Chapter 1

Introduction

Without control systems there could be no manufacturing, no vehicles, no computers, no regulated environment—in short, no technology. Control systems are what make machines, in the broadest sense of the term, function as intended. Control systems are most often based on the principle of feedback, whereby the signal to be controlled is compared to a desired reference signal and the discrepancy used to compute corrective control action. The goal of this book is to present a theory of feedback control system design that captures the essential issues, can be applied to a wide range of practical problems, and is as simple as possible.

1.1 Issues in Control System Design

The process of designing a control system generally involves many steps. A typical scenario is as follows:

1. Study the system to be controlled and decide what types of sensors and actuators will be used and where they will be placed.

2. Model the resulting system to be controlled.

3. Simplify the model if necessary so that it is tractable.

4. Analyze the resulting model; determine its properties.

5. Decide on performance specifications.

6. Decide on the type of controller to be used.

7. Design a controller to meet the specs, if possible; if not, modify the specs or generalize the type of controller sought.

8. Simulate the resulting controlled system, either on a computer or in a pilot plant.

9. Repeat from step 1 if necessary.

10. Choose hardware and software and implement the controller.

11. Tune the controller on-line if necessary.

It must be kept in mind that a control engineer's role is not merely one of designing control systems for fixed plants, of simply "wrapping a little feedback" around an already fixed physical system. It also involves assisting in the choice and configuration of hardware by taking a system-wide view of performance. For this reason it is important that a theory of feedback not only lead to good designs when these are possible, but also indicate directly and unambiguously when the performance objectives cannot be met.

It is also important to realize at the outset that practical problems have uncertain, non-minimum-phase plants (*non-minimum-phase* means the existence of right half-plane zeros, so the inverse is unstable); that there are inevitably unmodeled dynamics that produce substantial uncertainty, usually at high frequency; and that sensor noise and input signal level constraints limit the achievable benefits of feedback. A theory that excludes some of these practical issues can still be useful in limited application domains. For example, many process control problems are so dominated by plant uncertainty and right half-plane zeros that sensor noise and input signal level constraints can be neglected. Some spacecraft problems, on the other hand, are so dominated by tradeoffs between sensor noise, disturbance rejection, and input signal level (e.g., fuel consumption) that plant uncertainty and non-minimum-phase effects are negligible. Nevertheless, any general theory should be able to treat all these issues explicitly and give quantitative and qualitative results about their impact on system performance.

In the present section we look at two issues involved in the design process: deciding on performance specifications and modeling. We begin with an example to illustrate these two issues.

Example A very interesting engineering system is the Keck astronomical telescope, currently under construction on Mauna Kea in Hawaii. When completed it will be the world's largest. The basic objective of the telescope is to collect and focus starlight using a large concave mirror. The shape of the mirror determines the quality of the observed image. The larger the mirror, the more light that can be collected, and hence the dimmer the star that can be observed. The diameter of the mirror on the Keck telescope will be 10 m. To make such a large, high-precision mirror out of a single piece of glass would be very difficult and costly. Instead, the mirror on the Keck telescope will be a mosaic of 36 hexagonal small mirrors. These 36 segments must then be aligned so that the composite mirror has the desired shape.

The control system to do this is illustrated in Figure 1.1. As shown, the mirror segments are subject to two types of forces: disturbance forces (described below) and forces from actuators. Behind each segment are three piston-type actuators, applying forces at three points on the segment to effect its orientation. In controlling the mirror's shape, it suffices to control the misalignment between adjacent mirror segments. In the gap between every two adjacent segments are (capacitor-type) sensors measuring local displacements between the two segments. These local displacements are stacked into the vector labeled y; this is what is to be controlled. For the mirror to have the ideal shape, these displacements should have certain ideal values that can be pre-computed; these are the components of the vector r. The controller must be designed so that in the closed-loop system y is held close to r despite the disturbance forces. Notice that the signals are vector valued. Such a system is *multivariable*.

Our uncertainty about the plant arises from disturbance sources:

- As the telescope turns to track a star, the direction of the force of gravity on the mirror changes.
- During the night, when astronomical observations are made, the ambient temperature changes.
- The telescope is susceptible to wind gusts.

