Part I

Short Design Course

Chapter 1

A method for solving system design problems

This chapter outlines our approach to solving design problems. Section 1.1 gives basic facts about rational functions, which are the single most important type of functions in practical system design. The basic system and functions considered in this book are introduced in Section 1.2 and Section 1.3. A control design problem is presented in 1.4. A description of the method used in this book to solve control problems is in Section 1.5.

1.1 Rational functions

A rational function can be thought of as a function on the imaginary axis $F(j\omega)$, or as a function on the complex plane, where the notation $F(s)$ is used. If $N(s)$ and $D(s)$ are polynomials, then the rational function $F(s) = \frac{N(s)}{D(s)}$ has *relative degree* given by

$$d(F(s)) := degree(D(s)) - degree(N(s))$$

The function F is called *proper* if $d(F) \geq 0$, and *strictly proper* if $d(F) > 0$.

A function F is said to be *bounded* (on the imaginary axis) if there exists a constant $M > 0$ such that

$$|F(j\omega)| \;\leq\; M, \;\; \forall\omega \in R$$

The smallest of such M is the number $||F||_\infty = \sup_{\omega\in R} |F(j\omega)|$, called the *supremum of* $|F(j\omega)|$ (See Fig. 1.1).

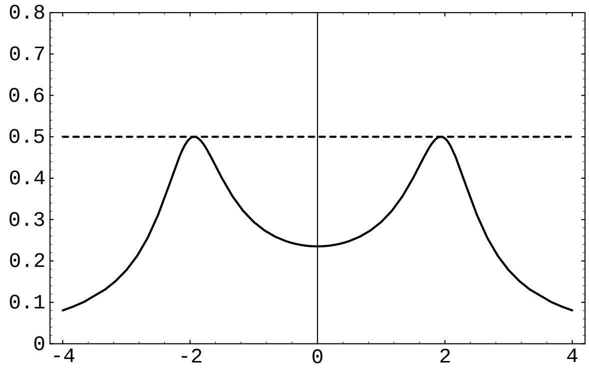


Figure 1.1: Plot of $|f(j\omega)|$ vs. frequency ω, where $f(s) = 1/((s-.5)^2+4)$. The dotted line marks $||f||_\infty$.

A rational function F is bounded if and only if it is proper and has no poles on the imaginary axis; the proof of this is left as an exercise. The value $||F||_\infty$ is not always attained, as the example below illustrates.

Example. The function $F(s) = s^2/(s^2-10.)$ is bounded, since $|F(j\omega)| \leq 1 \; \forall\omega$. However, for this example there is no single finite frequency ω_0 at which $|F(j\omega_0)| = 1$. In other words, the supremum is not attained.

Functions $F(s)$ that satisfy the equality

$$F(s) = \overline{F(\overline{s})}, \tag{1.1}$$

for all s, are sometimes referred to as *real* (on the real axis), and are very important in engineering. They arise as the Laplace transform of real valued functions of time. On the imaginary axis equality (1.1) becomes

$$F(j\omega) = \overline{F(-j\omega)} \tag{1.2}$$

All functions that appear in this book have this property. One can show that a rational function is real if and only if the coefficients are real numbers.

A rational function is *stable* if every pole of the function has negative real part, and the relative degree is not positive.

The usual convention is to denote the set of all rational functions that are stable and real by $\mathcal{RH}^\infty$. Note that functions in $\mathcal{RH}^\infty$ can be described as proper rational functions with real coefficients and with no poles in the closed Right-Hand-Plane RHP [1].

---

[1] Some of the concepts introduced here apply to other (non rational) functions as

1.2 The closed loop system $\mathcal{S}$

The basic system we consider is a linear, time invariant, finite dimensional system in the frequency domain description depicted in Fig. 1.2. The functions $P(s)$ and $C(s)$ are proper, real rational, functions of s. They are called the plant and the compensator respectively. Fig. 1.2 is called the closed loop system and denoted by $\mathcal{S}$. *We assume that the plant in Fig. 1.2 is given and can not be modified, and that many choices of the compensator are possible.*

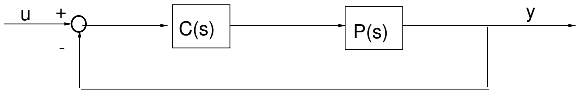


Figure 1.2: The closed loop system $\mathcal{S}$.

Besides $P(s)$ and $C(s)$, there are other functions commonly associated with the closed loop system $\mathcal{S}$, which we now derive. Begin with Fig. 1.2 to obtain the equations

$$\begin{aligned} y(s) &= P(s)C(s)e(s) \\ e(s) &= u(s) - y(s) \end{aligned} \tag{1.3}$$

Combine both equations in (1.3) to obtain

$$y(s) = P(s)C(s)u(s) - P(s)C(s)y(s)$$

Solve for $y(s)$ and obtain

$$y(s) = (1 + P(s)C(s))^{-1} P(s)C(s)u(s) \tag{1.4}$$

Equation (1.4) has in it the closed loop transfer function of $\mathcal{S}$, defined as

$$T = \frac{PC}{1 + PC}. \tag{1.5}$$

Note that the variable s has been suppressed in (1.5) to simplify notation. Other key functions are:

- The sensitivity function $S := (1 + PC)^{-1}$.

---

well. For example, a continuous function $F(j\omega)$ may be bounded or may satisfy (1.2). However, common mathematical techniques that work with rational functions do not work with more general types of functions (see [Fo]).

- The tracking error for input u, Su.
- The closed loop compensator $Q := CS$.
- The open loop transfer function $L := PC$.

1.3 Designable transfer function

Roughly speaking, to design a system one must choose all of the system's parts that have not been given in advance. This is done according to some criteria or requirements.

The most obvious way to select all the parts of system $\mathcal{S}$ is to specify C. Doing this determines the closed loop system $\mathcal{S}$ completely, in the sense that all its functions T, L, S, Q, etc. can be written in terms of the given part P and our selection for C.

In this book we will use another way to select a closed loop system $\mathcal{S}$. Think of T as a variable[2]. If T is specified, then one can calculate C by solving in (1.5):

$$C = \frac{T}{(1-T)P} \tag{1.6}$$

Now we see that it is possible to write C and every function in terms of T and the given part P. That is, to determine the closed loop system $\mathcal{S}$ one has only to pick a particular value for T. Under these circumstances we refer to T as a *designable transfer function.*

The designable transfer function cannot take an arbitrary form. There are many properties that the physics or mathematics of the system dictate. There is a set $\mathcal{I}$ of admissible functions to which the designable T must belong. For example, $\mathcal{I}$ contains functions that are continuous on the imaginary axis and uniformly bounded there. Clearly, the set $\mathcal{I}$ must be defined before the design process begins. In this book the set $\mathcal{I}$ is a subset of $\mathcal{RH}^{\infty}$.

1.4 A system design problem

We now formalize the generic problem we attempt to solve in this book.

The designer rejects or accepts a system by comparing the actual characteristics of the system to a set of requirements that describes what is

---

[2] This has the advantage that design specifications are usually presented in terms of the closed loop transfer function T. It also has a mathematical advantage which is important to the optimization algorithms which underlie our method.

expected from the system. The two main types of requirements we consider are internal stability and performance.

The system $\mathcal{S}$ is internally stable if its closed loop transfer function T is stable and there is no cancellation of unstable poles and zeros in the product PC. Internal stability is treated in Chapter 2.

In the simplest possible terms, a performance requirement is an inequality in terms of the designable closed loop transfer function T. An example is

$$|T(j\omega)| < 0.707, \qquad for\ w > 1.0$$

Performance requirements are discussed in Chapter 3 and Chapter 6.

We state the design problem as follows:

Design	Given a plant P, a set of performance requirements $\mathcal{P}$ and a set of feasible functions $\mathcal{I} \subset \mathcal{RH}^{\infty}$, determine if there exists $T \in \mathcal{I}$ which makes the closed loop system $\mathcal{S}$ internally stable and satisfy all the performance requirements in $\mathcal{P}$. If the answer is yes, find one such T.

1.5 The method

A solution to *Design* is found by the following sequence of steps:

I. Obtain a mathematical description of the performance and internal stability requirements.

Some designable transfer functions T give rise to internally stable systems. These T can be described with a formula (see Chapter 2).

Each performance requirement is an inequality in terms of the designable transfer function T. In Chapter 3 we present a list of frequency domain performance requirements from which the designer can choose to formulate problems. This list is expanded in Chapters 4 and 5 by adding time-domain requirements.

II. Obtain a performance function and a performance index.

The mathematical expressions for the performance requirements are combined to form a *performance function* $\Gamma(\omega, T(j\omega))$, a positive valued function of T and ω. In many cases the performance function is defined in such a way that a designable T satisfies the requirements if and only if

$$\Gamma(\omega, T(j\omega)) \leq 1, \qquad (\forall \omega) \tag{1.7}$$

By considering the worst case performance over all frequencies, a *performance index* that depends on the closed loop transfer function T is obtained:

$$\gamma(T) = \sup_{\omega} \Gamma(\omega, T(j\omega)) \tag{1.8}$$

The performance index is a single number that gives a measure of goodness of a choice of T. In other words it is a cost function; the smaller $\gamma(T)$, the better the choice of T. Several ways to form Γ from the requirements are discussed in Chapters 4 and 5.

III. Minimize the performance index over all T that correspond to an internally stable system $\mathcal{S}$.

This is the step that requires a special purpose computer optimization program. Minimizing the performance index over all possible T produces two outputs:

- A yes or no answer to the question of whether there exist closed loop systems that meet all the requirements.
- An optimal designable transfer function T^*.

The function T^* is among the closed loop transfer functions that satisfy internal stability requirements the best, in terms of overall performance as measured by the index. Optimization (minimizing the performance index) is treated in Chapter 4.

If it is determined in step III that there are solutions to *Design*, the optimal designable transfer function T^* can be selected to produce simulations, physical realization, etc. However, in some cases it is more appropriate to restate the problem by selecting some of the requirements to make them more stringent (this allows one to put special emphasis on these requirements). This action results in a new Design problem, which is subsequently solved with steps I, II and III.

If the result of step III is that no solutions to *Design* exist, the engineer is confronted with two possible paths. One is to redefine the problem completely by changing the model. The other is to reexamine the requirements to determine which of them can be relaxed. In this case, a problem is formulated with the new set of requirements, and then solved with steps I, II and III.

Since in practice it is common to repeat steps I, II and III several times in the manner described above, we add another step to our list. With step IV below one can treat a sequence of *Design* problems.

IV. If many satisfactory closed loop systems exist or none exist, tighten or loosen the specifications accordingly and go to I. Stop if this process can not be carried further.

The modification of requirements is illustrated with practical examples in Chapter 5.

The mathematics used here produce T^* and C^* given by a set of values on the $j\omega$ axis. In most instances the designer will want to represent C^* as a rational function. In some cases, it is desirable that this rational function has low order. These two objectives can be accomplished with the techniques provided by the subjects of system identification and model reduction. While these subjects are not treated in this book[3], the software package OPTDesign has functions for doing (stable or otherwise) rational fits of data, in either the supremum norm or the ℓ^2 norm.

We finish this chapter with a caveat. It may be desirable when doing system design to select compensators that are stable. For example, this would be the case if the engineer wants to build and test the compensator as an independent unit. In its current state, the theory and computational methods of H^∞ optimization do not handle stability of compensators. Thus optimal compensators C^* produced by the methods used here are not necessarily stable (see [DFT], page 79).

1.6 Exercises

1. Exercises on rational functions.

a. Verify that the properties of being real, bounded, stable, and proper are preserved under the following operations: addition of two rational functions, multiplication of rational functions, multiplication of a rational function by a Real number.

b. Prove that a rational function is real if and only if the coefficients are real numbers.

c. Prove that a rational function F is bounded if and only if it is proper and has no poles on the imaginary axis.

2. Use formula (1.6 in verifying the following relations:

---

[3] The reader is referred to [SZ] and [GKL] for theoretical aspects of system identification and model reduction. See also [Be] and [Th].

a. $\frac{dT}{dP} = S\frac{T}{P}$

b. $S + T = 1$

c. $L = \frac{T}{1-T}$

3. Write all functions L, C, S, Q in terms of P and T.

4. Consider a closed loop system $\mathcal{S}$ where T is stable and no RHP pole-zero cancelation occurs in PC. Prove that in this case,

a. if s_0 is a zero of P in the RHP, then $T(s_0) = 0$.

b. if s_1 is a pole of P in the RHP, then $T(s_1) = 1$.

5. Write as a frequency domain performance requirement the relation Rewrite the relation

$$|S(j\omega)| < 0.5 \qquad for\ 0 \leq \omega < 1$$

in terms of $T(j\omega)$.

1.7 References

[Be] Herbert Beke, Algorithm GKL, Documentation for computer code GKL. Department of EE, University of Minnesota, 1992.

[Fo] Ciprian Foias. Commutant Lifting Techniques for Computing Optimal H^∞ Controllers. In: H^∞-control Theory. With B. Francis,H. Kwakernaak, B. Pearson. Springer Verlag Lecture Notes Vol. 1496,1991.

[GKL] G.Gu, P. Khargonekar, and B. Lee, Approximation of Infinite-Dimensional Systems, IEEE TAC, Vol. 34, No. 6, June 1989.

[Th] L. N. Trefethen. Matlab programs for CF approximation. Numerical Analysis Report 86-3. Dep. of Math. MIT. June 1986.

[SS] Z. Shaifei and A. T. Shenton. Theory and application of H^∞ disk method. Report No. MES/ATS/BAE/002/90, Department of Mechanical Engineering, University of Liverpool, U.K., October 1990.

Chapter 2

Internal Stability

This chapter is an introduction to internal stability of systems. The main concept is discussed in Section 2.1, where it is illustrated with examples. Internal stability is closely connected to a mathematical topic called interpolation, and that is the subject of Section (8). We conclude the chapter with a section on control systems for a given stable plant. The chapter gives the student quick overview of internal stability, as well as enough background to treat design problems of certain types. The discussion of internal stability in full generality is left to the more advanced student and is presented in Chapter 8.

2.1 Control and stability.

Consider the closed loop system $\mathcal{S}$ depicted below.

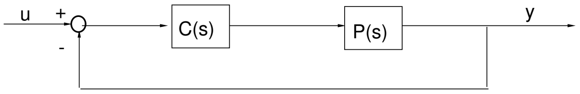


Figure 2.1: The closed loop system $\mathcal{S}$.

