function.	220
$\mathrm{sgn}(\cdot)$	The sign function.	97
T_1, T_2, T_3	Stable transfer matrices used in the free parameter representation of achievable closed-loop transfer matrices.	156
$\star$	A submatrix or entry of H not relevant to the current discussion.	172
ϕ^*	The minimum value of the function ϕ.	311
x^*	A minimizing argument of the function ϕ, *i.e.*, $\phi^* = \phi(x^*)$.	311
$\mathbf{Tv}(f)$	The total variation of the function f.	98

List of Acronyms

Acronym	Meaning	Page
1-DOF	One Degree-Of-Freedom	37
2-DOF	Two Degree-Of-Freedom	39
ARE	Algebraic Riccati Equation	122
FIR	Finite Impulse Response	380
I/O	Input/Output	38
LQG	Linear Quadratic Gaussian	278
LQR	Linear Quadratic Regulator	275
LTI	Linear Time-Invariant	29
MAMS	Multiple-Actuator, Multiple-Sensor	40
MIMO	Multiple-Input, Multiple-Output	110
PID	Proportional plus Integral plus Derivative	5
QP	Quadratic Program	345
RMS	Root-Mean-Square	72
SASS	Single-Actuator, Single-Sensor	34
SISO	Single-Input, Single-Output	95

Bibliography

[Aar51] M. R. Aaron. Synthesis of feedback control systems by means of the closed loop function. *Trans. AIEE*, 70:1439–1445, 1951.

[ABJ86] B. Anderson, R. Bitmead, C. Johnson, P. Kokotovic, R. Kosut, I. Mareels, L. Praly, and B. Riedle. *Stability of Adaptive Systems: Passivity and Averaging Analysis.* MIT Press, 1986.

[ac100] ac-100 *manual.* Integrated Systems, Inc., 1988.

[Ack85] J. Ackermann. *Sampled-Data Control Systems.* Springer-Verlag, 1985.

[AD84] V. Anantharam and C. A. Desoer. On the stabilization of nonlinear systems. *IEEE Trans. Aut. Control*, AC-29(6):569–572, June 1984.

[AF66] M. Athans and P. Falb. *Optimal Control.* McGraw-Hill, 1966.

[AIE51] AIEE Committee Report. Proposed symbols and terms for feedback control systems. *Electrical Engineering*, 70:905–909, 1951.

[Akg84] M. Akgül. *Topics in Relaxation and Ellipsoidal Methods.* Volume 97 of *Research Notes in Mathematics*, Pitman, 1984.

[AL84] W. F. Arnold and A. J. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proc. IEEE*, 72(12):1746–1754, December 1984.

[All80] R. Allan. New applications open up for silicon sensors. *Electronics*, 113–122, November 1980.

[AM69] B. Anderson and J. B. Moore. Linear system optimization with prescribed degree of stability. *Proc. IEE*, 116(12):2083–2087, December 1969.

[AM90] B. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods.* Prentice-Hall, 1990.

[AN87] E. J. Anderson and P. Nash. *Linear Programming in Infinite-Dimensional Spaces: Theory and Applications.* John Wiley & Sons, 1987.

[And67] B. Anderson. An algebraic solution to the spectral factorization problem. *IEEE Trans. Aut. Control*, AC-12(4):410–414, 1967.

[AP84] E. J. Anderson and A. B. Philpott, editors. *Infinite Programming*, Springer-Verlag Lecture Notes in Economics and Mathematical Systems, September 1984.

[AS82] Y. I. Al'ber and S. V. Shil'man. A unified approach to the problem of minimizing smooth and nonsmooth functions. *Engineering Cybernetics*, 20(1):21–27, 1982.

[AS84] P. J. Antsaklis and M. K. Sain. Feedback controller parameterizations: finite hidden modes and causality. In S. G. Tzafestas, editor, *Multivariable Control*, pages 85–104, D. Reidel, 1984.

[AST83] K. J. Åström. Computer aided modelling, analysis and design of control systems—A perspective. *IEEE Control Syst. Mag.*, 4–16, May 1983.

[AT90] D. Auslander and C. Tham. *Real-time Software for Control: Program Examples in* C. Prentice-Hall, 1990.

[ATH71] M. Athans. Special issue on the Linear-Quadratic-Gaussian problem. *IEEE Trans. Aut. Control*, AC-16, December 1971.

[AUB79] J. P. Aubin. *Applied Functional Analysis.* John Wiley & Sons, 1979.

[AV82] J. P. Aubin and R. Vinter, editors. *Convex Analysis and Optimization.* Pitman, 1982.

[AW89] K. J. Åström and B. Wittenmark. *Adaptive Control.* Addison-Wesley, 1989.

[AW90] K. J. Åström and B. Wittenmark. *Computer Controlled Systems: Theory and Design.* Prentice-Hall, second edition, 1990.

[BB89] C. Barratt and S. Boyd. Example of exact tradeoffs in linear controller design. *IEEE Control Syst. Mag.*, 9(1):46–52, January 1989.

[BB90] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its $\mathbf{L}_\infty$-norm. *Syst. Control Letters*, 15(1), 1990.

[BBB88] S. Boyd, V. Balakrishnan, C. Barratt, N. Khraishi, X. Li, D. Meyer, and S. Norman. A new CAD method and associated architectures for linear controllers. *IEEE Trans. Aut. Control*, AC-33(3):268–283, March 1988.

[BBK89] S. Boyd, V. Balakrishnan, and P. Kabamba. A bisection method for computing the $\mathbf{H}_\infty$ norm of a transfer matrix and related problems. *Mathematics of Control, Signals, and Systems*, 2(3):207–219, 1989.