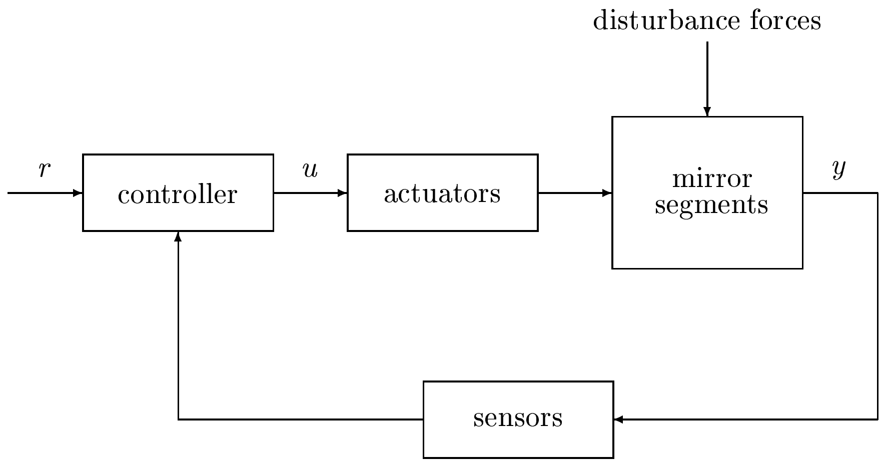

Figure 1.1: Block diagram of Keck telescope control system.

and from uncertain plant dynamics:

- The dynamic behavior of the components—mirror segments, actuators, sensors—cannot be modeled with infinite precision.

Now we continue with a discussion of the issues in general.

Control Objectives

Generally speaking, the objective in a control system is to make some output, say y, behave in a desired way by manipulating some input, say u. The simplest objective might be to keep y small (or close to some equilibrium point)—a *regulator problem*—or to keep $y - r$ small for r, a reference or command signal, in some set—a *servomechanism* or *servo problem*. Examples:

- On a commercial airplane the vertical acceleration should be less than a certain value for passenger comfort.

- In an audio amplifier the power of noise signals at the output must be sufficiently small for high fidelity.

- In papermaking the moisture content must be kept between prescribed values.

There might be the side constraint of keeping u itself small as well, because it might be constrained (e.g., the flow rate from a valve has a maximum value, determined when the valve is fully open) or it might be too expensive to use a large input. But what is small for a signal? It is natural to introduce norms for signals; then "y small" means "$\|y\|$ small." Which norm is appropriate depends on the particular application.

In summary, performance objectives of a control system naturally lead to the introduction of norms; then the specs are given as norm bounds on certain key signals of interest.

Models

Before discussing the issue of modeling a physical system it is important to distinguish among four different objects:

1. *Real physical system*: the one "out there."
2. *Ideal physical model*: obtained by schematically decomposing the real physical system into ideal building blocks; composed of resistors, masses, beams, kilns, isotropic media, Newtonian fluids, electrons, and so on.
3. *Ideal mathematical model*: obtained by applying natural laws to the ideal physical model; composed of nonlinear partial differential equations, and so on.
4. *Reduced mathematical model*: obtained from the ideal mathematical model by linearization, lumping, and so on; usually a rational transfer function.

Sometimes language makes a fuzzy distinction between the real physical system and the ideal physical model. For example, the word *resistor* applies to both the actual piece of ceramic and metal and the ideal object satisfying Ohm's law. Of course, the adjectives *real* and *ideal* could be used to disambiguate.

No mathematical system can precisely model a real physical system; there is always uncertainty. Uncertainty means that we cannot predict exactly what the output of a real physical system will be even if we know the input, so *we* are uncertain about the system. Uncertainty arises from two sources: unknown or unpredictable inputs (disturbance, noise, etc.) and unpredictable dynamics.

What should a model provide? It should predict the input-output response in such a way that we can use it to design a control system, and then be confident that the resulting design will work on the real physical system. Of course, this is not possible. A "leap of faith" will always be required on the part of the engineer. This cannot be eliminated, but it can be made more manageable with the use of effective modeling, analysis, and design techniques.

Mathematical Models in This Book

The models in this book are finite-dimensional, linear, and time-invariant. The main reason for this is that they are the simplest models for treating the fundamental issues in control system design. The resulting design techniques work remarkably well for a large class of engineering problems, partly because most systems are built to be as close to linear time-invariant as possible so that they are more easily controlled. Also, a good controller will keep the system in its linear regime. The uncertainty description is as simple as possible as well.