In Fig. 2.1, P and C are proper real rational functions. Given a plant P, by choosing an appropriate compensator C one can obtain a transfer function T for the closed loop system that is stable, i.e., that has no poles in the closed RHP. One way to achieve this is to choose C that cancels RHP poles and zeros of P. However, any RHP pole-zero cancellation in the product PC is highly undesirable, because small uncertainties in the plant P or in the compensator's construction lead to a radical change in the behavior of the closed loop system $\mathcal{S}$. We now illustrate this point with an example.

Example. For the given plant $P(s) = 1/(s-1)$, consider the compensators $C_1(s) = (s-1)/(s+1)$ and $C_2(s) = 2$. Observe that a RHP zero of the compensator C_1 cancels a RHP pole of the plant when the product $P(s)C_1(s)$ is formed:

$$P(s)C_1(s) = \frac{1}{s-1}\frac{s-1}{s+1} = \frac{1}{s+1}. \tag{2.1}$$

That the closed loop transfer function T_1 associated with C_1 is stable is clear from the following calculation:

$$T_1(s) = \frac{P(s)C_1(s)}{1+P(s)C_1(s)} = \frac{\frac{1}{s+1}}{1+\frac{1}{s+1}} = \frac{1}{s+2}$$

Now consider a small perturbation of the plant corresponding to a change in the location of the pole. Thus we now have the plant

$$P^\delta(s) = \frac{1}{s-1+\delta} \tag{2.2}$$

where δ is a real number with small absolute value. For this plant and the compensator C_1, the closed loop transfer function is

$$T_1^\delta(s) = \frac{P^\delta(s)C_1(s)}{1+P^\delta(s)C_1(s)} = \frac{\frac{s-1}{(s+1)(s-1+\delta)}}{1+\frac{s-1}{(s+1)(s-1+\delta)}} = \frac{s-1}{(s+1)(s-1+\delta)+(s-1)}$$

The function T_1^δ has an unstable pole at $s = s_\delta$ for all small values of δ, except for $\delta = 0$. Fig. 2.2 shows the plot of s_δ. Here

$$s_\delta = \frac{1}{2}\left(-(1+\delta)+\sqrt{(1+\delta)^2-4(\delta-2)}\right).$$

On the other hand, for the plant P^δ and compensator C_2 the closed loop transfer function is

$$T_2^\delta(s) = \frac{P^\delta(s)C_2(s)}{1+P^\delta(s)C_2(s)} = \frac{\frac{2}{s-1+\delta}}{1+\frac{2}{s-1+\delta}} = \frac{2}{s+1+\delta}.$$

Note that T_2^δ is stable for all small δ, while T_1^δ is not.

In this example we see that cancellation of a RHP pole of P with a zero of C produces an undesirable system, since the latter becomes unstable under

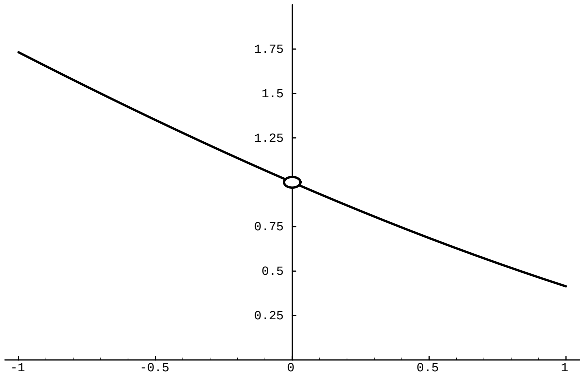


Figure 2.2: Plot of the unstable pole s_δ of P^δ as a function of δ. The parameter δ is on the horizontal axis. The point $(0, 1)$ is omited, since for $\delta = 0$ there is no unstable pole in P^δ.

small perturbations of the plant. A similar phenomenon occurs when a RHP zero of the plant is cancelled with a pole of the compensator (Exercise 1). Thus stability of T, sometimes referred to as *external stability* of the closed loop system, is not sufficient to guarantee a satisfactory closed loop system $\mathcal{S}$, although it is a necessary ingredient in the practical design of systems.

The closed loop transfer function describes external, or input-output, behavior of the system. A more satisfying concept of stability is that of *internal stability.*

Definition 1 *The system $\mathcal{S}$ is internally stable if*

- *i. The closed loop transfer function T is in $\mathcal{RH}^\infty$.*
- *ii. A RHP pole of the plant is not cancelled by a RHP zero of the compensator.*
- *iii. A RHP zero of the plant is not cancelled by a RHP pole of the compensator.*

Here $\mathcal{RH}^\infty$ is the linear space of all proper rational functions that are stable (poles off the closed *RHP*) and real (real coefficients).

The fundamental problem we treat in this chapter is, given a plant P in a certain class, find a description of *all* the internally stable systems $\mathcal{S}$ with P as plant. The same problem is discussed in Chapter 8 for any plant P.

There are many ways in which one can present an answer to this problem, all of which are mathematically equivalent. In this book we choose one that consists in writing *a formula* that is (and must be) satisfied by all designable closed loop transfer functions T of these systems. The reasons for choosing this particular approach are technical rather than physical: we have available a procedure that uses this formula to give an answer to system design problems.

To derive formulas for internally stable systems we need first the concept of interpolation condition which is presented in the next section.

2.2 Interpolation.

A basic result from elementary analysis states that given a point s_0 in the complex plane and a polynomial f, then

$$f(s_0) = 0 \tag{2.3}$$

if and only if one can write

$$f(s) = (s - s_0)h(s) \tag{2.4}$$

where h is some polynomial.

The equality (2.3) is called an interpolation condition. One can prove that (2.4) gives a formula or parameterization of all functions f satisfying (2.3). In this chapter the functions f are rational functions, rather than just polynomials.

To illustrate the role of interpolation in internal stability, we consider the case of a system $\mathcal{S}$ with a plant P that has a simple RHP pole at $s = s_0$. In the discussion below the following interpolation condition will be needed:

$$T(s_0) = 1 \tag{2.5}$$

Claim: If s_0 is a simple RHP pole of P and T satisfies (2.5), then there is no pole zero cancellation in the product $P(s)C(s)$ at $s = s_0$.
Proof: If there is cancellation, then $P(s_0)C(s_0)$ is finite, and this implies that

$$T(s_0) = \frac{P(s_0)C(s_0)}{1 + P(s_0)C(s_0)} \neq 1.$$

This proves the claim.

Claim: If s_0 is a simple RHP pole of P and if there is no RHP pole-zero cancellation in PC at $s = s_0$, then T satisfies (2.5).

Proof: If there is no cancellation in the product $P(s)C(s)$ at $s = s_0$, then $P(s)C(s)$ has an order $n_0 \geq 0$ pole at $s = s_0$. Let

$$\ell := \lim_{s \to s_0} (s - s_0)^{n_0} P(s)C(s)$$

Then ℓ is a number, and

$$\begin{aligned} T(s_0) \quad &= \lim_{s \to s_0} \tfrac{P(s)C(s)}{1+P(s)C(s)} \\ &= \lim_{s \to s_0} \tfrac{(s-s_0)^{n_0} P(s)C(s)}{(s-s_0)^{n_0} + (s-s_0)^{n_0} P(s)C(s)} = \tfrac{\ell}{0+\ell} = 1. \end{aligned}$$

More is true:

Proposition 1 *Let $\mathcal{S}$ be a system with plant P and closed loop transfer function T.*

- *If s_0 is a simple RHP pole of P, then pole-zero cancellation in the product PC does not occur at $s = s_0$ if and only if $T(s_0) = 1$.*
- *If s_0 is a simple RHP zero of P, then pole-zero cancellation in the product PC does not occur at $s = s_0$ if and only if $T(s_0) = 0$.*

One consequence of Proposition 1 is that for certain systems, internal stability is described in terms of a set $\mathcal{J}$ of interpolation conditions such as $T(s_0) = 0$ and $T(s_0) = 1$.

The good thing about interpolation conditions is that one can always write down a formula for all functions T in $\mathcal{RH}^\infty$ that satisfy given interpolation conditions. Such a formula looks like:

$$T(s) = A(s) + B(s)\, T_1(s) \tag{2.6}$$

where A and B are certain functions in RH^∞ that can be determined from $\mathcal{J}$, and T_1 could be any function in RH^∞. Thus as T_1 sweeps $\mathcal{RH}^\infty$, we have T sweeping all functions in $\mathcal{RH}^\infty$ meeting $\mathcal{J}$.

One can show that a system is internally stable if T satisfies (2.6) for the interpolation conditions $\mathcal{J}$ corresponding to poles and zeros of the plant P located in the RHP. A complete discussion of this can be found in Chapter (8), but for now we restrict our study to the case where the plant is stable.

2.3 Systems with a stable plant

Systems $\mathcal{S}$ with a RHP-stable plant are a common type of system occurring in practical control systems. These are also simple enough to provide a good introduction to a study of internal stability.

We now try to answer the question, given a stable plant P, what systems $\mathcal{S}$ have P as plant and are internally stable?

To give an answer we will find a formula for the closed loop transfer functions T that come from such systems. Here is a restatement of the question above: Given a stable plant P, describe all possible closed loop transfer functions T that come from an internally stable system $\mathcal{S}$ with plant P. The answer we seek is in the following theorem.

Theorem 1 *Let P be a strictly proper, RHP-stable plant for the system $\mathcal{S}$. Then the system $\mathcal{S}$ is internally stable if and only if there exists $T_1 \in RH^\infty$ such that $T = PT_1$*

Proof: If $\mathcal{S}$ is internally stable, then PC has the same zeros as P, including multiplicity. Thus the zeros of $T = PC/(1+PC)$ contain the zeros of P, and this implies that the function $T_1 := T/P$ has no RHP poles. The function T_1 is proper since the system $\mathcal{S}$ has proper C and $T_1 = C/(1+PC)$. Thus we have that T_1 is an element of RH^∞. This proves the only if side of the Theorem. Conversely, suppose that $T = PT_1$ for some $T_1 \in RH^\infty$. Note that the compensator is given by

$$C = \frac{T}{P(1-T)} = \frac{T_1}{1-PT_1}$$

It is clear from this formula and from P being strictly proper that C is proper. For C to have a pole at the same location $s = s_0$ as a RHP zero of P one must have $1 - P(s_0)T_1(s_0) = 0$, which is impossible since T_1 has no pole at s_0 and P is zero there. ■

Example. If the plant of an internally stable system is $P(s) = \frac{1}{(s+3.4)^2}$, then any stable closed loop transfer function T has the form

$$T(s) = \frac{1}{(s+3.4)^2} * T_1(s) \tag{2.7}$$

where T_1 is some (or any) element of RH^∞. We note that T_1 may not be strictly proper. In fact, $d(T_1) \neq 0$ forces $d(C) = 0$ upon the system.

Note that we used the actual plant P in the formula (2.7). While this may be convenient, many other formulas for T are possible. For example, we can correctly write

$$T(s) = \frac{1}{(s+1)^2}\, T_1(s) \qquad \forall T_1 \in \mathcal{RH}^\infty \tag{2.8}$$

to describe all T from internally stable $\mathcal{S}$.

Example. If now $P(s) = (s^2 - 4)/(s^4 + 2s^2 + 1)$ then P is stable and strictly proper, so for the system to be internally stable the closed loop transfer function must have the form

$$T(s) = \frac{s^2 - 4}{s^4 + 2s^2 + 1}\, T_1(s)$$

for some $T_1 \in RH^\infty$.

2.4 Exercises

1. Consider the family of plants given by

   $$P^\delta(s) = (s - 1 + \delta)/(s + 1)$$

   where δ is a real constant. Show that

   a. for small $|\delta|$, the compensator $C_1(s) = 1/(s-1)$ and plant P^δ yields a closed loop transfer function T^δ that is stable if and only if $\delta = 0$.

   b. for all small $|\delta|$, the compensator $C_2(s) = (s+1)/(s+3)$ and plant P^δ yields a stable closed loop transfer function T^δ.

Chapter 3

Frequency domain performance requirements

This chapter treats frequency domain performance requirements. These have a convenient graphical interpretation in terms of disks. Section 3.1 reviews basic concepts and introduces disk inequalities. The most common performance requiurements for control systems are given in Section 3.2. Section 3.3 discusses disk inequalities arising from performance requirements. The first time reader may stop here; the topics covered at this point are enough to give basic tools to solve simple design problems, when used in conjunction with material from Chapter 2 and Chapter 4. The more interested reader may continue with Sections 3.4 and 3.5, which explain additional measures of performance and the corresponding disk inequalities.

3.1 Introduction

3.1.1 The closed loop system $\mathcal{S}$

Consider the closed loop system depicted in Fig. 3.1. In this system, $P(s)$ and $C(s)$ are proper real rational functions of s. We take the plant P to be a given rational function, and use the closed loop transfer function T to parameterize the closed loop systems $\mathcal{S}$ obtained with different compensators. Hence many systems $\mathcal{S}$ with given plant P are possible by letting the designable transfer function T take different values. Key functions are

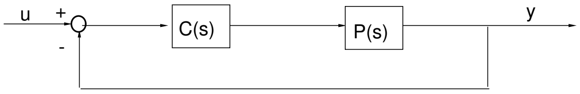


Figure 3.1: The closed loop system $\mathcal{S}$.

- The closed loop function $T = \frac{PC}{1+PC}$
- The sensitivity function $S = (1 + PC)^{-1}$
- The tracking error for input u, Su
- The closed loop compensator $Q := CS$
- The open loop transfer function $L := PC$
- The closed loop plant PS

3.1.2 Frequency domain performance requirements

Two types of requirements imposed on the system $\mathcal{S}$ are discussed in this book: internal stability and performance. The internal stability requirement was studied in Chapter 2. Performance requirements are inequalities involving the functions of the system. In this chapter we focus on performance requirements in the frequency domain, defined below.

Definition. A *frequency domain performance requirement* is an inequality that involves the designable transfer function T, and an interval in ω for which the inequality is required to hold.

The function

3.1.3 Disk inequalities

All the frequency domain performance requirements treated in this chapter can be written in the form of a *disk inequality*:

$$|K(j\omega) - T(j\omega)| \leq R(j\omega), \qquad \omega_a \leq \omega \leq \omega_b \tag{3.1}$$

Disk inequalities are easy to plot as regions in 3-D space (see Fig. 3.2).