[BBN90] S. Boyd, C. Barratt, and S. Norman. Linear controller design: Limits of performance via convex optimization. *Proc. IEEE*, 78(3):529–574, March 1990.

[BD85] S. Boyd and C. A. Desoer. Subharmonic functions and performance bounds on linear time-invariant feedback systems. *IMA J. of Mathematical Control and Information*, 2:153–170, 1985.

[BD86] A. Bhaya and C. A. Desoer. Necessary and sufficient conditions on $Q(= C(I+PC)^{-1})$ for stabilization of the linear feedback system $S(P,C)$. *Syst. Control Letters*, 7(1):35–38, February 1986.

[BD87] S. Boyd and J. Doyle. Comparison of peak and RMS gains for discrete-time systems. *Syst. Control Letters*, 9:1–6, June 1987.

[BEN76] S. Bennett. The emergence of a discipline: Automatic control 1940-1960. *Automatica*, 12:113–121, 1976.

[BEN79] S. Bennett. *A History of Control Engineering 1800–1930.* Volume 8 of *IEE Control Engineering Series*, IEE, 1979.

[BER56] J. E. Bertram. Factors in the design of digital controllers for sampled-data feedback systems. *AIEE Transactions*, 75:151–59, July 1956.

[BF68] S. Blum and K. A. Fegley. A quadratic programming solution of the minimum energy control problem. *IEEE Trans. Aut. Control*, 206–207, April 1968.

[BGT81] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: a survey. *Operations Research*, 29(6):1039–1091, 1981.

[BH75] A. E. Bryson and Y. C. Ho. *Applied Optimal Control.* Hemisphere Publishing, 1975.

[BIG58] S. C. Bigelow. The design of analog computer compensated control systems. *AIEE Transactions*, 77:409–15, November 1958.

[BJ70] G. Box and G. Jenkins. *Time Series: Data Analysis and Theory.* Holden-Day, 1970.

[BLA34] H. S. Black. Stabilized feedback amplifiers. *Bell Syst. Tech. J.*, 13:1–18, 1934.

[BOD45] H. W. Bode. *Network Analysis and Feedback Amplifier Design.* Van Nostrand, 1945.

[BOY86] S. Boyd. A note on parametric and nonparametric uncertainties in control systems. In *Proc. American Control Conf.*, pages 1847–1849, June 1986.

[BP78] V. Barbu and T. Precupanu. *Convexity and Optimization in Banach Spaces.* Sijthoff and Noordhoff, 1978.

[BRO70] R. W. Brockett. *Finite Dimensional Linear Systems.* John Wiley & Sons, 1970.

[BS72] R. Bartels and G. Stewart. Solution of the equation $AX + XB = C$. *Comm. Assoc. Comp. Mach.*, 15:820–826, 1972.

[BS80] R. K. Brayton and R. Spence. *Sensitivity and Optimization.* Elsevier Scientific, 1980.

[CAI88] P. E. Caines. *Linear Stochastic Systems.* John Wiley & Sons, 1988.

[CAN67] R. Cannon. *Dynamics of Physical Systems.* McGraw-Hill, 1967.

[CD82A] F. M. Callier and C. A. Desoer. *Multivariable Feedback Systems.* Springer-Verlag, 1982.

[CD82B] M. J. Chen and C. A. Desoer. Necessary and sufficient condition for robust stability of linear distributed feedback systems. *Int. J. Control*, 35(2):255–267, 1982.

[CD83] M. J. Chen and C. A. Desoer. The problem of guaranteeing robust disturbance rejection in linear multivariable feedback systems. *Int. J. Control*, 37(2):305–313, 1983.

[CF68] A. J. Calise and K. A. Fegley. Quadratic programming in the statistical design of sampled-data control systems. *IEEE Trans. Aut. Control*, 77–80, February 1968.

[CG59] E. Cheney and A. Goldstein. Newton's method for convex programming and Tchebycheff approximation. *Numer. Math.*, 253–268, 1959.

[CH53] R. Courant and D. Hilbert. *Methods of Mathematical Physics.* Volume I, John Wiley & Sons, 1953.

[CHE82] C. S. Chen. Application of one-chip signal processor in digital controller implementation. *IEEE Control Syst. Mag.*, 16–22, September 1982.

[CHE84] C. T. Chen. *Linear System Theory and Design.* Holt, Rinehart and Winston, 1984.

[CHE87] C. T. Chen. *Linear Control System Design and Analysis.* Holt, Rinehart and Winston, 1987.

[CIF89] A. Cifuentes. *Using* MSC/NASTRAN. Springer-Verlag, 1989.

[CKK68] S. Crandall, D. Karnopp, E. Kurtz, and D. Pridmore-Brown. *Dynamics of Mechanical and Electromechanical Systems.* McGraw-Hill, 1968.

[CLA83] F. H. Clarke. *Optimization and Nonsmooth Analysis. Canadian Math. Soc. Series*, John Wiley & Sons, 1983.

[CP64] J. Cruz and W. Perkins. New approach to the sensitivity problem in multivariable feedback system design. *IEEE Trans. Aut. Control*, AC-9:216–223, 1964.

[CP67] N. O. Da Cunha and E. Polak. Constrained minimization under vector-valued criteria in finite dimensional spaces. *J. Math. Anal. and Appl.*, 19:103–124, 1967.

[CRR69] R. Calvin, C. Ray, and V. Rhyne. The design of optimal convolutional filters via linear programming. *IEEE Trans. Geoscience Elec.*, 7(3):142–145, July 1969.