The basic form of the plant model in this book is

$$y = (P + \Delta)u + n.$$

Here y is the output, u the input, and P the nominal plant transfer function. The model uncertainty comes in two forms:

n: unknown noise or disturbance
Δ: unknown plant perturbation

Both n and Δ will be assumed to belong to sets, that is, some *a priori* information is assumed about n and Δ. Then every input u is capable of producing a *set* of outputs, namely, the set of all outputs $(P + \Delta)u + n$ as n and Δ range over their sets. Models capable of producing sets of outputs for a single input are said to be *nondeterministic*. There are two main ways of obtaining models, as described next.

Models from Science

The usual way of getting a model is by applying the laws of physics, chemistry, and so on. Consider the Keck telescope example. One can write down differential equations based on physical principles (e.g., Newton's laws) and making idealizing assumptions (e.g., the mirror segments are rigid). The coefficients in the differential equations will depend on physical constants, such as masses and physical dimensions. These can be measured. This method of applying physical laws and taking measurements is most successful in electromechanical systems, such as aerospace vehicles and robots. Some systems are difficult to model in this way, either because they are too complex or because their governing laws are unknown.

Models from Experimental Data

The second way of getting a model is by doing experiments on the physical system. Let's start with a simple thought experiment, one that captures many essential aspects of the relationships between physical systems and their models and the issues in obtaining models from experimental data. Consider a real physical system—the plant to be controlled—with one input, u, and one output, y. To design a control system for this plant, we must understand how u affects y.

The experiment runs like this. Suppose that the real physical system is in a rest state before an input u is applied (i.e., $u = y = 0$). Now apply some input signal u, resulting in some output signal y. Observe the pair (u, y). Repeat this experiment several times. Pretend that these data pairs are all we know about the real physical system. (This is the *black box* scenario. Usually, we know something about the internal workings of the system.)

After doing this experiment we will notice several things. First, the same input signal at different times produces different output signals. Second, if we hold $u = 0$, y will fluctuate in an unpredictable manner. Thus the real physical system produces just one output for any given input, so it itself is deterministic. However, we observers are uncertain because we cannot predict what that output will be.

Ideally, the model should *cover* the data in the sense that it should be capable of producing every experimentally observed input-output pair. (Of course, it would be better to cover not just the data observed in a finite number of experiments, but anything that can be produced by the real physical system. Obviously, this is impossible.) If nondeterminism that reasonably covers the range of expected data is not built into the model, we will not trust that designs based on such models will work on the real system.

In summary, for a useful theory of control design, plant models must be nondeterministic, having uncertainty built in explicitly.

Synthesis Problem

A synthesis problem is a theoretical problem, precise and unambiguous. Its purpose is primarily pedagogical: It gives us something clear to focus on for the purpose of study. The hope is that the principles learned from studying a formal synthesis problem will be useful when it comes to designing a real control system.

The most general block diagram of a control system is shown in Figure 1.2. The generalized plant consists of everything that is fixed at the start of the control design exercise: the plant, actuators that generate inputs to the plant, sensors measuring certain signals, analog-to-digital and digital-to-analog converters, and so on. The controller consists of the designable part: it may be an electric circuit, a programmable logic controller, a general-purpose computer, or some other

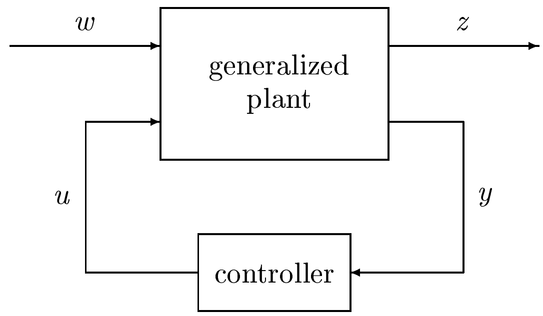

Figure 1.2: Most general control system.

such device. The signals w, z, y, and u are, in general, vector-valued functions of time. The components of w are all the exogenous inputs: references, disturbances, sensor noises, and so on. The components of z are all the signals we wish to control: tracking errors between reference signals and plant outputs, actuator signals whose values must be kept between certain limits, and so on. The vector y contains the outputs of all sensors. Finally, u contains all controlled inputs to the generalized plant. (Even open-loop control fits in; the generalized plant would be so defined that y is always constant.)

Very rarely is the exogenous input w a fixed, known signal. One of these rare instances is where a robot manipulator is required to trace out a definite path, as in welding. Usually, w is not fixed but belongs to a set that can be characterized to some degree. Some examples:

- In a thermostat-controlled temperature regulator for a house, the reference signal is always piecewise constant: at certain times during the day the thermostat is set to a new value. The temperature of the outside air is not piecewise constant but varies slowly within bounds.