In many cases the requirements give one disk inequality on one frequency band, and other disk inequalities on other frequency bands. When the various frequency bands are disjoint, it is possible to piece these disk inequalities together into a single one, valid for all frequencies:

$$|K(j\omega) - T(j\omega)| \leq R(j\omega), \quad (\forall\ \omega) \tag{3.2}$$

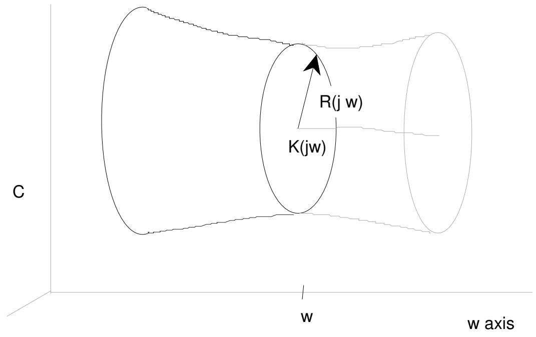


Figure 3.2: Tube like region $\{(\omega, z) : \; |k(j\omega) - z| \leq r(j\omega)\}$ determined by a disk inequality $|k(j\omega) - T(j\omega)| \leq r(j\omega)$.

In these cases the inequality has a nice graphical representation, as shown in Fig. 3.3. We shall see in Chapter 4 how to use disk inequalities to pose optimization problems of the simplest kind discussed in this book. Many system design problems can be solved by finding solutions to these optimization problems.

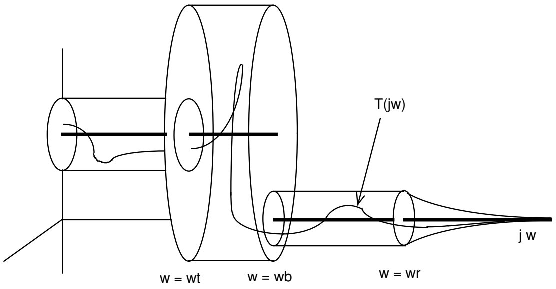


Figure 3.3: Region determined by piecing together disk inequalities.

We see in Fig. 3.2. how a disk inequality is represented graphically by a tube-like domain . If the frequency ω is fixed, then a single disk inequality is represented in the complex plane by a region S_ω consisting of a solid disk. However, if two or more disk inequalities apply at the same frequency ω_0, the region S_ω is the intersection of the disks that corresponds to individual constraints. Thus in this case, S_ω is not a disk. This situation, which is common in systems design, is treated in Chapter 5.

3.2 Measures of performance

A set of basic frequency domain performance requirements (gain-phase margin, tracking error, bandwidth and closed loop roll-off) are presented in this section. More requirements are introduced later in the chapter; however, it is possible to state physically sensible design problems with the requirements given in this section alone.

3.2.1 Gain-phase margin

In classical control, the gain margin and phase margin are the chief measures of how robustly stability is achieved. In order to formulate constraints in terms of the closed loop transfer function T we use a composite measure of stability, which seems at least as good as the gain margin and phase margin taken together. Define the *gain-phase margin* of the system $\mathcal{S}$ to be

$$m = \inf_{\omega} |1 + PC(j\omega)| \tag{3.3}$$

Graphically m is just the distance of the Nyquist plot of PC to the point -1 in the complex plane. Simple algebra converts (3.3) to

$$|T(j\omega) - 1| \leq \frac{1}{m} \quad \text{for all} \quad \omega \tag{3.4}$$

We can easily compare the gain-phase margin m with the gain margin g and the phase margin ϕ by looking at the Nyquist plot in Fig. 3.4. Typically m is more conservative than either ϕ or g.

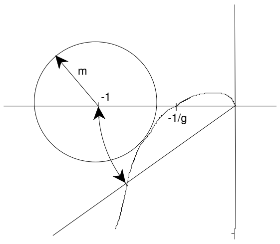


Figure 3.4: The gain margin g, phase margin ϕ, and g-p. margin m

If a given stable system has m near 0, then it is close to the unstable case, which is undesirable. If α_m denotes the largest value of $1/m$ considered to be acceptable in inequality (3.4), then we can formulate a constraint in terms of α_m as (see Fig. 3.5):

$$|1 - T(j\omega)| \leq \ \alpha_m \ \text{(for all } \omega) \ . \tag{3.5}$$

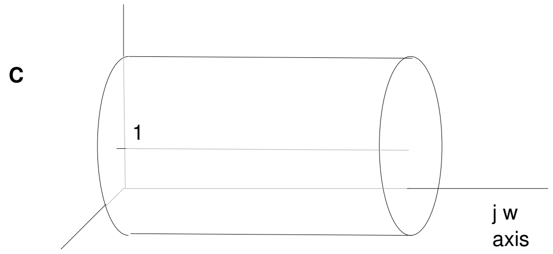


Figure 3.5: Region defined by the gain-phase margin constraint.

Fancier notions are very natural. For example, if W is a given nonnegative function of frequency, we can define

$$m_W \triangleq \inf_{\omega} |W(j\omega)(1 + PC)(j\omega)| \tag{3.6}$$

to be the *weighted gain-phase margin.* The corresponding constraint is derived from this definition in the obvious way and produces $\alpha_m(j\omega) = W(j\omega)/m_W$.

Gain-phase margin constraint

For a given α_m in the interval $(0, 1)$,

$$|1 - T(j\omega)| \leq \ \alpha_m \ \text{(for all } \omega) \ . \tag{3.7}$$

3.2.2 Tracking error

Fundamentally, tracking error is a time domain concept. The issue is to measure how close the output $y(t)$ of the closed loop system is to a given input $u(t)$. Indeed, we want the error function

$$\epsilon_u(t) = u(t) - y_u(t) \tag{3.8}$$

to be "small" for each function u in a big class of possible inputs. To use frequency domain techniques, one must translate time domain criteria back to the frequency domain. We do this below.

Consider system $\mathcal{S}$ with given time domain input $u(t)$ and output $y(t)$, and let $U(s)$ and $Y(s)$ be the Laplace transform of $u(t)$ and $y(t)$. Then $Y(s) = T(s)U(s)$, and the Laplace transform of ϵ_u is

$$E_u(s) = (1 - T(s))U(s) \tag{3.9}$$

For $T \in \mathcal{RH}^\infty$, a condition that generates small ϵ_u compared to the size of u is

$$\sup_\omega |1 - T(j\omega)| < \alpha_{tr} \tag{3.10}$$

for some small α_{tr}. Thus good tracking is generated by requiring (3.10) to hold[1].

It turns out that (3.10) is too stringent to be obtainable in a practical engineering problem since the function T rolls off at high frequency. What is realistic is to require the system to track low frequency functions well, that is, to require that $T(j\omega)$ be close to 1 over some specified frequency range $[-\omega_{tr}, \omega_{tr}]$. Thus we introduce one of the key constraints of control.

---

[1] A standard theorem (Plancherel theorem, [Y]) applied to (3.9) yields immediately that for $T \in RH^\infty$ the error satisfies a useful bound:

$$\begin{aligned}\int_0^\infty |\epsilon_u(t)|^2 dt \quad &= \int_{-\infty}^\infty |E_u(j\omega)|^2 d\omega = \int_{-\infty}^\infty |1 - T(j\omega)|^2 |U(j\omega)|^2 d\omega \\ &\leq \left\{\sup_\omega |1 - T(j\omega)|^2\right\} \int_{-\infty}^\infty |U(j\omega)|^2 d\omega \\ &= \left\{\sup_\omega |1 - T(j\omega)|^2\right\} \int_0^\infty |u(t)|^2 dt\end{aligned} \tag{3.11}$$

for all functions u on $[0, \infty)$ having finite *energy* $\int_0^\infty |u(t)|^2 dt$. The inequality is sharp in the sense that there are input functions that make this inequality as close as you desire to equality. That is,

$$\sup_u \frac{\int_0^\infty |\epsilon_u(t)|^2 dt}{\int_0^\infty |u(t)|^2 dt} = \sup_\omega |1 - T(j\omega)|^2 \tag{3.12}$$

Consequently, if

$$\sup_\omega |1 - T(j\omega)|^2 \leq c \tag{3.13}$$

then the normalized error in the left-hand-side of (3.12) is no larger than c. Thus specs of the form (3.13) guarantee that

$$\text{energy}(\epsilon_u) \leq c \ \text{energy}(u)$$

for all u with finite energy. This is a very strong form of tracking.

Tracking Error Constraint

For given $\alpha_{tr} > 0$, $\omega_{tr} > 0$,

$$|1 - T(j\omega)| < \alpha_{tr} \quad (|\omega| < \omega_{tr})\ . \tag{3.14}$$

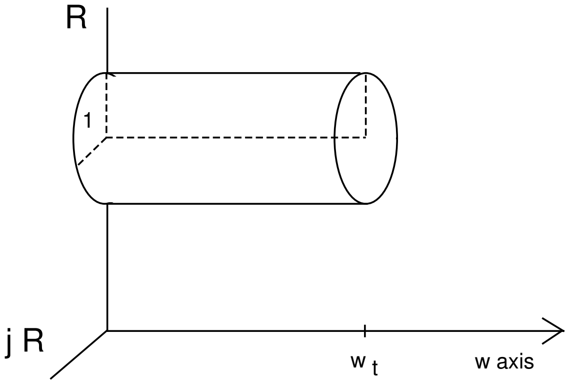


Figure 3.6: Region defined by the tracking error constraint

3.2.3 Bandwidth

Bandwidth of a system is commonly defined as the frequency ω_b at which $|T(j\omega)|$ falls below a given constant times the low frequency value of the input. This constant usually is taken to be $\alpha_b = \frac{1}{\sqrt{2}}$. A particular bandwidth is required to insure that the system at high frequency is not upset by noise, plant uncertainty, actuator sluggishness, etc.

Bandwidth Constraint

Given $\alpha_b > 0$, $\omega_b > 0$,

$$|T(j\omega)| < \alpha_b, \ \text{for} \ |\omega| > \omega_b\ . \tag{3.15}$$

In practice a more refined constraint is required for very high frequency, since it is desirable that T rolls off to 0. This is discussed in Section 3.2.4.

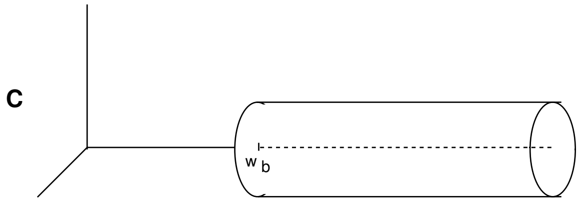


Figure 3.7: Region defined by the bandwidth constraint.

3.2.4 Closed loop roll-off

The open loop transfer function $L = PC$ of a given system rolls off at high frequency, since the plant P and the compensator C are strictly proper. That is, $|P(j\omega)C(j\omega)| \to 0$ as $\omega \to \infty$.

From the relation

$$\begin{aligned} |T(j\omega)| &\leq \frac{|P(j\omega)C(j\omega)|}{1-|P(j\omega)C(j\omega)|} && (3.16) \\ &\approx |P(j\omega)C(j\omega)| \text{ (large } |\omega|) && (3.17) \end{aligned}$$

we see that both T and PC roll off at the same rate. To obtain an inequality useful for the design process, we must eliminate the compensator C from the right-hand-side of relation (3.17). To do this, use the fact that compensators roll off at high frequency with the asymptotic form[2]

$$C(j\omega) \leq \frac{\alpha_r}{|\omega|^n} \tag{3.18}$$

for some α_r and some n. At the outset of design, an engineer should specify n and α_r for the class of compensators that is to be built [3]. Once α_r and n are specified, combine (3.17) and (3.18) to obtain the closed loop roll-off constraint.

---

[2] Since behavior near $j\infty$ is what is important in (3.18), α_r can be taken to be a weight function of frequency that is not too close to 0 at any given frequency.

[3] We thank L. Desoer for emphasizing this constraint to us.

Closed loop roll-off constraint

For given $\omega_r > 0$, $\alpha_r > 0$ and $n > 0$,

$$|T(j\omega)| \leq \frac{\alpha_r |P(j\omega)|}{|\omega|^n} \quad (|\omega| > \omega_r)\ . \tag{3.19}$$

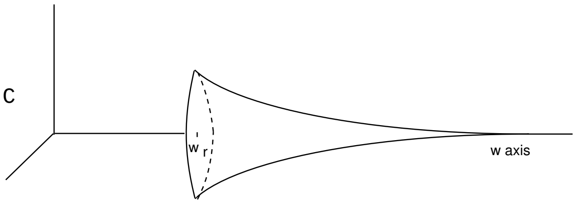


Figure 3.8: Region defined by the closed loop roll-off constraint.

3.2.5 Fundamental tradeoffs

The most fundamental tradeoff in control is between bandwidth constraints and performance such as tracking or gain-phase margin which dominate considerations at low frequency. Bandwidth and roll-off are consistent constraints while tracking error competes strongly with them. Thus a basic challenge in control is to get tracking over a broad enough frequency band, subject to the roll-off constraints dictated by actuator sluggishness and uncertainty in plants, sensors, and the environment. In a particular engineering problem, to obtain any precise feel for this tradeoff requires the use of a computer program.

3.2.6 Choosing sets of performance requirements

The designer must choose a set of performance requirements in order to state and then solve a design problem. This set should be selected with care, since a bad choice leads to problems that do not make sense from either the numerical or the physical point of view. We illustrate this with examples a, and b below.

Example a

Consider the problem *Design* for the plant $P(s) = 1/(1-s)$ and the performance requirement

$$|T(j\omega)| < 0.707, \qquad for\ \omega > 1.0. \tag{3.20}$$

An obvious mistake here is the absence of constraints valid on the band $0 \leq \omega \leq 1.0$. It is easy to find functions T that satisfy (3.20) and that produce internally stable systems $\mathcal{S}$, but that have undesirable behavior at low frequency.

Example b

Now consider *Design* for $P(s)$ with performance requirements

$$\begin{array}{rcll} |T(j\omega)| & < & 0.707, & \text{for } \omega > 1.0 \\ |1 - T(j\omega)| & < & 2.0, & \text{for } 0 \leq \omega \leq 1.0 \end{array} \tag{3.21}$$

For any physical system, T must roll off to 0 at very high frequency. This is not enforced by the constraints (3.21). One way to remedy this is to require roll-off on T, for example with the constraint

$$|T(j\omega)| < \alpha_c |P(j\omega)|, \ for\ \omega > \omega_c$$

The examples above illustrate two basic principles for choosing a set of frequency domain performance requirements:

- At each frequency ω, there must be a constraint in the set of requirement that is active at this frequency.
- Behavior of the system at very high frequency must be specified with a roll-off constraint on T.

There are cases where performance measures not described in the present section are important. In particular, if the plant has a zero or a pole right on the $j\omega$ axis or near it, the *compensator bound* and *plant bound* constraints must be used to set up the problem correctly. See Section 3.4 for details on this.