[CRU73] J. Cruz, editor. *System Sensitivity Analysis.* Dowden, Hutchison, and Ross, 1973.

[DAN71] J. Daniel. *The Approximate Minimization of Functionals.* Prentice-Hall, 1971.

[DC75] C. A. Desoer and W. S. Chan. The feedback interconnection of lumped linear time-invariant systems. *J. Franklin Inst.*, 300(5 and 6):335–351, 1975.

[DC81A] C. A. Desoer and M. C. Chen. Design of multivariable feedback systems with stable plant. *IEEE Trans. Aut. Control*, AC-26(2):408–415, April 1981.

[DC81B] C. A. Desoer and M. C. Chen. Extension of the design algorithm of "design of multivariable feedback systems with stable plant" to the tracking problem. *IEEE Trans. Aut. Control*, AC-26(2):526–527, April 1981.

[DEB59] G. Debreu. *Theory of Value.* Volume 17 of *Cowles Foundation Monographs*, Yale University Press, 1959.

[DEL88] D. F. Delchamps. *State-Space and Input-Output Linear Systems.* Springer-Verlag, 1988.

[DEM89] J. Demmel. LAPACK: A portable linear algebra library for supercomputers. In *Proceedings of the Workshop on Computer-Aided Control System Design*, pages 1–7, IEEE Control Systems Society, December 1989.

[DEN84] M. J. Denham. Design issues for CACSD systems. *Proc. IEEE*, 72(12):1714–1723, December 1984.

[DES89] C. DeSilva. *Control Sensors and Actuators.* Prentice-Hall, 1989.

[DG84A] C. A. Desoer and C. L. Gustafson. Algebraic theory of linear multivariable feedback systems. *IEEE Trans. Aut. Control*, AC-29(10):909–917, October 1984.

[DG84B] C. A. Desoer and C. L. Gustafson. Design of multivariable feedback systems with simple unstable plant. *IEEE Trans. Aut. Control*, AC-29(10):901–908, October 1984.

[DGK89] J. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard $\mathbf{H}_2$ and $\mathbf{H}_\infty$ control problems. *IEEE Trans. Aut. Control*, 34(8):831–847, August 1989.

[DL82] C. A. Desoer and R. W. Liu. Global parametrization of feedback systems with nonlinear plants. *Syst. Control Letters*, 1:249–251, January 1982.

[DL83] C. A. Desoer and C. A. Lin. Two-step compensation of nonlinear systems. *Syst. Control Letters*, 3(1):41–45, June 1983.

[DL84A] C. A. Desoer and C. A. Lin. Non-linear unity-feedback systems and Q-parametrization. *Int. J. Control*, 40(1):37–51, 1984.

[DL84B] C. A. Desoer and C. A. Lin. Simultaneous stabilization of nonlinear systems. *IEEE Trans. Aut. Control*, AC-29(5):455–457, May 1984.

[DL85] C. A. Desoer and C. A. Lin. A comparative study of linear and nonlinear MIMO feedback configurations. *Int. J. Systems Sci.*, 16(7):789–813, 1985.

[DLM80] C. A. Desoer, R. W. Liu, J. Murray, and R. Saeks. Feedback system design: The fractional representation approach to analysis and synthesis. *IEEE Trans. Aut. Control*, AC-25(3):399–412, June 1980.

[DMB79] J. Dongarra, C. Moler, J. Bunch, and G. Stewart. LINPACK *users' guide*. SIAM, 1979.

[DOO81] P. Van Dooren. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Sci. Stat. Comp.*, 2:121–135, 1981.

[DOR87] P. Dorato, editor. *Robust Control*. IEEE Press, 1987.

[DOR88] R. C. Dorf. *Modern Control Systems*. Addison-Wesley, 5th edition, 1988.

[DOY78] J. Doyle. Guaranteed margins for LQG regulators. *IEEE Trans. Aut. Control*, AC-23(4):756–757, August 1978.

[DOY82] J. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proc.*, 129-D(6):242–250, November 1982.

[DOY84] J. Doyle. *Matrix Interpolation Theory and Optimal Control*. PhD thesis, University of California, Berkeley, 1984.

[DP87A] M. A. Dahleh and J. B. Pearson, Jr. $\mathbf{L}_1$-Optimal compensators for continuous-time systems. *IEEE Trans. Aut. Control*, AC-32(10):889–895, October 1987.

[DP87B] M. A. Dahleh and J. B. Pearson, Jr. $\mathbf{l}_1$-Optimal feedback controllers for MIMO discrete-time systems. *IEEE Trans. Aut. Control*, AC-32(4):314–322, April 1987.

[DP88A] M. A. Dahleh and J. B. Pearson, Jr. Minimization of a regulated response to a fixed input. *IEEE Trans. Aut. Control*, AC-33(10):924–930, October 1988.

[DP88B] M. A. Dahleh and J. B. Pearson, Jr. Optimal rejection of persistent disturbances, robust stability, and mixed sensitivity minimization. *IEEE Trans. Aut. Control*, AC-33(8):722–731, August 1988.

[DS81] J. Doyle and G. Stein. Multivariable feedback design: Concepts for a classical/modern synthesis. *IEEE Trans. Aut. Control*, AC-26(1):4–16, February 1981.

[DV75] C. A. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, 1975.

[DV85] V. F. Demyanov and L. V. Vasilev. *Nondifferentiable Optimization*. Optimization Software (Springer-Verlag), 1985.

[DWS82] J. Doyle, J. E. Wall, and G. Stein. Performance and robustness analysis for structured uncertainties. In *Proc. IEEE Conf. on Decision and Control*, pages 629–636, 1982.