- In a vehicle such as an airplane or ship the pilot's commands on the steering wheel, throttle, pedals, and so on come from a predictable set, and the gusts and wave motions have amplitudes and frequencies that can be bounded with some degree of confidence.

- The load power drawn on an electric power system has predictable characteristics.

Sometimes the designer does not attempt to model the exogenous inputs. Instead, she or he designs for a suitable response to a test input, such as a step, a sinusoid, or white noise. The designer may know from past experience how this correlates with actual performance in the field. Desired properties of z generally relate to how large it is according to various measures, as discussed above.

Finally, the output of the design exercise is a mathematical model of a controller. This must be implementable in hardware. If the controller you design is governed by a nonlinear partial differential equation, how are you going to implement it? A linear ordinary differential equation with constant coefficients, representing a finite-dimensional, time-invariant, linear system, can be simulated via an analog circuit or approximated by a digital computer, so this is the most common type of control law.

The synthesis problem can now be stated as follows: Given a set of generalized plants, a set of exogenous inputs, and an upper bound on the size of z, design an implementable controller to achieve this bound. How the size of z is to be measured (e.g., power or maximum amplitude) depends on the context. This book focuses on an elementary version of this problem.

1.2 What Is in This Book

Since this book is for a first course on this subject, attention is restricted to systems whose models are single-input/single-output, finite-dimensional, linear, and time-invariant. Thus they have transfer functions that are rational in the Laplace variable s. The general layout of the book is that Chapters 2 to 4 and 6 are devoted to analysis of control systems, that is, the controller is already specified, and Chapters 5 and 7 to 12 to design.

Performance of a control system is specified in terms of the size of certain signals of interest. For example, the performance of a tracking system could be measured by the size of the error signal. Chapter 2, *Norms for Signals and Systems*, looks at several ways of defining norms for a signal $u(t)$; in particular, the 2-norm (associated with energy),

$$\left(\int_{-\infty}^{\infty} u(t)^2 dt\right)^{1/2},$$

the ∞-norm (maximum absolute value),

$$\max_t |u(t)|,$$

and the square root of the average power (actually, not quite a norm),

$$\left(\lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} u(t)^2 dt\right)^{1/2}.$$

Also introduced are two norms for a system's transfer function $G(s)$: the 2-norm,

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

and the ∞-norm,

$$\|G\|_\infty := \max_\omega |G(j\omega)|.$$

Notice that $\|G\|_\infty$ equals the peak amplitude on the Bode magnitude plot of G. Then two very useful tables are presented summarizing input-output norm relationships. For example, one table gives a bound on the 2-norm of the output knowing the 2-norm of the input and the ∞-norm of the transfer function. Such results are very useful in predicting, for example, the effect a disturbance will have on the output of a feedback system.

Chapters 3 and 4 are the most fundamental in the book. The system under consideration is shown in Figure 1.3, where P and C are the plant and controller transfer functions. The signals are as follows:

r	reference or command input
e	tracking error
u	control signal, controller output
d	plant disturbance
y	plant output
n	sensor noise

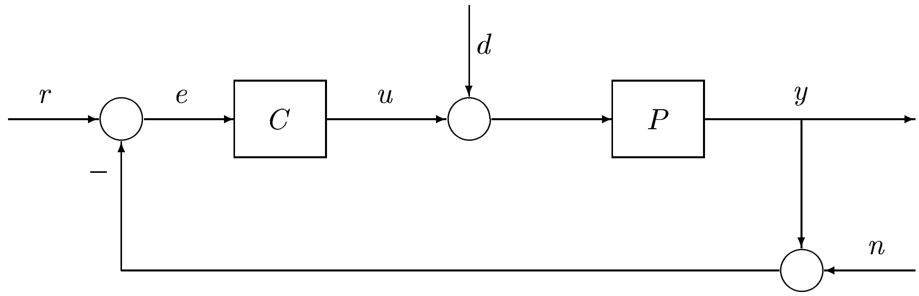

Figure 1.3: Single-loop feedback system.

In Chapter 3, *Basic Concepts*, internal stability is defined and characterized. Then the system is analyzed for its ability to track a *single* reference signal r—a step or a ramp—asymptotically as time increases. Finally, we look at tracking a *set* of reference signals. The transfer function from reference input r to tracking error e is denoted S, the *sensitivity function.* It is argued that a useful tracking performance criterion is $\|W_1S\|_\infty < 1$, where W_1 is a transfer function which can be tuned by the control system designer.