3.3 Disk inequalities

The performance measures introduced in Section 3.2 can be expressed as *disk inequalities*. In this section, we present an example in which several disk inequalities are put together and expressed as one disk inequality that is valid for all frequencies. Being able to combine disk inequalities as a single one is a necessary skill for students of control who want to solve system design problems with the methods proposed in this book. We note that it is not always possible to combine several disk inequalities as a single one; however, the type of problem that results if this is done leads to the simplest optimization problem[4] when solving the problem *Design*. Such problems are treated on Chapter 4 and on first reading one may want to go from this section to Chapter 4.

Consider a case where the requirements are given by constraints only on the tracking error, gain-phase margin, and bandwidth [5]. We will formulate a single disk constraint from the given constraints.

Suppose the following constraints are given:

- $|1 - T(j\omega)| < 0.25$ for $\omega < 0.5$ (tracking for unit step input).
- $|1 - T(j\omega)| \leq 2.$ for all ω (gain-phase margin).
- $|T(j\omega)| \leq 0.707 \quad \text{for } |\omega| > 3$ (bandwidth).

Note that for ω fixed, the large disks from the gain-phase margin constraint contain the smaller disks from the bandwidth and the tracking error constraints. Thus the smaller disks define the region for either low or high frequencies. For midrange frequencies, the gain-phase margin constraint is the only constraint that is applicable. We collect this information in Table 3.1. The notation there is the same as the notation in inequality (3.1). The region defined by these constraints is drawn in Fig. 3.9.

Constraint	Band	$K(j\omega)$	$R(j\omega)$
Tracking	$0. \leq \|\omega\| < 0.5$	1	0.25
g-p margin	$0.5 < \|\omega\| < 3$	1	2.
Bandwidth	$3 < \|\omega\|$	0	0.707

---

[4] That is, an optimization problem with circular performance function.

[5] This is not a physical example, since there is no decay requirement on the magnitude of T.

Table 3.1

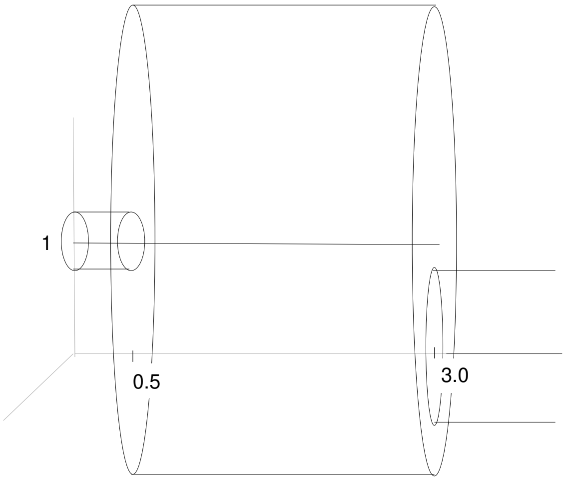


Figure 3.9: The region defined by the constraints in Table 3.1.

3.4 More performance measures

In Section 3.2 we showed how to convert the most basic specifications in a control problem to precise performance measures (suitable for computer optimization). A beginner who masters these alone will have a very good feel for the basics of control[6]. However, there are common control problems that cannot be formulated correctly using only the constraints in Section 3.2; for example, if the plant has a pole or a zero on the $j\omega$ axis. In this section we give additional performance measures, in particular the ones that deal with zeros or poles of the plant on the $j\omega$ axis.

3.4.1 Peak magnitude

One common constraint is that the magnitude of the closed loop transfer function T not become too large, because if it does then T is close to unstable. Also, it has been seen in practical design (see [Hor], page 190) that the peak magnitude of T is related to a large overshoot in the step response function. In many cases a peak value of 1.1–1.5 is acceptable, while

---

[6] Also, the examples in Chapter 4 and Section 5.2 will be clear to him.

in others this is too high. We now define the peak magnitude constraint; the corresponding graphical representation is Fig. 3.10.

Peak Magnitude Constraint

For given $\alpha_M > 0$,

$$|T(j\omega)| \leq \alpha_M \quad (\text{all } \omega) . \tag{3.22}$$

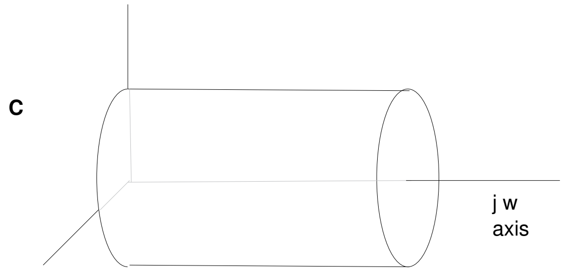


Figure 3.10: Region defined by the peak magnitude constraint.

3.4.2 Compensator bound

Desoer and Gustafson proposed another very sensible criterion for good compensation in [DG1]. It is that

$$|C(j\omega)(1 - T(j\omega))| \leq \alpha_c \ \ (\text{all } \ \omega) \tag{3.23}$$

for some prescribed number α_c. In terms of the closed loop system $\mathcal{S}$, this says that the "closed loop compensator" has magnitude bounded by α_c. The *closed loop compensator* is what the compensator puts out in response to an input to the system. The reason that (3.23) is imposed on the system is that if it is violated, then the closed loop system can saturate, a small current into the system can cause arcing at the output of C, or other problems may appear. Doyle and Stein discuss in [DS] drawbacks of high loop gain in somewhat different terms. Their main concern is that (3.23) holds at high frequencies (low frequencies are less important).

Let us analyze (3.23) in terms of T. Since $PC = T/(1-T)$, we obtain

$$|T(j\omega)| \leq \alpha_c |P(j\omega)| \ \ (\text{all } \ \omega) . \tag{3.24}$$

Typically this constraint adds no serious restriction unless $|P|$ is small (e.g., as $\omega \rightarrow \infty$ or near zeros of P on the $j\omega$-axis). Inequality (3.24) adds a restriction that is binding near the zeros of P ($\omega = \infty$ included). Thus the designer should require (3.24) to hold whenever $|\omega - \omega_z| < \eta$, where $j\omega_z$ is any zero of P on the axis and η is a small positive number. Note that for large ω, inequality (3.24) is contained in the closed loop roll-off constraint (set $n = 0$ in (3.19)).

Compensator Bound Constraint

For given $\alpha_c > 0$, η_z and for any zero $j\omega_z$ of P on the $j\omega$ axis,

$$|T(j\omega)| < \alpha_c |P(j\omega)|, \quad \text{whenever} \quad |\omega - \omega_z| < \eta_z \ . \tag{3.25}$$

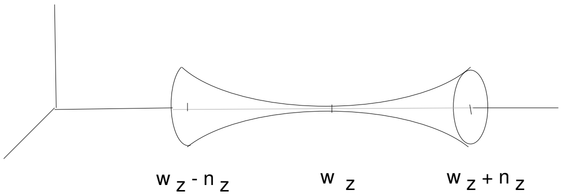


Figure 3.11: Region defined by the compensator bound constraint, if there is a zero of P at $s = j\omega_z$.

3.4.3 Plant bound

Another constraint is that the "closed loop plant" be bounded by a specified value α_p. That is,

$$|(1 - T(j\omega))P(j\omega)| \leq \alpha_p \qquad (\forall \omega) \tag{3.26}$$

The closed loop plant is the output of the closed loop system of Fig. 3.1 to an input to the plant.

Inequality (3.26) is analyzed in similar fashion to inequality (3.24). Recall that internal stability of the system implies that $T(j\omega) = 1$ when

$P(j\omega) = \infty$. Inequality 3.26 is binding near the poles of P; it must hold whenever $|\omega - \omega_p| < \eta_p$, where ω_p is any pole of P on the $j\omega$ axis and η_p is a small positive number. From this we get the plant bound constraint, illustrated in Fig. 3.12.

Plant Bound Constraint

For given $\alpha_p > 0$ and $\eta_p > 0$ and all poles $j\omega_p$ of $P(j\omega)$,

$$|1 - T(j\omega)| < \frac{\alpha_p}{|P(j\omega)|} \quad \text{whenever} \quad |\omega - \omega_p| < \eta_p \ . \tag{3.27}$$

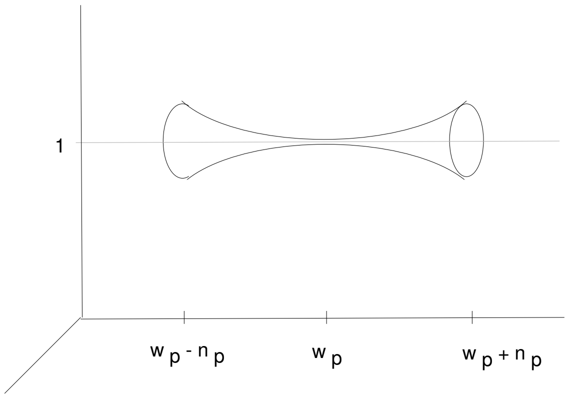


Figure 3.12: Region defined by the plant bound constraint, if P has a pole at $s = j\omega_p$.

3.4.4 Disturbance rejection

Systems are affected by disturbances, and one of the purposes of feeback control is to minimize the effect of these disturbances. As an example, consider the system given by an airplane, with input (established by the pilot) a certain yaw rate $u(t)$, and output the actual yaw rate of the airplane $y(t)$. A gust of wind produces yaw and so it affects the output of the system.

A simple way of modelling this situation is shown in Figure 3.13. There, the disturbance is represented as an additive input to the system at the output to the plant. If there is no other input to the system in Fig. 3.13,

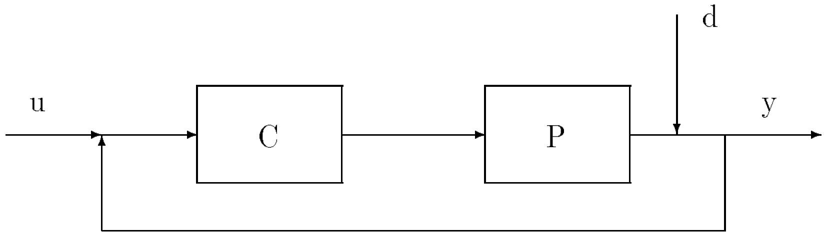


Figure 3.13: Disturbance in a feedback system.

the transfer function from disturbance d to plant output y is easily found to be

$$T_{d \to y} = \frac{1}{1 + P(s)C(s)} \tag{3.28}$$

For good behavior of the feedback system it is necessary that $T_{d \to y}$ be RHP-stable. This is always the case with internally stable systems. More can be required, for example, that there is a restriction on the size of $T_{d \to y}$. The following inequality gives a precise statement, where c is a given constant:

$$|\frac{1}{1 + P(j\omega)C(j\omega)}| \leq c \quad \forall \omega \tag{3.29}$$

We have already encountered inequalities similar to (3.29) before under the name tracking error constraint, so at this point we refer the reader to the discussions of the latter.

3.4.5 More on tracking error

Consider a system $\mathcal{S}$ with input u and output y. The steady-state error for input $u(t)$ is

$$e_{ss} = \lim_{t \to \infty} u(t) - y(t) \tag{3.30}$$

Relation (3.30) has a counterpart in the frequency domain, which can be obtained with the Final Value Theorem. Applying this theorem to 3.30 we obtain

$$e_{ss} = \lim_{s \to 0} s(U(s) - Y(s)) \tag{3.31}$$

Since $Y(s) = T(s)U(s)$, it follows from (3.31) that

$$e_{ss} = \lim_{s \to 0} (1 - T(s))sU(s) \tag{3.32}$$

If α_{tr} denotes the largest acceptable value of e_{ss} in (3.32), then

$$|e_{ss}| = |\lim_{s \to 0} (1 - T(s))sU(s)| \leq \alpha_{tr} \tag{3.33}$$

If α_{tr} is not 0, one can write an inequality in the variable ω that guarantees (3.33). We call this inequality the tracking error constraint for input $U(s)$.

Tracking Error Constraint for input U

For given $\alpha_{tr} > 0$, $\omega_{tr} > 0$,

$$|(1 - T(j\omega))(j\omega)U(j\omega)| < \alpha_{tr} \quad (|\omega| < \omega_{tr}) \ . \tag{3.34}$$

We now discuss the case where the plant P has n pure integrations (a pole of order n at $s = 0$). In this case, the system $\mathcal{S}$ is called a "type n" system.

We assume that internal stability is a requirement, so RHP poles of the plant P are not canceled by zeros of the compensator C. Since the system is type n, the product PC also has a pole of order n at $s = 0$. In this situation, the closed loop transfer function T satisfies the interpolation conditions

$$T(0) = 1, \ T'(0) = 0, \ \ldots, \ T^{(n-1)}(0) = 0 \tag{3.35}$$

Let us now take $m \leq n + 1$ and $U(s) = 1/s^m$. Because of (3.35), the function

$$(1 - T(s))sU(s) = (1 - T(s))s^{1-m} \tag{3.36}$$

has no poles on the closed RHP, except possibly a pole of order 1 at $s = 0$. Therefore the Final Value Theorem applies and one has

$$e_{ss} = \lim_{s \to 0} (1 - T(s))s^{1-m} \tag{3.37}$$

By combining (3.35) with (3.37), and applying L'hopital's rule for limits we obtain

Proposition 2 *An internally stable type n system $\mathcal{S}$ produces a steady state error e_{ss} for input $U(s) = 1/s^m$ given by*

$$e_{ss} = \begin{cases} 1 - T(0) & \textit{if } m = 1 \\ \\ -\frac{T^{(m-1)}(0)}{(m-1)!} & \textit{if } 1 < m \leq n+1 \end{cases} \tag{3.38}$$

Hence $e_{ss} = 0$ if $1 \leq m \leq n$.

Proposition 2 implies that for internally stable type n systems, the most appropriate input $U(s) = 1/s^m$ in the constraint (3.34) is $U(s) = 1/s^{n+1}$. Other inputs U that may be suitable are functions that are stable on the closed RHP, except for a pole of order $n+1$ at $s = 0$.

Common time domain specifications on type n systems are on the position, velocity and acceleration error constants [7] K_0, K_1 and K_2, where

$$K_m = \lim_{s \to 0} s^m P(s) C(s) \tag{3.39}$$

One can show that an internally stable type n system $\mathcal{S}$ satisfies $K_0 = T(0)/(1 - T(0))$ and $K_m = \frac{-m!}{T^{(m)}(0)}$ if $1 \leq m \leq n$.

3.5 A fully constrained problem

In Section 3.3 we obtained a single disk inequality from performance requirements bandwidth, tracking and gain-phase margin. We consider here a similar problem, but now with requirements on bandwidth, closed loop roll-off, plant bound, tracking and gain-phase margin. Again, we will derive a single disk inequality from the requirements.