[ECC80] *DC Motors, Speed Controls, Servo Systems, including Optical Encoders.* Hopkins, MN, fifth edition, 1980.

[EK83] J. Ecker and M. Kupferschmid. An ellipsoid algorithm for nonlinear programming. *Math. Program. Studies*, 27:83–106, 1983.

[EK85] J. Ecker and M. Kupferschmid. A computational comparison of the ellipsoid algorithm with several nonlinear programming algorithms. *SIAM J. Control and Opt.*, 23(5):657–674, September 1985.

[EM75] J. Elzinga and T. Moore. A central cutting plane algorithm for the convex programming problem. *Math. Program. Studies*, 8:134–145, 1975.

[Eva50] W. R. Evans. Control system synthesis by root locus method. *AIEE Transactions*, 69:66–69, 1950.

[Evt85] Y. Evtushenko. *Numerical Optimization Techniques. Translation Series in Mathematics and Engineering*, Optimization Software (Springer-Verlag), 1985.

[FBB71] K. A. Fegley, S. Blum, J. Bergholm, A. J. Calise, J. E. Marowitz, G. Porcelli, and L. P. Sinha. Stochastic and deterministic design and control via linear and quadratic programming. *IEEE Trans. Aut. Control*, AC-16(6):759–766, December 1971.

[FBS87] D. Flamm, S. Boyd, G. Stein, and S. Mitter. Tutorial workshop on $\mathbf{H}_\infty$-optimal control. December 1987. Preconference workshop, CDC.

[Feg64] K. A. Fegley. Designing sampled-data control systems by linear programming. *IEEE Trans. on Appl. and Ind.*, 83:198–200, May 1964.

[FH65] K. A. Fegley and M. I. Hsu. Optimum discrete control by linear programming. *IEEE Trans. Aut. Control*, 114–115, January 1965.

[FL85] J. S. Freudenberg and D. P. Looze. Right half plane poles and zeros and design trade-offs in feedback systems. *IEEE Trans. Aut. Control*, AC-30(6):555–565, June 1985.

[FL88] J. S. Freudenberg and D. P. Looze. *Frequency Domain Properties of Scalar and Multivariable Feedback Systems. Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1988.

[FPE86] G. F. Franklin, J. D. Powell, and A. Emami-Naeni. *Feedback Control of Dynamic Systems.* Addison-Wesley, 1986.

[FPW90] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital Control of Dynamic Systems.* Addison-Wesley, 1990.

[Fra87] B. A. Francis. *A course in* $\mathbf{H}_\infty$ *Control Theory.* Volume 88 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1987.

[Fra88] B. A. Francis. *Snippets of* $\mathbf{H}_\infty$ *Control Theory.* Technical Report 8802, Systems Control Group, Department of Electrical Engineering, University of Toronto, Toronto, Canada, July 1988.

[Fuk84] M. Fukushima. A descent algorithm for nonsmooth convex optimization. *Math. Program. Studies*, 30:163–175, 1984.

[FWK89] M. Fan, L. Wang, J. Koninckx, and A. Tits. Software package for optimization-based design with user-supplied simulators. *IEEE Control Syst. Mag.*, 9(1):66–71, 1989.

[GD83] C. L. Gustafson and C. A. Desoer. Controller design for linear multivariable feedback systems with stable plants, using optimization with inequality constraints. *Int. J. Control*, 37(5):881–907, 1983.

[GD85] C. L. Gustafson and C. A. Desoer. A CAD methodology for linear multivariable feedback systems based on algebraic theory. *Int. J. Control*, 41(3):653–675, 1985.

[GD88] K. Glover and J. Doyle. State-space formulae for all stabilizing controllers that satisfy an $\mathbf{H}_\infty$-norm bound and relations to risk sensitivity. *Syst. Control Letters*, 11:167–172, 1988.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-Completeness.* W. H. Freeman, 1979.

[GL81] P. Gács and L. Lovász. Khachiyan's algorithm for linear programming. *Math. Program. Studies*, 14:61–68, 1981.

[GL89] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, second edition, 1989.

[GLO84] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their $\mathbf{L}_\infty$-error bounds. *Int. J. Control*, 39(6):1115–1193, 1984.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Volume 2 of *Algorithms and Combinatorics*, Springer-Verlag, 1988.

[GM89] K. Glover and D. Mustafa. Derivation of the maximum entropy $\mathbf{H}_\infty$-controller and a state-space formula for its entropy. *Int. J. Control*, 50(3):899–916, 1989.

[GNL79] G. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the matrix problem $AX + XB = C$. *IEEE Trans. Aut. Control*, AC-24:909–913, 1979.

[GOF83] J. L. Goffin. Convergence rates of the ellipsoid method on general convex functions. *Math. Operations Research*, 8(1):135–150, 1983.

[GOF84] J. L. Goffin. Variable metric relaxation methods, part II: the ellipsoid method. *Math. Program. Studies*, 30:147–162, 1984.

[GP79] C. Gonzaga and E. Polak. On constraint dropping schemes and optimality functions for a class of outer approximation algorithms. *SIAM J. Control and Opt.*, 17(4):477–493, 1979.

[GPS84] Global positioning system. 1984. Two volume set of reprinted articles from *Navigation*, published by the Institute of Navigation.

[GS84] G. Goodwin and K. Sin. *Adaptive Filtering, Prediction, and Control.* Prentice-Hall, 1984.

[GV68] A. Gelb and W. E. Vander Velde. *Multiple-Input Describing Functions and Nonlinear System Design.* McGraw-Hill, 1968.