Since no mathematical system can exactly model a physical system, we must be aware of how modeling errors might adversely affect the performance of a control system. Chapter 4, *Uncertainty and Robustness*, begins with a treatment of various models of plant uncertainty. The basic technique is to model the plant as belonging to a set $\mathcal{P}$. Such a set can be either *structured*—for example, there are a finite number of uncertain parameters—or *unstructured*—the frequency response lies in a set in the complex plane for every frequency. For us, unstructured is more important because it leads to a simple and useful design theory. In particular, multiplicative perturbation is chosen for detailed study, it being typical. In this uncertainty model there is a nominal plant P and the family $\mathcal{P}$ consists of all perturbed plants $\tilde{P}$ such that at each frequency ω the ratio $\tilde{P}(j\omega)/P(j\omega)$ lies in a disk in the complex plane with center 1. This notion of disk-like uncertainty is key; because of it the mathematical problems are tractable.

Generally speaking, the notion of robustness means that some characteristic of the feedback system holds for every plant in the set $\mathcal{P}$. A controller C provides *robust stability* if it provides internal stability for every plant in $\mathcal{P}$. Chapter 4 develops a test for robust stability for the multiplicative perturbation model, a test involving C and $\mathcal{P}$. The test is $\|W_2T\|_\infty < 1$. Here T is the *complementary sensitivity function*, equal to $1 - S$ (or the transfer function from r to y), and W_2 is a transfer function whose magnitude at frequency ω equals the radius of the uncertainty disk at that frequency.

The final topic in Chapter 4 is robust performance, guaranteed tracking in the face of plant uncertainty. The main result is that the tracking performance spec $\|W_1S\|_\infty < 1$ is satisfied for all plants in the multiplicative perturbation set if and only if the magnitude of $|W_1S| + |W_2T|$ is less than 1 for all frequencies, that is,

$$\||W_1S| + |W_2T|\|_\infty < 1. \tag{1.1}$$

This is an analysis result: It tells exactly when some candidate controller provides robust performance.

Chapter 5, *Stabilization*, is the first on design. Most synthesis problems can be formulated like this: Given P, design C so that the feedback system (1) is internally stable, and (2) acquires some additional desired property or properties, for example, the output y asymptotically tracks a step input r. The method of solution presented here is to parametrize all Cs for which (1) is true and then to find a parameter for which (2) holds. In this chapter such a parametrization is derived; it has the form

$$C = \frac{X + MQ}{Y - NQ},$$

where N, M, X, and Y are fixed stable proper transfer functions and Q is the parameter, an arbitrary stable proper transfer function. The usefulness of this parametrization derives from the fact that all closed-loop transfer functions are very simple functions of Q; for instance, the sensitivity function S, while a nonlinear function of C, equals simply $MY - MNQ$. This parametrization is then applied to three problems: achieving asymptotic performance specs, such as tracking a step; internal stabilization by a stable controller; and simultaneous stabilization of two plants by a common controller.

Before we see how to design control systems for the robust performance specification, it is important to understand the basic limitations on achievable performance: Why can't we achieve both arbitrarily good performance and stability robustness at the same time? In Chapter 6, *Design Constraints*, we study design constraints arising from two sources: from algebraic relationships that must hold among various transfer functions and from the fact that closed-loop transfer functions must be stable, that is, analytic in the right half-plane. The main conclusion is that feedback control design always involves a tradeoff between performance and stability robustness.

Chapter 7, *Loopshaping*, presents a graphical technique for designing a controller to achieve robust performance. This method is the most common in engineering practice. It is especially suitable for today's CAD packages in view of their graphics capabilities. The loop transfer function is $L := PC$. The idea is to shape the Bode magnitude plot of L so that (1.1) is achieved, at least approximately, and then to back-solve for C via $C = L/P$. When P or P^{-1} is not stable, L must contain Ps unstable poles and zeros (for internal stability of the feedback loop), an awkward constraint. For this reason, it is assumed in Chapter 7 that P and P^{-1} are both stable.

Thus Chapters 2 to 7 constitute a basic treatment of feedback design, containing a detailed formulation of the control design problem, the fundamental issue of performance/stability robustness tradeoff, and a graphical design technique suitable for benign plants (stable, minimum-phase). Chapters 8 to 12 are more advanced.