The given plant is

$$P(s) = \frac{1 - s/5}{s(1-s)}$$

with requirements

a. $|1 - T(jw)| < 0.3$, for $0 \leq \omega < 1.0$.

b. $|1 - T(jw)| < 1/|P(j\omega)|$, for $0 \leq \omega < 1.0$.

c. $|1 - T(jw)| < 2.$ for all ω.

d. $|T(j\omega)| < 0.707$, for $2 < \omega$.

e. $|T(j\omega)| < 4|P(j\omega)|$, for $2 < \omega$.

---

[7] denoted by K_p, K_v and K_a in most books

The constraints a, b and c are active at all frequencies $0 \leq \omega < 1$. Clearly constraint c is contained in constraint a, so we restrict attention to a and b. Solve the equation

$$\frac{1}{|P(j\omega)|} = 0.3$$

to obtain the solution $\omega_0 \approx 0.29$. Thus b is more stringent than a on the frequency band $0 \leq \omega \leq 0.29$, and b is less stringent than a on the band $0.29 \leq \omega < 1$. We see that c is the only constraint that applies on the band $1 \leq \omega \leq 2$. For $\omega \geq 2$ it is clear that every T that satisfies d also satisfies c, so it is enough to study constraints d and e on this band. We solve the equation

$$0.707 = 4|P(j\omega)|$$

and obtain as solution $\omega_1 \approx 2.41$. Therefore constraint d is more stringent on the band $2 \leq \omega < 2.41$, whereas e is more stringent on $2.41 < \omega$. We summarize what we have in Table 3.2, that defines the center function K and the radius function K at each frequency. See Figs. 3.14, 3.15 and 3.16 for a graphical representation of the functions K and R.

Constraint	Band	$K(j\omega)$	$R(j\omega)$
Plant	$\lvert\omega\rvert < 0.29$	1	$1/\lvert P(j\omega)\rvert$
Tracking	$0.29 < \lvert\omega\rvert < 1$	1	0.3
g-p margin	$1 < \lvert\omega\rvert < 2$	1	2.
Bandwidth	$2 < \lvert\omega\rvert < 2.41$	0	0.707
Roll off	$\lvert\omega\rvert > 2.41$	0	$4.\lvert P(j\omega)\rvert$

Table 3.2

The format of Table 3.2 is convenient for representing constraints in a way that is easy to read, thus useful in similar problems. Note that in this example we could write the constraints as a single disk inequality. This was possible here because overlapping constraints can be written as a disk constraint again. We will see in Chapters 4 and 5 that this is not always the case. That is, sometimes one cannot define a radius and center functions that incorporate all the information contained in the requirements.

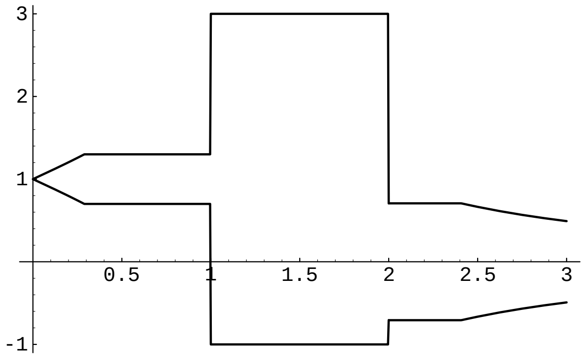


Figure 3.14: Section of the envelope defined by the center and radius functions. The horizontal axis is frequency ω, and the two functions are $K(j\omega) + R(j\omega)$ and $K(j\omega) - R(j\omega)$.

3.6 References and further reading

[DO] Dorf, R. C. *Modern Control Systems*, Addison-Wesley, 1981.

[O] Ogata, K. *Modern Control Engineering*, Prentice Hall, 1970.

[SS] Saucedo, R. and Schering, E. E. *Introductions to Continuous and Digital Control Systems*, MacMillan, 1968.

[FPE] Franklin, G., et al., *Feedback Control of Dynamic Systems*, Addison Wesley, 1986.

[H] Helton, J. W. Worst case analysis in the frequency domain; on H^∞ approach to control, IEEE Trans. on Automatic Control, **AC-30**, 1154–1170. December 1985.

[HS] Helton, J. W. and Schwartz, D. A Primer on the H^∞ Disk Method in Frequency Domain Design Control. Documentation for Fortran software, developed at Lab. for Math. and Statistics, UCSD. 1985.

[V] Vidyasagar, M. *Control Systems Synthesis: a Factorization Approach*, The MIT Press, 1985.

[D] John C. Doyle. CDC83.

[F] Francis, Bruce *A Course in H^∞ Control*, Springer Verlag, 1986.

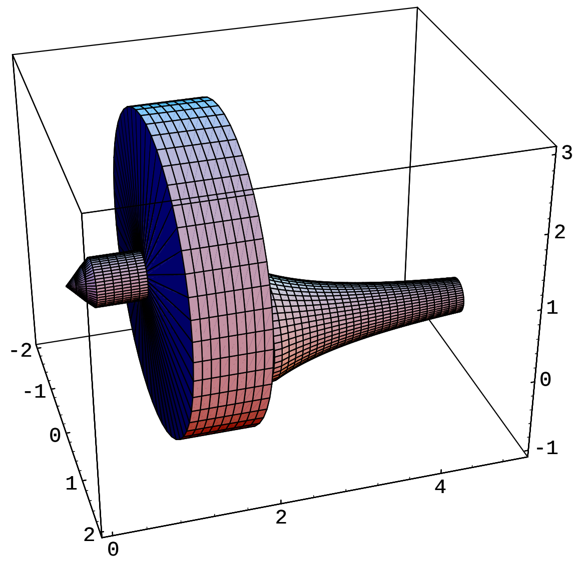


Figure 3.15: Envelope defined by the constraints. Frequency ω is on the axis that extends from left to right in the picture.

[BB] Boyd, Stephen P., and Barrat, Craig H. *Linear controller Design: Limits of Performance*, Prentice Hall, 1991.

[DG1] Desoer, C. A., and Gustafson, C. L. Design of Multivariable feedback system with simple unstable plant, (Berkeley ERL Memorandum No. M82/60).

[DG2] Desoer, C. A., and Gustafson, C. L. Controller design for linear multivariable feedback systems with stable plants, using optimization with inequality constraints, Int. J. Contr.

[Y] Young, N. An Introduction to Hilbert Space.

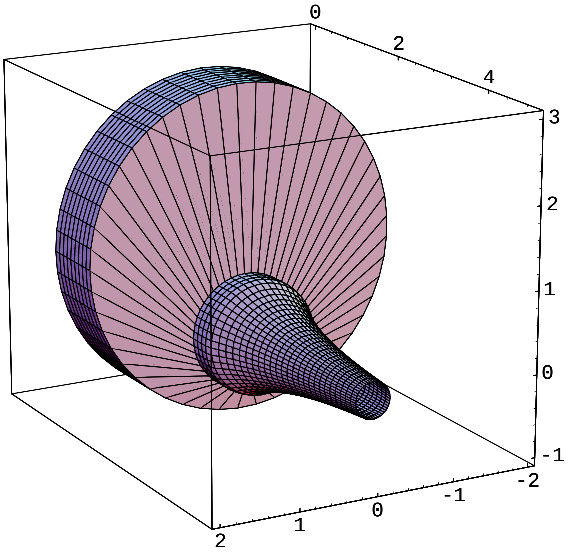


Figure 3.16: Envelope defined by the constraints, as seen from another vantage point.

Chapter 4

Optimization

In this chapter we discuss performance functions and how to use them as objective functions in optimization problems and . These concepts are applied to system design in examples. We begin with a review. A reader who is interested in getting quickly to a point where examples can be solved by computer should read sections 4.1 — 4.4 and then skip to Chapter 5.

4.1 Review of concepts

Consider the closed loop system $\mathcal{S}$ depicted in Fig.4.1.

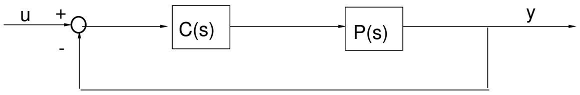


Figure 4.1: The closed loop system $\mathcal{S}$.

Here $P(s)$ and $C(s)$ are proper real rational functions of s. The function $P(s)$ is given, and the function $C(s)$ depends on the choice of designable transfer function $T(s)$. The functions $T(s)$ and $C(s)$ are related by the formula

$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}$$

or equivalently,

$$P(s)C(s) = \frac{T(s)}{1 - T(s)}.$$

The system $\mathcal{S}$ is internally stable if T has no RHP poles, and there is no RHP pole-zero cancellation in the product PC.

Performance requirements are given inequalities to be satisfied by functions associated with the system $\mathcal{S}$. The inequalities may be stated in terms of time or frequency. Frequency domain performance requirements were dealt with in Chapter 3. Time domain performance requirements are briefly discussed in Chapter 5 and treated in more detail in Chapter 6.

The basic control problem we consider is

Design

Given a plant P, a set of performance requirements $\mathcal{P}$ and a set of feasible functions

$$\mathcal{I} = \{T \in \mathcal{RH}^\infty \ : \ T \text{ is the closed loop of an internally stable system } \mathcal{S} \text{ with plant } P\}$$

determine if there exists $T \in \mathcal{I}$ satisfies all the performance requirements in $\mathcal{P}$. If the answer is yes, find one of such T.

The method used to solve *Design* is as follows:

I. Obtain a mathematical description of the performance and internal stability requirements (write down disk inequalities, take notice of the unstable poles and zeros of the plant).

II. Obtain a performance function (e.g., derive a formula for the radius and center functions).

III. Find a feasible T that yields the best performance.

Often in practice steps I – III above are followed by step IV below.

IV. If many satisfactory T exist or none exist, tighten or loosen the specifications accordingly and go to step I. Stop if this process can not be carried further.

In this chapter we will emphasize steps II and III, especially when the performance function is of the "circular" type (for a definition see Subsection 4.2). Step III requires the use of computer software, since numerical calculations are involved [1]. Step I was treated in Chapters 2 and 3, and step IV is illustrated in the problems solved in Chapter 5. See Chapter 5 for a discussion of performance functions that are not circular.

---

[1] The examples in this chapter were solved with the software package OTPDesign.

4.2 Generating a performance function

Let us suppose the we have one or more performance requirements, that can be expressed as disk inequalities. We consider in this section the case when all of these can be put together as a single disk inequality, valid over all frequencies:

$$|K(j\omega) - T(j\omega)| \leq R(j\omega) \qquad \forall\omega \tag{4.1}$$

We will build a "performance function" from (4.1) that can be used to solve design problems. Begin by rewriting (4.1) in the following way:

$$\frac{1}{R(j\omega)^2}|k(j\omega) - T(j\omega)|^2 \leq 1 \; \forall\omega \tag{4.2}$$

Now for a given T calculate the largest value of the left-hand-side of (4.2):

$$\gamma(T) = \sup_{\omega} \frac{1}{R(j\omega)^2}|k(j\omega) - T(j\omega)|^2 \tag{4.3}$$

If a particular T is available, then one can check if the original performance requirements are satisfied. To do it one just calculates $\gamma(T)$. It is not hard for the reader to convince himself that two possible outcomes of calculating $\gamma(T)$ are:

$$\begin{array}{lll} \gamma(T) \leq 1 & \Rightarrow & T \text{ satisfies (4.1)} \\ \gamma(T) > 1 & \Rightarrow & T \text{ does not satisfy (4.1)} \end{array} \tag{4.4}$$

Example. Suppose that the center k and radius r are given by

$$\begin{array}{rcl} k(\omega) & = & -1/2\frac{j\omega}{(j\omega-1)^2} \\ r(\omega) & = & \frac{2}{(\omega^2+1)^{1/2}} \end{array} \tag{4.5}$$

We now check if

$$T(j\omega) = \frac{1}{j\omega + 1}$$

satisfies the disk inequality (4.1):

$$\begin{array}{rcl} g(\omega) & := & \frac{1}{r(j\omega)^2}|k(j\omega) - T(j\omega)|^2 \\ & = & \sup_\omega \frac{(\omega^2+1)}{4}| - 1/2\frac{j\omega}{(j\omega-1)^2} - \frac{1}{(\omega^2+1)^{1/2}}|^2 \\ & = & \frac{4+21\omega^2+9\omega^4}{16(1+\omega^2)^2} \end{array} \tag{4.6}$$

After some calculations one obtains

$$\frac{dg}{d\omega} = \frac{13\omega - 3\omega^3}{8(1+\omega^2)^3} \tag{4.7}$$

Thus $g(\omega)$ has three critical points $\omega = 0$, $\omega = \pm\sqrt{13}/3$. The value $g(0) = 1/4$ is a minimum, and $g(\pm\sqrt{13}/3) = 297/512$ is a maximum. Also, $g(\omega) \to 9/16$ as $\omega \to \infty$. A plot is most informative (see Fig. 4.2). Thus

$$\gamma(T) = \sup_{\omega} g(\omega) = \frac{297}{512} < 1$$

We conclude that T does satisfy the disk inequality (4.1).

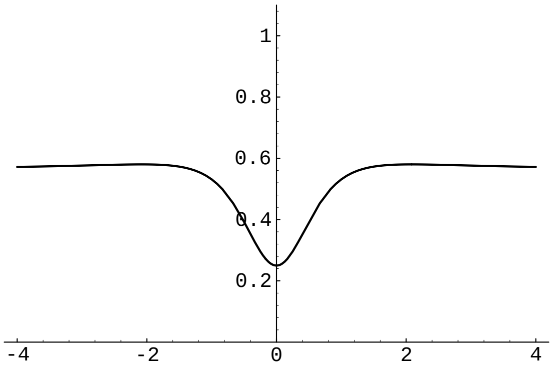


Figure 4.2: Plot of the function $g(\omega) = 1/r(\omega)^2|k(j\omega) - T(j\omega)|^2$.

For designable T and given center and radius functions k and r respectively, the expression

$$\frac{1}{R(j\omega)^2}|k(j\omega) - T(j\omega)|^2 \tag{4.8}$$

is called *circular performance function*, or simply *performance function*. The number $\gamma(T)$ is the *worst case performance of* T, but most of the time we will simply call it *performance of* T.