[HAM88] J. Hammer. Assignment of dynamics for non-linear recursive feedback systems. *Int. J. Control*, 48(3):1183–1212, 1988.

[HAN87] H. Hanselmann. Implementation of digital controllers—a survey. *Automatica*, 7–32, January 1987.

[HEP89] HP3563A *Control Systems Analyzer.* Hewlett-Packard Co., 1989.

[HET78] R. Hettich, editor. *Semi-Infinite Programming*, Springer-Verlag, 1978.

[Hor63] I. M. Horowitz. *Synthesis of Feedback Systems.* Academic Press, 1963.

[Hor87] M. F. Hordeski. *Transducers for Automation.* Van Nostrand Reinhold, 1987.

[HS78] C. A. Harvey and G. Stein. Quadratic weights for asymptotic regulation properties. *IEEE Trans. Aut. Control*, AC-23:378–387, 1978.

[HSM83] L. Hunt, R. Su, and G. Meyer. Global transformations of nonlinear systems. *IEEE Trans. Aut. Control*, AC-28:24–31, 1983.

[Ise84] R. Isermann. Process fault detection based on modeling and estimation methods—A survey. *Automatica*, 20(4):387–404, 1984.

[Isi89] A. Isidori. *Nonlinear Control Systems: An Introduction.* Springer-Verlag, second edition, 1989.

[JH85] M. Jamshidi and C. J. Herget, editors. *Computer-Aided Control Systems Engineering.* North-Holland, 1985.

[JNP47] H. James, N. Nichols, and R. Philips. *Theory of Servomechanisms.* McGraw-Hill, 1947.

[JTP85] V. J. Jaswa, C. E. Thomas, and J. T. Pedicone. CPAC—concurrent processor architecture for control. *IEEE Trans. on Computers*, 163–169, February 1985.

[KA82] L. V. Kantorovich and G. P. Akilov. *Functional Analysis.* Pergamon Press, 1982.

[Kai80] T. Kailath. *Linear Systems.* Prentice-Hall, 1980.

[Kel60] J. E. Kelley. The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math*, 8(4):703–712, December 1960.

[KF75] A. N. Kolmogorov and S. V. Fomin. *Introductory Real Analysis.* Dover, 1975.

[Kha79] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Doklady*, 20:191–194, 1979.

[Kiw85] K. C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization.* Volume 1133 of *Lecture Notes in Mathematics*, Springer-Verlag, 1985.

[KKO86] P. Kokotovic, H. Khalil, and J. O'Reilly. *Singular Perturbation Methods in Control: Analysis and Design.* Academic Press, 1986.

[KL85] T. Kane and D. Levinson. *Dynamics: Theory and Application. McGraw-Hill Series in Mechanical Engineering*, McGraw-Hill, 1985.

[KL86] N. T. Koussoulas and C. T. Leondes. The multiple Linear Quadratic Gaussian problem. *Int. J. Control*, 43(2):337–349, 1986.

[KME85] M. Kupferschmid, K. Mohrmann, J. Ecker, and H. Kaufman. The ellipsoid algorithm: a new method for feedback gain optimization. In *Control Applications of Nonlinear Programming and Optimization*, pages 85–94, IFAC, June 1985.

[KS72] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems.* John Wiley & Sons, 1972.

[Kuc79] V. Kučera. *Discrete Linear Control: The Polynomial Equation Approach.* John Wiley & Sons, 1979.

[Lau79] A. J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Aut. Control*, AC-24(6):913–921, December 1979.

[LAU85] A. J. Laub. Numerical linear algebra aspects of control design computations. *IEEE Trans. Aut. Control*, AC-30(2):97–108, February 1985.

[LEM75] C. Lemarechal. An extension of Davidon methods to nondifferentiable problems. In M. L. Balinski and P. Wolfe, editors, *Math. Programming Study 3, Nondifferentiable Optimization*, pages 95–109, North-Holland, 1975.

[LEV65] A. Levin. An algorithm for the minimization of convex functions. *Soviet Math. Doklady*, 6:286–290, 1965.

[LJU86] L. Ljung. *System Identification Toolbox for use with* PRO-MATLAB*: User's Guide.* The MathWorks, Inc., 1986.

[LJU87] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, 1987.

[LL87] A. Laub and J. Little. *Control System Toolbox for use with* PRO-MATLAB*: User's Guide.* The MathWorks, Inc., 1987.

[LM70] Y. I. Lyubich and G. D. Maistrovskii. The general theory of relaxation processes for convex functions. *Russian Mathematical Surveys*, 57–117, 1970.

[LP66] E. Levitin and B. Polyak. Constrained minimization methods. *USSR Computational Math. and Math. Physics*, 6(5):1–50, 1966.

[LUC89] D. T. Luc. *Theory of Vector Optimization.* Volume 319 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, 1989.

[LUE69] D. G. Luenberger. *Optimization By Vector Space Methods.* John Wiley & Sons, 1969.

[LUE84] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, 2nd edition, 1984.

[LUN89] J. Lunze. *Robust Multivariable Feedback Control.* Prentice-Hall, 1989.

[MAC79] A. G. J. MacFarlane, editor. *Frequency Response Methods in Control Systems. Selected Reprints*, IEEE Press, 1979.

[MAC86] J. Mackerle. Survey of general purpose finite element and boundary element computer programs for structural and solid mechanics applications. In A. Niku-Lari, editor, *Structural Analysis Systems*, pages 257–298, Pergamon Press, 1986.

[MAC89] J. M. Maciejowski. *Multivariable Feedback Design.* Addison-Wesley, 1989.