Chapter 8, *Advanced Loopshaping*, is a bridge between the two halves of the book; it extends the loopshaping technique and connects it with the notion of optimal designs. Loopshaping in Chapter 7 focuses on L, but other quantities, such as C, S, T, or the Q parameter in the stabilization results of Chapter 5, may also be "shaped" to achieve the same end. For many problems these alternatives are more convenient. Chapter 8 also offers some suggestions on how to extend loopshaping to handle right half-plane poles and zeros.

Optimal controllers are introduced in a formal way in Chapter 8. Several different notions of optimality are considered with an aim toward understanding in what way loopshaping controllers can be said to be optimal. It is shown that loopshaping controllers satisfy a very strong type of optimality, called *self-optimality*. The implication of this result is that when loopshaping is successful at finding an adequate controller, it cannot be improved upon uniformly.

Chapters 9 to 12 present a recently developed approach to the robust performance design problem. The approach is mathematical rather than graphical, using elementary tools involving interpolation by analytic functions. This mathematical approach is most useful for multivariable systems, where graphical techniques usually break down. Nevertheless, the setting of single-input/single-output systems is where this new approach should be learned. Besides, present-day software for control design (e.g., MATLAB and Program CC) incorporate this approach.

Chapter 9, *Model Matching*, studies a hypothetical control problem called the model-matching problem: Given stable proper transfer functions T_1 and T_2, find a stable transfer function Q to minimize $\|T_1 - T_2 Q\|_\infty$. The interpretation is this: T_1 is a model, T_2 is a plant, and Q is a cascade controller to be designed so that $T_2 Q$ approximates T_1. Thus $T_1 - T_2 Q$ is the error transfer function. This problem is turned into a special interpolation problem: Given points $\{a_i\}$ in the right half-plane and values $\{b_i\}$, also complex numbers, find a stable transfer function G so that $\|G\|_\infty < 1$ and $G(a_i) = b_i$, that is, G interpolates the value b_i at the point a_i. When such a G exists and how to find one utilizes some beautiful mathematics due to Nevanlinna and Pick.

Chapter 10, *Design for Performance*, treats the problem of designing a controller to achieve the performance criterion $\|W_1 S\|_\infty < 1$ alone, that is, with no plant uncertainty. When does such a controller exist, and how can it be computed? These questions are easy when the inverse of the plant transfer function is stable. When the inverse is unstable (i.e., P is non-minimum-phase), the questions are more interesting. The solutions presented in this chapter use model-matching theory. The procedure is applied to designing a controller for a flexible beam. The desired performance is given in terms of step response specs: overshoot and settling time. It is shown how to choose the weight W_1 to accommodate these time domain specs. Also treated in Chapter 10 is minimization of the 2-norm of some closed-loop transfer function, e.g., $\|W_1 S\|_2$.

Next, in Chapter 11, *Stability Margin Optimization*, is considered the problem of designing a controller whose sole purpose is to maximize the stability margin, that is, performance is ignored. The maximum obtainable stability margin is a measure of how difficult the plant is to control. Three measures of stability margin are treated: the ∞-norm of a multiplicative perturbation, gain margin, and phase margin. It is shown that the problem of optimizing these stability margins can also be reduced to a model-matching problem.

Chapter 12, *Design for Robust Performance*, returns to the robust performance problem of designing a controller to achieve (1.1). Chapter 7 proposed loopshaping as a graphical method when P and P^{-1} are stable. Without these assumptions loopshaping can be awkward and the methodical procedure in this chapter can be used. Actually, (1.1) is too hard for mathematical analysis, so a compromise criterion is posed, namely,

$$\||W_1 S|^2 + |W_2 T|^2\|_\infty < 1/2. \tag{1.2}$$

Using a technique called spectral factorization, we can reduce this problem to a model-matching problem. As an illustration, the flexible beam example is reconsidered; besides step response specs on the tip deflection, a hard limit is placed on the plant input to prevent saturation of an amplifier.

Finally, some words about frequency-domain versus time-domain methods of design. Horowitz (1963) has long maintained that "frequency response methods have been found to be especially useful and transparent, enabling the designer to see the tradeoff between conflicting design factors." This point of view has gained much greater acceptance within the control community at large in recent years, although perhaps it would be better to stress the importance of input-output or operator-theoretic versus state-space methods, instead of frequency domain versus time domain. This book focuses almost exclusively on input-output methods, not because they are ultimately more fundamental than state-space methods, but simply for pedagogical reasons.

Notes and References

There are many books on feedback control systems. Particularly good ones are Bower and Schultheiss (1961) and Franklin et al. (1986). Regarding the Keck telescope, see Aubrun et al. (1987, 1988).