4.3 Finding T with best performance

A feasible function T^* with the property that

$$\gamma(T^*) \le \gamma(T) \qquad \forall T \text{ feasible} \tag{4.9}$$

is called *optimal.* We reserve the symbol γ^* for the "optimal performance", that is, the performance of the optimal designable transfer function:

$$\gamma^* = \gamma(T^*) \tag{4.10}$$

The optimal designable function T^* is a precious object since there is no other T that has better performance. By calculating the best performance possible γ^*, the designer has conquered a peak. This vantage point offers information that is cruxial in making decisions on the next steps. If T^* satisfies the constraints one can use T^* as the choice of design, or modify T to get a low order model that still satifies the constraints. If T^* does not satisfy the constraints, then is immediately known that there is no T that satisfies the constraints! The designer then can take other actions such as to modify the "soft" constraints and then do the calculation again.

The process of finding the optimal T^*, and the corresponding optimal performance γ^* is called *optimization.* The optimal T minimizes $\gamma(T)$ over all possible T (recall that this includes internal stability). More precisely, we say that T^* is a solution to

$$\gamma^* = \inf_T \sup_\omega \frac{1}{R(j\omega)^2}|k(j\omega) - T(j\omega)|^2 \tag{4.11}$$

Because the optimization is a complicated procedure, to carry it out one needs a computer with the appropiate software. When computer algorithms and software are not available to do the optimization, one has to resort to unsatisfactory approaches, such as writing a formula for T as a rational functions with given order and unknown coefficients. Then one tries to do the get a good T by simply trying lots of possible choices of coefficients. These kind of methods are not always practical, and more importantly, the designer usually gets no indication of how far from optimal the choice of T is.

4.3.1 Example

Physically the problem is to stabilize the plant and impose closed loop gain-phase margin and roll-off performance.

Problem: Solve *Design* for the plant $P(s) = (s+5)/(s+1)^2$, with performance requirements gain-phase margin $m = 0.5$ and closed loop roll-off $|T(j\omega)| < |P(j\omega)|$, for $1 < \omega$.

Solution. Observe that the plant is strictly proper and stable. At this point we will not spend more time discussing internal stability, since the software we intend to use to do the optimization takes care of this automatically.

We first construct the center and radius functions for our problem. Recall that the gain-phase margin constant m is the closest acceptable distance from the function $L = PC$ to the point -1, or in other words

$$|1 - T(j\omega)| < \frac{1}{0.5} = 2, \text{ for all } \omega$$

The requirement on the rolloff can be written as

$$|T(j\omega)| < 0.707|P(j\omega)|, \text{ for } 1 < \omega$$

Thus the frequency domain performance requirements are described by (see Fig. 4.3):

$$\begin{array}{rcll} |1 - T(j\omega)| & < & 2, & \text{for } 0 \le \omega \le 1 \\ |T(j\omega)| & < & 0.707|P(j\omega)|, & \text{for } 1 < \omega \end{array} \tag{4.12}$$

The inequalities in (4.12) can be cast as a single disk inequality

$$|K(j\omega) - T(j\omega)| < R(j\omega)$$

where

$$K(j\omega) = \begin{cases} 1 & \text{if } |\omega| < 1 \\ 0 & \text{if } |\omega| \ge 1 \end{cases} \tag{4.13}$$

and

$$R(j\omega) = \begin{cases} 2.0 & \text{if } 0 \le |\omega| < 1 \\ 0.707|P(j\omega)| & \text{if } 1 < |\omega| \end{cases} \tag{4.14}$$

The optimal T^* is the solution to

$$\gamma^* = \inf_{T \in \mathcal{I}} \sup_{\omega} \frac{1}{R(j\omega)^2} |K(j\omega) - T(j\omega)|^2 \tag{4.15}$$

To give an idea of what the answer is, some results of calculations with the software package OPTDesign are presented below. Numerical issues have not been considered here, so the reader interested in those should read Part II of this book.

Calculations by computer give $\gamma^* = 0.09923$, so there exist solutions to *Design*. Here is a rational function that is a low order approximation to the optimal one:

$$T_0(s) = \frac{5 - s}{(s+1)^2} \frac{(0.0311 + 0.2613\, s)}{(0.8809 + s)} \tag{4.16}$$

See Figures 4.4 – 4.7 for plots of the optimal T^*.

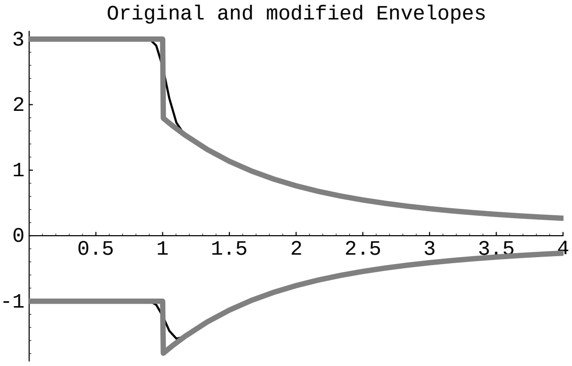


Figure 4.3: The envelope defined by center and radius functions in in Example 1 is given by the thick gray lines. The thin black line the result of automatic smoothing, a modification that is performed to improve the numerics on the computer. The horizontal axis is frequency ω, and the two functions are $K(j\omega) + R(j\omega)$ and $K(j\omega) - R(j\omega)$.

4.4 Good performance functions

There are two basic rules that have to be followed to define a useful performance function. These are:

Rule 1. The performance at any T and any frequency ω (including $\omega = \infty$) should be a well defined real number[2].

Rule 2. For any fixed frequency ω, the performance as a function of T cannot be independent of T.

The justification of rules 1 and 2 requires the results in Part III of this book, so we refer the interested reader to it.

The following are examples of performance functions that are common in the engineering literature. We assume that the underlying design problem has internal stability requirements, and that the plant P is strictly proper, and has no poles or zeros on the $j\omega$ axis. Also, W, W_1, W_2 denote given rational weight functions of frequency.

1. $\Gamma(\omega, T(j\omega)) = W(j\omega)|T(j\omega)|$. If the control problem has a plant P with relative degree p, then W must have relative degree $-p$ (or larger) for the

---

[2] The performance may turn out to be infinitely large at frequencies ω_0 such that the plant has a zero at $s = j\omega_0$, or at infinite frequency. This should not be a problem, provided only T that give internally stable systems are considered.

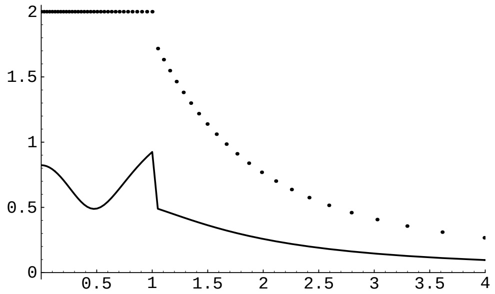


Figure 4.4: The dots represent the radius function $R(j\omega)$. The solid line represents the magnitude of the error function $|K(j\omega)-T_0(j\omega)|$. Frequency ω in on the horizontal axis. Clearly T satisfies the constraints.

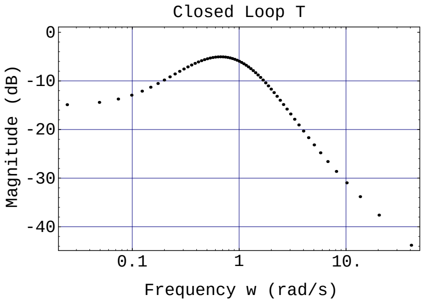


Figure 4.5: Magnitude of T_0.

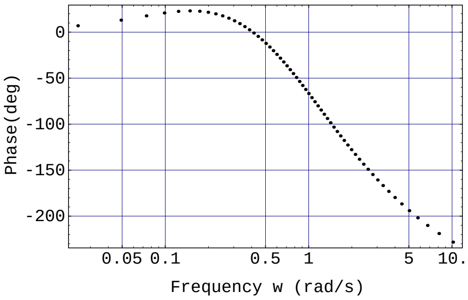


Figure 4.6: Phase of T_0.

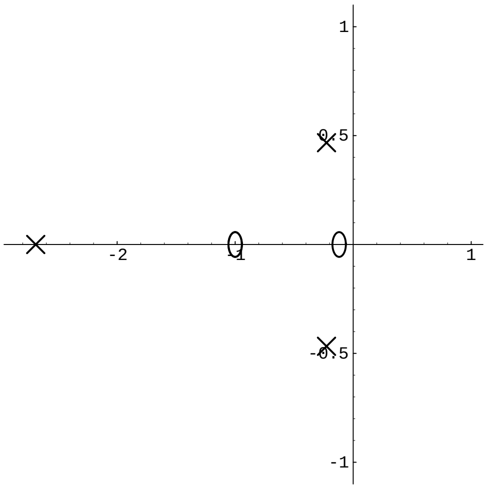


Figure 4.7: The zeros and poles of the compensator $C = (-0.0310 - 0.3235s - 0.5537s^2 - 0.2613s^3)/(0.8808 + s)$ that corresponds to T_0 are indicated by "o" and "x" respectively. Note that this compensator is stable. Stability of the compensator is not guaranteed a priori by the design method of this book.

performance to be well defined. If W does not 2 is violated at $\omega = \infty$.

2. $\Gamma(\omega, T(j\omega)) = W(j\omega)|1 - T(j\omega)|$. This is not a good performance, no matter what W is. The reason is that since T rolls off at high frequency, the performance at very high frequency is determined by W only. It has no way measuring how fast T rolls off or any relevant high frequency property of T.

3. $\Gamma(\omega, T(j\omega)) = W_1(j\omega)|T(j\omega)| + W_2(j\omega)|1 - T(j\omega)|$. The weight W_2 should be 0 at infinite frequency, so that the term with this weight does not influence performance at very high frequency. The comment in the first example also applies to the weight W_1.

4. $\Gamma(\omega, T(j\omega)) = W_1(j\omega)|T(j\omega)|^2 + W_2(j\omega)|1 - T(j\omega)|^2$. This continues example 3. The relative degree of W_2 should be $-p/2$ or larger, where p is the relative degree of the plant[3].

4.5 Performance not of circular type

In this section we discuss performance functions more general than (4.8). To motivate the discussion suppose that we have a design problem where one of the constraints is

$$W_1(\omega)|T(j\omega)| + W_2(\omega)|1 - T(j\omega)| < 1, \quad \omega_1 \le \omega \le \omega_2 \tag{4.17}$$

where W_1 and W_2 are known weight functions[4]. Clearly this constraint is not a disk constraint. Indeed, the region

$$S_\omega(c) = \{z : W_1(\omega)|z| + W_2(\omega)|1 - z| \le c\}$$

is not a disk. However, the inequality (4.17) can be used to define a performance function, valid at least for those ω satisfying $\omega_1 \le \omega \le \omega_2$. For this we set, for z any complex number,

$$\Gamma(\omega, z) = W_1(\omega)|z| + W_2(\omega)|1 - z|$$

Thus a designable transfer function T satisfies the constraint at frequency ω if and only if

$$\Gamma(\omega, T(j\omega)) \le 1 \tag{4.18}$$

---

[3] When zeros or poles of the plant occur on the $j\omega$ axis, weight functions have to be chosen so that the weights in the terms $W_1(j\omega)|T(j\omega)|^2$ and $W_2(j\omega)|1 - T(j\omega)|^2$ have the zeros (resp. poles) at the same values of ω than those of the plant, order included. The reason for this is the internal stability requirement. See Chapter 8

[4] This may arise when considering competing constraints, see Section 6.5 in Chapter 6.

The function Γ is called a *performance function*. Let us define for $T \in \mathcal{RH}^\infty$ the number

$$\gamma(T) := \sup_{\omega} \Gamma(\omega, T(j\omega)) \tag{4.19}$$

The number $\gamma(T)$ represents the cost, or overall performance, associated with the designable transfer function T. It is called the *performance index*. Note that (4.18) can be written as

$$\gamma(T) \leq 1 \tag{4.20}$$

Observe that $\gamma(T) < 1$ means that T satisfies the performance requirements, with some slack. Thus we associate small values of $\gamma(T)$ with *good performance*.

The simplest functions Γ arise when a single disk inequality applies at each frequency;

4.6 Optimization

It is a fact that there is a formula that depends on the RHP poles and zeros of the given plant P which gives all $T \in \mathcal{I}$. Indeed we have that there exist functions $A,\ B \in RH^\infty$ such that

$$T \in \mathcal{I} \iff T = A + BT_1, \text{ for some } T_1 \in RH^\infty \tag{4.21}$$

For general P such formula is proved in Chapter 8, while for RHP-stable P we already saw in Chapter 2 that

$$T \in \mathcal{I} \iff T = PT_1, \text{ for some } T_1 \in RH^\infty \tag{4.22}$$

One approach to treating the design problem in practice is to use formulas (4.21) and (4.22) to account for internal stability. Depending on the specific approach and the tools available, to solve *Design* the formulas are either manipulated directly by the designer, or they are automatically handled by the computer. In this case only the list of unstable zeros and poles of the plant is necessary for the software to build and use the necessary formulas.

4.6.1 The optimization problem $OPT_{\mathcal{I}}$

Suppose that steps I and II have been completed, that is, there is available a performance function Γ and a set $\mathcal{I}$ of admissible functions. To find the "best" T possible, one must find the smallest possible value for the performance index $\gamma(T)$ for all $T \in \mathcal{I}$. That is, one must solve the following problem:

$OPT_{\mathcal{I}}$

Given $\Gamma(\omega, z)$ a positive valued function of $\omega \in R$ and of $z \in C$, and a set $\mathcal{I}$ of admissible functions, find

$$\gamma^* := \inf_{T \in \mathcal{I}} \sup_{\omega} \Gamma(\omega, T(j\omega)) \tag{4.23}$$

and find an optimal T^* if it exists.

In general $OPT_{\mathcal{I}}$ is a constrained problem, since one may have $\mathcal{I} \neq RH^{\infty}$. By solving $OPT_{\mathcal{I}}$ we solve *Design.* This follows from Proposition 3 below.

Proposition 3 *If* $\gamma^* < 1$ *then there exist solutions to Design, and if* $\gamma^* > 1$ *then there are no solutions to Design.*

Proof. By definition, the number γ^* is the smallest value possible for γ so that the inequality

$$\Gamma(\omega, T(j\omega)) \leq \gamma \tag{4.24}$$

has solutions $T \in \mathcal{I}$. Therefore $\gamma^* > 1$ says that there are no functions Γ that satisfy (4.20) i.e., that there are no solutions to *Design.* Similarly, $\gamma^* < 1$ says that there is at least one function $T \in \mathcal{I}$ that satisfies (4.20), which implies that there exist solutions to *Design.* ■

4.7 Internal stability and optimization

It is a very simple matter to transform $OPT_{\mathcal{I}}$ into an unconstrained optimization problem OPT. This is desirable because powerful mathematical algorithms for solving unconstrained problems are available. It is so easy that even the computer can do this in a manner that is transparent to the user, for example, as implemented in the package OPTDesign. Ocasionally the designer may want to do it directly by hand (or be forced to do it). Thus our discussion of this below.