[MAT88] MATRIX-X SYSTEM IDENTIFICATION *manual.* Integrated Systems, Inc., 1988.

[MAX68] J. C. Maxwell. On governors. *Proc. Royal Soc.*, 16:270–283, 1868.

[MAY70] O. Mayr. *The Origins of Feedback Control.* MIT Press, 1970.

[MB75] A. Mees and A. Bergen. Describing functions revisited. *IEEE Trans. Aut. Control*, 20:473–478, August 1975.

[MEY90] D. Meyer. A parametrization of stabilizing controllers for multirate sampled-data systems. *IEEE Trans. Aut. Control*, 35(2):233–235, February 1990.

[MF71] J. E. Marowitz and K. A. Fegley. System design using quadratic programming. *IEEE Trans. Aut. Control*, AC-16(3):241–247, June 1971.

[MG90] D. Mustafa and K. Glover. *Minimum Entropy* $\mathbf{H}_\infty$ *Control. Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1990.

[MGT90] J. B. Moore, K. Glover, and A. Telford. All stabilizing controllers as frequency-shaped state estimate feedback. *IEEE Trans. Aut. Control*, 35(2):203–208, 1990.

[MLB87] C. Moler, J. Little, S. Bangert, and S. Kleiman. PRO-MATLAB *for Sun Workstations: User's Guide.* The MathWorks, Inc., 1987.

[MOO68] J. B. Moore. A circle criterion generation for relative stability. *IEEE Trans. Aut. Control*, 13:127–128, 1968.

[MT87] P. M. Mäkilä and H. T. Toivonen. Computational methods for parametric LQ minimization: A survey. *IEEE Trans. Aut. Control*, AC-32(8):658–671, August 1987.

[MUS89] D. Mustafa. *Minimum Entropy* $\mathbf{H}_\infty$ *Control.* PhD thesis, Cambridge University, May 1989.

[MZ89] M. Morari and E. Zafirou. *Robust Process Control.* Prentice-Hall, 1989.

[NET86] C. N. Nett. Algebraic aspects of linear control system stability. *IEEE Trans. Aut. Control*, AC-31:941–949, October 1986.

[NEW65] D. J. Newman. Location of the maximum on unimodal surfaces. *J. of the Assoc. for Computing Machinery*, 12(3):395–398, July 1965.

[NG64] K. S. Narendra and R. M. Goldwyn. A geometrical criterion for the stability of certain nonlinear nonautonomous systems. *IEEE Trans. Circuit Theory*, CT-11:406–408, September 1964.

[NG89] W. Y. Ng. *Interactive Multi-Objective Programming as a Framework for Computer-Aided Control System Design. Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1989.

[NGK57] G. Newton, L. Gould, and J. Kaiser. *Analytical Design of Linear Feedback Controls.* John Wiley & Sons, 1957.

[NIK86] A. Niku-Lari, editor. *Structural Analysis Systems.* Volume 1, Pergamon Press, 1986.

[NM53] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton Univ. Press, third edition, 1953.

[NOR86] J. P. Norton. *An Introduction to Identification.* Academic Press, 1986.

[NT86] W. Nye and A. Tits. An application-oriented, optimization-based methodology for interactive design of engineering systems. *Int. J. Control*, 43(6):1693–1721, 1986.

[NY83] A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley & Sons, 1983.

[OB90] C. Oakley and C. Barratt. End-point controller design for an experimental two-link flexible manipulator using convex optimization. *Proc. American Control Conf.*, 1990.

[OF85] S. D. O'Young and B. A. Francis. Sensitivity tradeoffs for multivariable plants. *IEEE Trans. Aut. Control*, AC-30(7):625–632, July 1985.

[OF86] S. D. O'Young and B. A. Francis. Optimal performance and robust stabilization. *Automatica*, 22(2):171–183, 1986.

[OGA87] K. Ogata. *Discrete-Time Control Systems.* Prentice-Hall, 1987.

[OGA90] K. Ogata. *Modern Control Engineering.* Prentice-Hall, second edition, 1990.

[OKU88] Y. Ohashi, H. Kando, H. Ukai, and T. Iwazumi. Optimal design of FIR linear phase filters via convex quadratic programming method. *Int. J. Systems Sci.*, 19(8):1469–1482, 1988.

[OS70] A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing.* Prentice-Hall, 1970.

[PAR96] V. Pareto. *Cours d'Économie Politique.* Rouge, 1896.

[PET82] K. E. Petersen. Silicon as a mechanical material. *Proc. IEEE*, 70(5):420–457, May 1982.

[PF66] G. Porcelli and K. A. Fegley. Optimal design of digitally compensated systems by quadratic programming. *J. Franklin Inst.*, 282:303–317, 1966.

[PIE84] Special issue on CACSD. IEEE Proceedings, December 1984.

[PM79] I. Postlethwaite and A. G. J. MacFarlane. *A Complex Variable Approach to the Analysis of Linear Multivariable Feedback Systems.* Springer-Verlag, 1979.

[PMS84] E. Polak, D. Mayne, and D. Stimler. Control system design via semi-infinite optimization: A review. *Proc. IEEE*, 72(12):1777–1794, December 1984.

[POL73] E. Polak. An historical survey of computational methods in optimal control. *SIAM Rev.*, 15(2):553–584, April 1973.

[POL83] E. Polak. Semi-infinite optimization in engineering design. In A. Fiacco and K. Kortanek, editors, *Semi-infinite Programming and Applications*, Springer-Verlag, 1983.

[POL87] E. Polak. On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Rev.*, 29:21–89, March 1987.

[PON61] L. Pontryagin. *Optimal Regulation Processes.* Volume 18, American Mathematical Society Translations, 1961.