4.7.1 The optimization problem OPT

To do the advertised transformation of $OPT_{\mathcal{I}}$ into OPT we use the formula (4.21) in combination with the performance $\Gamma(\omega, z)$. Suppose T is as in (4.21). Define a new performance function as follows:

$$\Gamma_1(\omega, z_1) = \Gamma(\omega, A(j\omega) + B(j\omega)\, z_1\,) = \Gamma(\omega, z) \tag{4.25}$$

Therefore $OPT_{\mathcal{I}}$ is equivalent to the following unconstrained optimization problem:

Given $\Gamma_1(\omega, z)$ a positive valued function of $\omega \in R$ and of $z \in C$, find

OPT

$$\gamma_1^* := \inf_{T_1 \in \mathcal{I}} \sup_{\omega} \Gamma_1(\omega, T_1(j\omega)) \tag{4.26}$$

and find an optimal T_1^* if it exists.

4.7.2 *OPT* with circular Γ

Note that if the function Γ is circular, then the function Γ_1 in (4.25) is also circular. To see this, combine the disk inequality (??) with (??) to obtain the inequality

$$\frac{1}{R(j\omega)^2}|K(j\omega) - A(j\omega) - B(j\omega)T_1(j\omega)|^2 \leq 1 \quad (\forall\omega) \tag{4.27}$$

The resulting inequality is itself a disk inequality in the variable T_1:

$$\frac{|B(j\omega)|^2}{R(j\omega)^2}|\frac{K(j\omega) - A(j\omega)}{B(j\omega)} - T_1(j\omega)|^2 \leq 1 \quad (\forall\omega) \tag{4.28}$$

Thus we can choose $\Gamma_1(\omega, T_1(j\omega))$ to be the left-hand-side of inequality (4.27) to arrive at the unconstrained optimization problem *OPT*.

4.8 Exercises

4.9 References and further reading

Chapter 5

A Design Example with OPTDesign

5.1 Introduction

This chapter presents a computer session and introduces the reader to practical issues. This lets the student see how the ideas he has learned feed into a computer implementation. We use our own package OPTDesign, so this chapter can be utilized as an introduction to it.

5.1.1 The package OPTDesign

OPTDesign and supporting software are written in Mathematica, and they were produced at the Laboratory for Mathematics and Statistics at the University of California San Diego, La Jolla, California, from 1990 - 1994. By J. W. Helton, O. Merino, J. Myers and T. Walker. The development of OPTDesign and its companion the optimization program Anopt has been supported by the Air Force Office for Scientific Research and by the National Science Foundation.

5.1.2 How to use this notebook

This chapter actually is something called a Mathematica Notebook, which has been automatically converted to the standard scientific typesetting language called TeX in order to include it in this book. Notebooks have the advantage that they integrate text, computer input and computer output. The latter includes excellent graphics. The user may modify it and re-execute commands easily, and the new output replaces the old one automatically.

You can use this notebook in two ways:

1. As an illustrated guide to the use of OPTDesign package. A design example is treated here in depth to illustrate the basic commands and the interpretation of results.

2. As a template to treat many problems. Once you are using a Notebook it is easy to modify equations and parameters.

The notebook has been organized so that there are very few places where the user specifies the input. This has the tremendous advantage of making the set up of computer runs a very simple task.

The name OPTDesign denotes a collection of programs (or "package") for doing control design. OPTDesign is also the name of the main optimization program in the package. To avoid confusion we will denote the program as OPTDesign[].

5.1.3 Starting a session

Begin by loading OPTDesign into a Mathematica session. Computer input and output below is in "typewriter type" (darker) style. The input has regular size (this is what you can modify), while the output appears in smaller type. The rest of the text are comments.

```
<<OPTDesign`
```

```
 Anopt 3.0 (c) Copyright 1991-94
 J.W.Helton and O.Merino.  All rights reserved.
OPTDesign 1.0 (c) Copyright 1991-94
    J.W. Helton and O.Merino.  All rights reserved.
```

5.2 The problem

The design problem is, given a plant P and a set of performance requirement, find a system (S) that is internally stable and that satisfies the performance requirements.

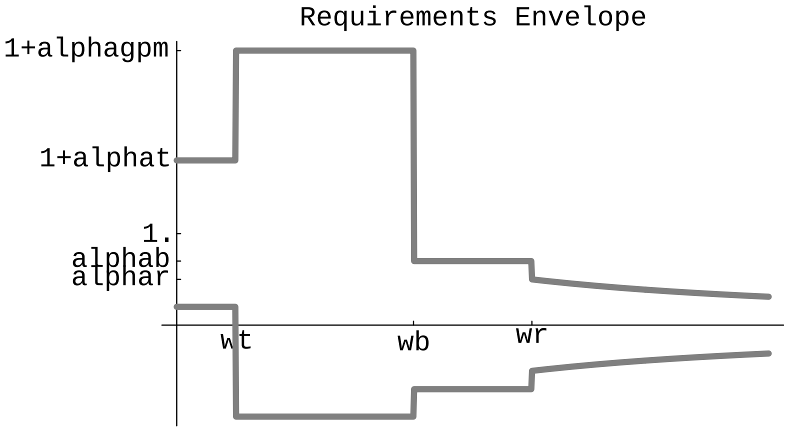
We now give the data for a specific problem. The plant is

$$P(s) = \frac{(1 - s/5)}{(1 + s/2)^2}$$

and the requirements are, tracking on the band $0 \leq \omega \leq 0.3$ with bound 1.5, gain-phase margin constant of 0.5, bandwidth $0 < \omega \leq 2.$, and closed loop roll off with bound 2.5 on the band $4.0 \leq \omega$.

A list of the requirements in this problem in the form of inequalities is the following:

Tracking	$\lvert T(j\omega)-1\rvert$	$<$	α_t	$0<\omega<\omega_t$
Gain-Phase	$\lvert T(j\omega)-1\rvert$	$<$	α_{gpm}	$\omega_t<\omega<\omega_b$
Bandwidth	$\lvert T(j\omega)\rvert$	$<$	α_b	$\omega_b<\omega<\omega_r$
Roll-Off	$\lvert T(j\omega)\rvert$	$<$	$\alpha_r \lvert P(j\omega)\rvert$	$\omega_r<\omega$



Thus we see that in our problem, $\omega_t = 0.3$, $\alpha_t = 1.5$, $\omega_b = 2.0$, $\alpha_b = 0.7$, $\omega_r = 4.0$, $\alpha_r = 2.5$.

5.2.1 Specifying the problem.

The plant and a set of design requirements are specified now. *Frequently this will be the only place where you enter data.* The plant should be entered as a rational function:

```
p[s_]= ( 1 - s/5 )/( 1 + s/2 )^2;
```

We set the values of important constants below:

```
     wt  = 0.3;
 alphat  = 1.5;

alphagpm = 2.0;

      wb = 2.0;
```

```
alphab = 0.7;

    wr = 4.0;
alphar = 2.5;
```

5.2.2 The requirements envelope

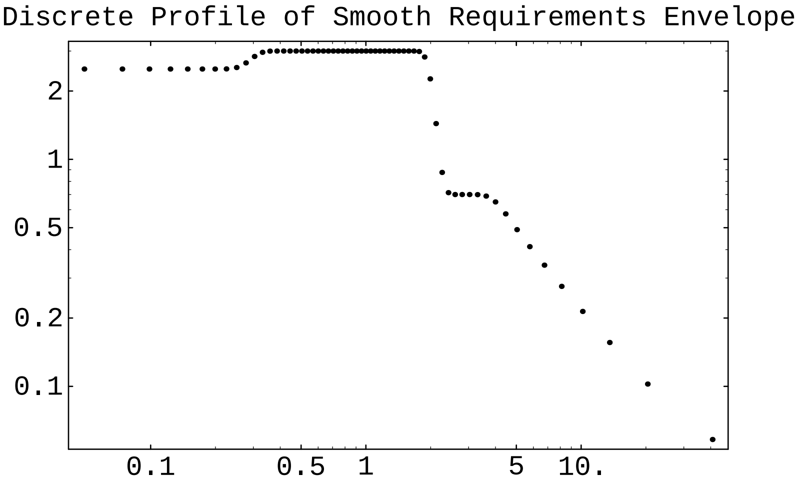
Now we define a center and radius functions, which determine the envelope.

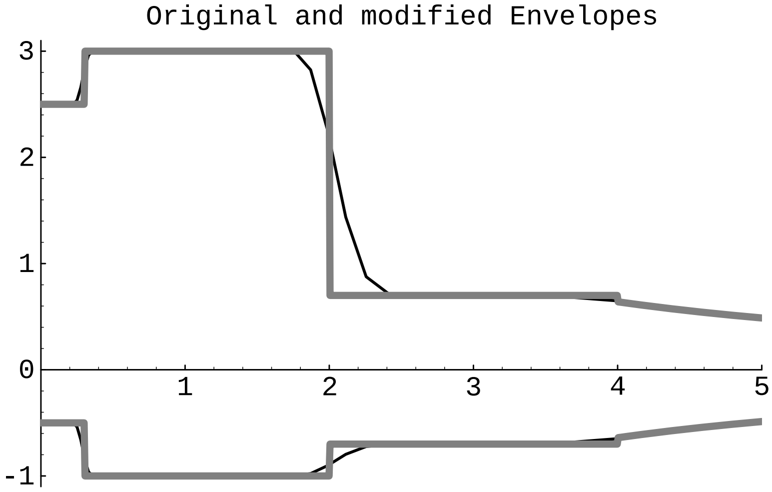
```
k0[w_]= Which[ Abs[w] < wb, 1.0,
                     True, 0.0];

r0[w_] = Which[Abs[w] < wt,  alphat,
               Abs[w] < wb,  alphagpm,
               Abs[w] < wr,  alphab,
                     True ,  alphar*Abs[p[I w]]
                ];
```

The requirements envelope has jump discontinuities at two locations on the $\omega > 0$ semiaxis. For better results, smoothing of functions is convenient in this case. OPTDesign does a very small amount of smoothing automatically; we will rely on it for now.

The function EnvelopePlot[ ] produces two plots. The first is a plot of the discrete upper limit of the requirements envelope. The second figure has a plot of the original envelope and of the envelope resulting after smoothing.

```
EnvelopePlot[Radius->r0,Center->k0,FrequencyBand->{0.,wr+1.}]
```


Discrete Profile of Smooth Requirements Envelope


Original and modified Envelopes

```
-Graphics-
```

The plot of the upper limit of the profile of the requirements envelope can be used to judge if there are enough frequency gridpoints in the band of interest. This is of course relative to the precision you want to attain (to get a feel for this several examples must be run) . In our example we have that most of the gridpoints are in the band $0.1 < \omega < 10.0$, which is where the center and radius functions have their distinctive features. This is good.

The plot of the requirements envelopes before and after smoothing gives an idea of how much distortion we are introducing into the problem by doing smoothing of functions. We must keep in mind that the solution obtained with OPTDesign will be optimal with respect to the "modified" envelope.

If we do not like either one of the two plots, then several parameters can be modified to obtain more satisfactory plots. We leave this for a later subsection, and proceed now with a run of the program OPTDesign[] with the given data.

5.3 OPTDesign iteration

The program OPTDesign[] is now run with a minimal set of explicit input. Other input that can be specified to modify default values will be discussed later.

Running OPTDesign[] produces information on the progress of the run that is printed on the screen.

```
OPTDesign[p,Radius->r0,Center->k0];
```

```
Parameterization for internal stability:
  T = A + B * T1, where T1 is RHP stable and
      -5 + s
A[s]=---------
             2
     (2. + s)
      -5 + s
B[s]=---------
             2
     (2. + s)


Processing performance function...
Sampling Radius and Center ...
Smoothing Radius and Center ...
Optimization routine Anopt now number-crunching ...

It :    Current Value    :  Step   : Optimality Tests  :  Error  : Sm. : Grid
   :      SupGamma       :         :   OT1       OT2   :   NED   :     :
-----------------------------------------------------------------------------
 0 : 2.2485924730083E+00 :   N/A   : 8.9E-01 : 1. E+00 :   N/A   : NON : 128
 1 : 1.5969785364195E-01 : 1.8E+00 : 3.6E-03 : 0  E+00 : 1.5E-02 : NON : 128


Summary
-------
Supremum of gamma: SupGamma = 1.596978536419496E-01
Optimality Test  : OT1      = 3.5999393585E-03
Error diagnostic : NED      = 1.54E-02
```

```
Calculating output functions ...
Resetting options ...
Done!
```

The output parameter `SupGamma` is less than 1. In a disk - type problem this means that there is a solution that satisfies the (smooth) constraints. The figure NED is a measure for numerical error or noise, and OT1, OT2 are diagnostics for optimality: OT near 0 means the calculated solution is nearly optimal. We say more on this later.

Another important output is a discrete set of values of the optimal closed loop, which can be recalled with the symbol `T` .

Conclusion

The conclusion we draw with respect to the OPTDesign run above is that there are solutions to the modified Design problem obtained after the envelope was smoothed, and that accuracy of the output is acceptable.

5.3.1 Generating a rational closed loop

A low order rational representation of the closed loop transfer function T can be generated after an OPTDesign[] run with the function RationalModel[]. By default it finds a rational that approximates T.

```
Trat = RationalModel[]
```

```
Error = 0.171731

 -5 + s     (-4.67853 - 1.92117 s) (-5 + s)
--------- + -------------------------------
        2              2
(2. + s)        (2. + s)  (3.769 + 1. s)
```

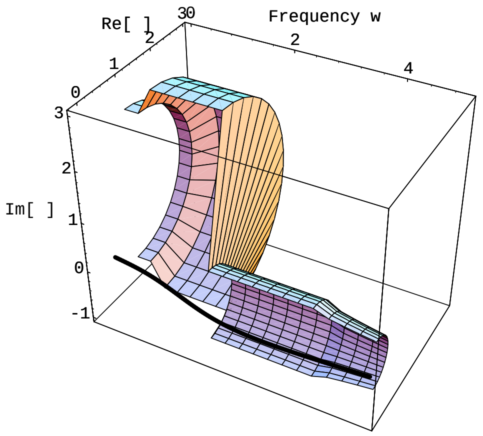
RationalModel[] produces a rational of the form

$$T_{rat}(s) = A(s) + B(s)T_{1rat}(s)$$

The proper rational functions A, B depend on the plant and incorporate automatically the internal stability requirements on T. The rational T_{1rat} is produced by the optimization procedure. If you run OPTDesign[] first, you do not have to specify A, B or $T1$ as an input to `RationalModel[ ]`.