[PON62] L. Pontryagin. *The Mathematical Theory of Optimal Processes.* John Wiley & Sons, 1962.

[PR87] P. M. Pardalos and J. B. Rosen. *Constrained Global Optimization: Algorithms and Applications.* Springer-Verlag, 1987.

[PRE89] *LM628 Precision Motion Controller.* National Semiconductor, 1989.

[PS89] E. Polak and S. Salcudean. On the design of linear multivariable feedback systems via constrained nondifferentiable optimization in $\mathbf{H}_\infty$ spaces. *IEEE Trans. Aut. Control*, AC-34(3):268–276, 1989.

[PSW85] E. Polak, P. Siegel, T. Wuu, W. Nye, and D. Mayne. DELIGHT.MIMO: An interactive, optimization-based multivariable control system design package. In M. Jamshidi and C. J. Herget, editors, *Computer-Aided Control Systems Engineering*, North-Holland, 1985. Reprinted from *IEEE Cont. Sys. Mag.*, No. 4 (Dec. 1982).

[PW82] E. Polak and Y. Wardi. A nondifferentiable optimization algorithm for the design of control systems subject to singular value inequalities over a frequency range. *Automatica*, 18(3):267–283, 1982.

[RAB72] L. Rabiner. Linear program design of finite impulse response (FIR) digital filters. *IEEE Trans. Audio and Electroacoustics*, 20(4):280–288, October 1972.

[REI80] T. Reiland. Optimality conditions and duality in continuous programming I: Convex programs and a theorem of the alternative. *J. Math. Anal. and Appl.*, 77:297–325, 1980.

[RF58] J. R. Ragazzini and G. F. Franklin. *Sampled-Data Control Systems.* McGraw-Hill, 1958.

[RG75] L. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing.* Prentice-Hall, 1975.

[ROB89] G. Robel. On computing the infinity norm. *IEEE Trans. Aut. Control*, AC-34(8):882–884, August 1989.

[ROC70] R. T. Rockafellar. *Convex Analysis.* Princeton Univ. Press, second edition, 1970.

[ROC74] R. T. Rockafellar. *Conjugate Duality and Optimization.* Volume 16 of *Regional Conference Series in Applied Mathematics*, SIAM, 1974.

[ROC81] R. T. Rockafellar. *The Theory of Subgradients and its Application to Problems of Optimization.* Heldermann Verlag, 1981.

[ROC82] R. T. Rockafellar. *Convex Analysis and Optimization.* Pitman, 1982.

[ROS70] H. H. Rosenbrock. *State-space and Multivariable Systems.* John Wiley & Sons, 1970.

[RS86] D. Rosenthal and M. Sherman. High performance multibody simulations via symbolic equation manipulation and Kane's method. *J. Astronautical Sciences*, 34(3):223–239, 1986.

[SA81] M. G. Safonov and M. Athans. A multiloop generalization of the circle criterion for stability margin analysis. *IEEE Trans. Aut. Control*, AC-26:415–422, 1981.

[SAF80] M. G. Safonov. *Stability and Robustness of Multivariable Feedback Systems.* MIT Press, 1980.

[SAF82] M. G. Safonov. Stability margins of diagonally perturbed multivariable feedback systems. *IEE Proc.*, 129-D:251–256, 1982.

[SAL86] S. Salcudean. *Algorithms for Optimal Design of Feedback Compensators.* PhD thesis, University of California, Berkeley, 1986.

[SAN64] I. W. Sandberg. A frequency-domain condition for the stability of feedback systems containing a single time-varying nonlinear element. *Bell Syst. Tech. J.*, 43(3):1601–1608, July 1964.

[SB87] C. Slivinsky and J. Borninski. Control system compensation and implementation with the TMS32010. In K. S. Lin, editor, *Digital Signal Processing Applications with the* TMS*320 Family*, pages 689–716, Prentice-Hall, 1987.

[SB88] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness.* Prentice-Hall, 1988.

[SBD76] B. Smith, J. Boyle, J. Dongarra, B. Garbow, Y. Ikebe, V. Klema, and C. Moler. *Matrix eigensystem routines:* EISPACK *guide.* Springer-Verlag, 1976.

[SD84] M. G. Safonov and J. Doyle. Minimizing conservativeness of robust singular values. In S. G. Tzafestas, editor, *Multivariable Control*, pages 197–207, D. Reidel, 1984.

[SFL85] S. C. Shah, M. A. Floyd, and L. L. Lehman. MATRIX-X: Control design and model building CAE capability. In M. Jamshidi and C. J. Herget, editors, *Computer-Aided Control Systems Engineering*, North-Holland, 1985.

[SHO85] N. Z. Shor. *Minimization Methods for Non-differentiable Functions. Springer Series in Computational Mathematics*, Springer-Verlag, 1985.

[SL88] D. Schaechter and D. Levinson. Interactive computerized symbolic dynamics for the dynamicist. *J. Astronautical Sciences*, 36(4):365–388, 1988.

[SNT85] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization.* Volume 176 of *Mathematics in Science and Engineering*, Academic Press, 1985.

[SR88] M. Sherman and D. Rosenthal. SDEXACT *User's Manual.* Symbolic Dynamics, Inc., Mountain View, CA, February 1988.

[SS89] J. Stokes and G. Sohie. Implementation of PID controllers on the Motorola DSP56000/DSP56001. 1989. Motorola, Inc.

[STE79] G. Stein. Generalized quadratic weights for asymptotic regulator properties. *IEEE Trans. Aut. Control*, AC-24:559–566, 1979.

[SW70] J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I.* Springer-Verlag, 1970.

[TAN80] A. Tannenbaum. Feedback stabilization of linear dynamical plants with uncertainty in the gain factor. *Int. J. Control*, 32(1):1–16, 1980.

[TAN82] A. Tannenbaum. Modified Nevanlinna-Pick interpolation and feedback stabilization of linear plants with uncertainty in the gain factor. *Int. J. Control*, 36(2):331–336, 1982.

[TL80] D. Tabak and G. Lipovski. Move architecture in digital controllers. *IEEE Trans. Comp.*, 29(2):180–189, February 1980.

[TM89] H. T. Toivonen and P. M. Mäkilä. Computer-aided design procedure for multiobjective LQG control problems. *Int. J. Control*, 49(2):655–666, 1989.

[TOI84] H. T. Toivonen. A multiobjective linear quadratic Gaussian control problem. *IEEE Trans. Aut. Control*, AC-29(3):279–280, March 1984.

[TRA89A] IEEE. *Transducers 1989: 5th International Conference on Solid-State Sensors and Actuators*, 1989.

[TRA89B] *Transducers: Pressure and Temperature.* National Semiconductor, 1989.

[TRU50] J. G. Truxal. *Servomechanism Synthesis through Pole-Zero Configurations.* Technical Report 162, M. I. T. Research Laboratory of Electronics, Cambridge, Massachusetts, August 1950.

[TRU55] J. G. Truxal. *Automatic Feedback Control System Synthesis.* McGraw-Hill, 1955.

[VER88] M. S. Verma. Coprime fractional representations of stability of non-linear feedback systems. *Int. J. Control*, 48(3):897–918, 1988.

[VID78] M. Vidyasagar. *Nonlinear Systems Analysis.* Prentice-Hall, 1978.

[VID85] M. Vidyasagar. *Control System Synthesis: A Factorization Approach.* MIT Press, 1985.

[VID86] M. Vidyasagar. Optimal rejection of persistent bounded disturbances. *IEEE Trans. Aut. Control*, AC-31(6):527–535, June 1986.

[WAR88] I. Warnock. *Programmable Controllers: Operations and Application.* Prentice-Hall International, 1988.

[WEL87] D. Wells. *Guide to GPS Positioning.* Canadian GPS Associates, 1987.

[WH85] E. Wong and B. Hajek. *Stochastic Processes in Engineering Systems.* Springer-Verlag, 1985.

[WHI90] P. Whittle. *Risk-Sensitive Optimal Control.* John Wiley & Sons, 1990.

[WIE82] A. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3:391–405, 1982.

[WIL69] J. C. Willems. *The Analysis of Feedback Systems.* Volume 62 of *Research Monographs*, MIT Press, 1969.

[WIL73] J. L. Willems. The circle criterion and quadratic Lyapunov functions for stability analysis. *IEEE Trans. Aut. Control*, 18:184–186, 1973.

[WOL74] W. A. Wolovich. *Linear Multivariable Systems.* Springer-Verlag, 1974.

[WOL75] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. In M. L. Balinski and P. Wolfe, editors, *Math. Programming Study 3, Nondifferentiable Optimization*, pages 145–173, North-Holland, 1975.

[WON85] W. M. Wonham. *Linear Multivariable Control.* Springer-Verlag, 1985.

[WSG84] R. A. Walker, S. C. Shah, and N. K. Gupta. Computer-aided engineering (CAE) for system analysis. *Proc. IEEE*, 72(12):1732–1745, December 1984.

[YBJ76] D. C. Youla, J. J. Bongiorno, and H. A. Jabr. Modern Wiener-Hopf design of optimal controllers—Part I: The single-input-output case. *IEEE Trans. Aut. Control*, AC-21:3–13, February 1976.

[YJB76] D. C. Youla, H. A. Jabr, and J. J. Bongiorno. Modern Wiener-Hopf design of optimal controllers—Part II: The multivariable case. *IEEE Trans. Aut. Control*, AC-21:319–338, June 1976.

[YN77] D. Yudin and A. Nemirovsky. The effectiveness of random search in control problems. *Engineering Cybernetics*, 15(3):1–14, 1977.

[YOU61] D. C. Youla. On the factorization of rational matrices. *IRE Trans. IT*, IT-7(3):172–189, July 1961.

[ZA73] V. Zakian and U. Al-Naib. Design of dynamical and control systems by the method of inequalities. *Proc. IEE*, 120:1421–1427, 1973.

[ZAD63] L. A. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Trans. Aut. Control*, 59–60, January 1963.

[ZAM66A] G. Zames. On the input-output stability on nonlinear time-varying feedback systems—Part I: conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Trans. Aut. Control*, AC-11:228–238, April 1966.

[ZAM66B] G. Zames. On the input-output stability on nonlinear time-varying feedback systems—Part II: conditions involving circles in the frequency plane and sector nonlinearities. *IEEE Trans. Aut. Control*, AC-11:465–476, July 1966.

[ZAM81] G. Zames. Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Trans. Aut. Control*, AC-26(2):301–320, April 1981.

[ZD63] L. A. Zadeh and C. A. Desoer. *Linear System Theory: The State Space Approach.* McGraw-Hill, 1963. Reprinted by Krieger, 1979.

[ZF83] G. Zames and B. A. Francis. Feedback, minimax sensitivity and optimal robustness. *IEEE Trans. Aut. Control*, AC-28(5):585–601, May 1983.

[ZN42] J. Ziegler and N. Nichols. Optimum settings for automatic controllers. *ASME Trans.*, 64:759–768, 1942.

Index