The Error figure above refers to the fit of $T1$ and not to the fit of T. Usually we can get information from a plot. With the command shown below we obtain a 3D picture of the envelope, together with the plot of our choice of rational closed loop.

```
EnvelopePlot3D[Radius->r0,Center->k0,ClosedLoop->Trat]
```



```
-Graphics3D-
```

5.4 Producing a rational compensator

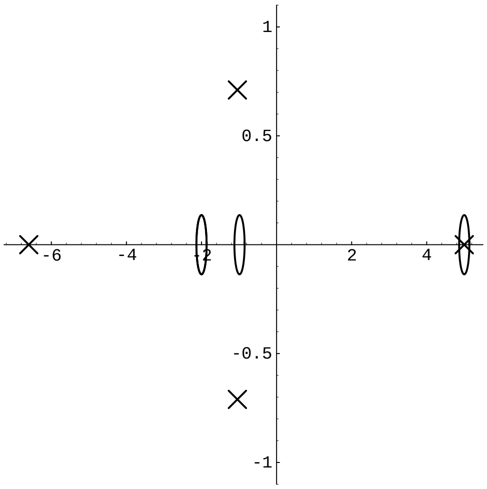
We obtain the compensator from the formula $C = (1/P) * T/(1 - T)$:

```
comp1 = Together[ 1/p[s] Trat/(1-Trat) ]
```

```
                 2                             2
  1.15147 (2 + s)  (-4.93682 - 4.01264 s + 1. s )
---------------------------------------------------
                                         2       3
(-5 + s) (10.5283 + 15.3797 s + 8.69017 s  + 1. s )
```

The compensator seems to have a pole at s = 5, but this shouldn't be the case since OPTDesign guarantees an internally stable system. The zeros and poles of `comp1` are shown below:

```
PlotZP[comp1]
```



```
-Graphics-
```

The figure shows that comp1 has a pole - zero pair at s=5. We verify this now:

```
comp1zeros = s /. NSolve[ Numerator[comp1]==0,s]
```

```
{-2., -2., -0.987364, 5.}
```

```
comp1poles = s /. NSolve[ Denominator[comp1]==0,s]
```

```
{5, -6.60224, -1.04396 - 0.710493 I, -1.04396 + 0.710493 I}
```

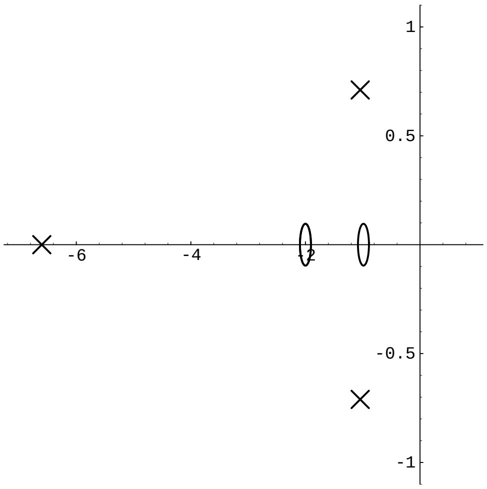
The standard Mathematica functions do not cancel terms with decimal notation. To do it we can use a function provided with OPTDesign:

```
comp2 = Cancel2[comp1,s - 5,s]
```

```
                               2            3
4.54766 + 9.15353 s + 5.74278 s  + 1.15147 s
---------------------------------------------
                                 2       3
  10.5283 + 15.3797 s + 8.69017 s  + 1. s
```

Hence `comp2` results from the simplification of `comp1`.

```
PlotZP[comp2]
```



```
-Graphics-
```

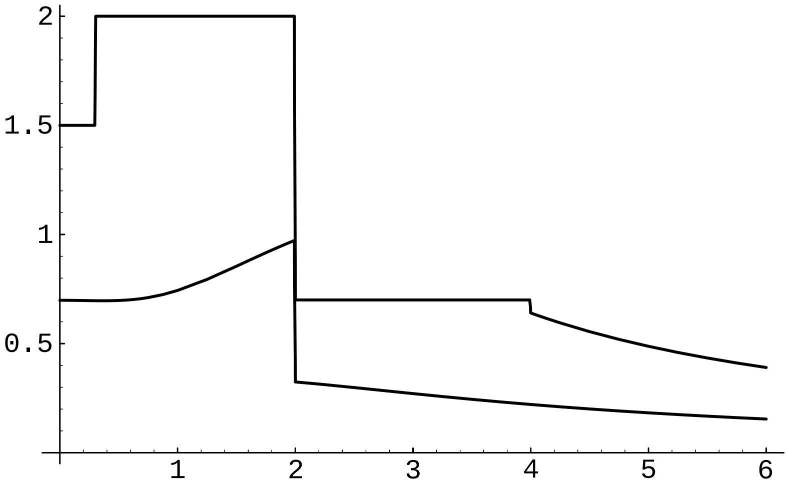
5.5 How good is the answer that we obtained?

OPTDesign smooths out the envelope automatically, thus distorts it a bit near points of sharp changes. Another source of error is the rational approximation to the answer given by OPTDesign[ ]. An important question is, do we obtain in the end a closed loop or a compensator that solve the original problem?

It is clear from the plots obtained above that the compensator is proper and does not cancel RHP poles or zeros of the plant. Also Trat is stable, thus for this choice of compensator the system is internally stable.

To verify that we have met the performance requirements we can do it in the following manner:

```
closedlp[w_] = ( Trat /. s-> I w );

Show[ Plot[Abs[k0[w]-closedlp[w]],{w,0,6},
                DisplayFunction->Identity],
      Plot[r0[w],{w,0,6},
                DisplayFunction->Identity],
     DisplayFunction->$DisplayFunction]
```



```
-Graphics-
```

5.5.1 Plots of the calculated closed loop function

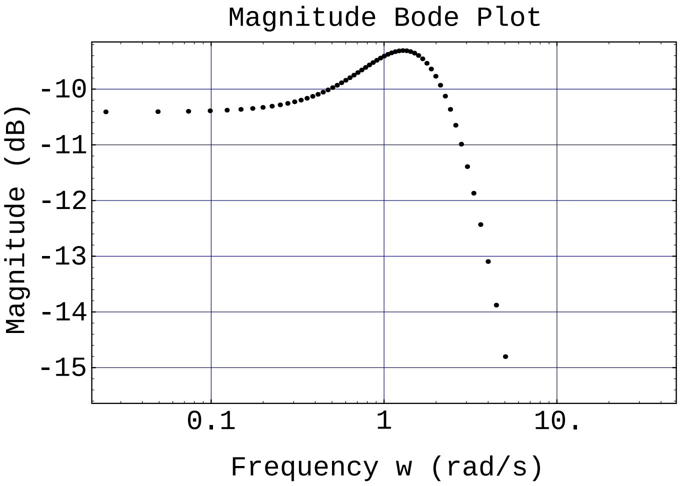
Plots can be produced with the objective of evaluating the choice of grid-size and smoothing input parameters versus the results of the run and the frequency bands of interest. If the results are not satisfactory the problem has to be re-run with modified input parameters.

First the rational closed loop is sampled on the grid used in the calculations above.

```
Tdisc = Discretize[closedlp];
```

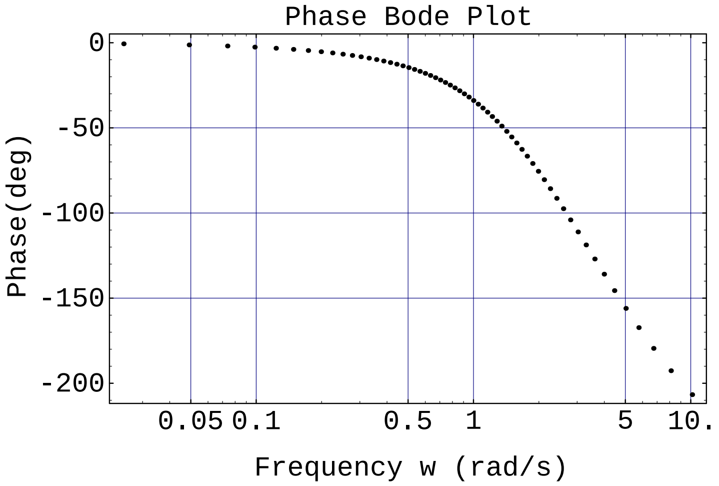
The Bode plots are generated as follows.

```
bm = BodeMagnitude[Tdisc,PlotLabel->"Magnitude Bode Plot"];
```



One can see in the magnitude plot of the closed loop that most of the sample points are located in the frequency band $0.1 < \omega < 10.$, which we judge (bassed on the fact that both the input functions k0 and r0 change mostly in this band) to be acceptable for the design problem we are considering. We will see in a later Section how to give input parameters to get enough sample points in the frequency bands of our choice. Here is a plot of the phase of T:

```
BodePhase[Tdisc, PlotLabel->"Phase Bode Plot"]
```



```
-Graphics-
```

Plots for the open loop L = p c can be produced in similar fashion.

5.6 Reducing the error

The rational approximation entails an error, as does the smoothing of the envelope of constraints by OPTDesign. Re-running OPTDesign with modified grid or with more smoothing may be necessary in some cases to improve the answer.

There are two input parameters that specify the grid. One parameter is the number of gridpoints, specified with the option $Ngrid \rightarrow nn$. Another parameter is a scaling, which is specified as $GridSpread \rightarrow nn$. Here are the values used in the run above:

```
{Ngrid, GridSpread} /. Options[OPTDesign]
```

```
{128, 1.}
```

Therefore 128 gridpoints were used, and most of the points are located near the frequency $\omega = 1$, as the Bode plots above indicate.

Another input parameter is the amount of smoothing, specified as $Nsmth \rightarrow nn$. We used above

```
Nsmth /. Options[OPTDesign]
```

```
1
```

The following is an example of what you would type to re-run Anopt, with more gridpoints, more smoothing and gridpoints accumulating near $\omega = 5$.

```
OPTDesign[p,Center->k0,Radius->r0,
            Ngrid->256,
            Nsmth->5,
            GridSpread->5
           ];
```

5.6.1 Improving the fit

The order of the rational model of T can be increased to obtain a better fit:

```
Trat2 = RationalModel[DegreeOfDenominator->5]
```

```
Error = 0.122737

                                                      2            3
 -5 + s     (-5 + s) (-4.80837 - 4.65535 s - 5.22911 s  - 1.87092 s )
--------- + ---------------------------------------------------------
        2             2                                 2       3
(2. + s)      (2. + s)  (3.74005 + 3.45048 s + 3.89256 s  + 1. s )
```

One sees that the improvement is small in this example.

There are a couple of things to keep in mind when using `RationalModel`: 1. The routine RationalModel[] may fail to work satisfactorily when (a.) the numerical error diagnostic in the optimization procedure is not small, or, (b.) the DegreeOfDenominator option specifies a large number.

2. Recall that if the plant is strictly proper, then the relative degrees of the plant, closed loop and of the compensator functions satisfy

$$d(T_{rat}) = d(P) + d(C)$$

Both $d(P)$ and $d(T)$ are set in advance as input to OPTDesign[], when P is given explicitly, and when a roll off constraint is specified on T_{rat}. This determines $d(C)$ once and for all.

5.6.2 When the relative degree of the compensator is not zero

In the run above we obtained a compensator with relative degree = 0, that it, the compensator is proper but not strictly proper.

If you want to specify the relative degree of the compensator to be $nn = 1$ or larger, set the roll off constraint so that the radius function goes to 0 asymptotically with $1/w^{(nn+pp)}$, where pp is the relative degree of the plant.

Here is an example where the relative degree of the compensator is specified as $nn = 2$:

```
k0[w_]= Which[ Abs[w] < wb, 1.0,
                     True, 0.0];

r0[w_] = Which[Abs[w] < wt,  alphat,
               Abs[w] < wb,  alphagpm,
               Abs[w] < wr,  alphab,
                     True ,  alphar*Abs[p[I w]/w^2 ]
               ];
OPTDesign[p,Center->k0,Radius->r0];
```

5.7 Other performance functions (including weights)

5.7.1 Modifying the performance

OPTDesign[ ] is powerful enough to take a great variety of performance functions as input. In the sections above we discussed the class of disk performance functions. These have the form

```
Abs[ (center - z )/ radius ]^2
```

where "z" represents the closed loop transfer function. By choosing the center and radius you can solve many problems. On occasions you may want to use performance functions which are not of this type. We now show how to do this via an example.

```
p[w_] = 2/(1 - s/2)^2;

OPTDesign[p,Performance->Abs[(1+I w)/(4+w^2)*(1-z)]^2 +
                                Abs[(w^4)/(4+w^2)*z]^2]
```

For OPTDesign[] to work without problems, a requirement is that performance functions should be positive, differentiable functions of z. Note that weight functions can be made part of the performance when the latter is defined as above.

In our example above we specified a performance of the form

```
Abs[ W1 (s-z)]^2 + Abs[ W2 z]^2,
```

where W1 is not proper. This is not a problem: in fact, it is necessary in order for the problem to be well defined as a mathematical optimization problem! This is a delicate point and we direct the reader that wants more details on this to a later chapter in this book.

5.7.2 Non-rational weight functions

Weights can also be easily defined as functions of frequency. You can even define them by pieces, as the following example shows. First you define weight functions:

```
fn1[w_] := If[  Abs[w] < 1, 2., 2./Abs[w] ] ;

fn2[w_]:= Which[ Abs[w] < 1,    Abs[w]^2,
                 Abs[w] < 2,
```

```
Show[Plot[fn1[w],{w,0,6}],Plot[fn2[w],{w,0,6}]]
```

Now you are ready for the OPTDesign[] run.

```
OPTDesign[ p,Weights -> { fn1, fn2},
          Performance ->  W[1] * Abs[  1 - z ]^2 +
                             W[2]* Abs[z]^2  ]
```

OPTDesign[ ] requires that the weight functions defined in this way be bounded on the frequency axis. However, if the weight functions enter the performance function directly as rational function, they may not be bounded, as we mentiond before.

5.8 OPTDesign[] output

5.8.1 Interpretation of output diagnostics

As the computer run of OPTDesgin progresses, some numbers are printed on the screen under the labels OT1, OT2, SupGamma. These give information on the progress of the run. There is more output that is not printed, e.g., the discrete optimal closed loop T. The diagnostics on accuracy and performance are discussed below.

SupGamma. In the run above the output SupGamma = 0.75 is less than 1, thus it implies that there are solutions to the Design problem (after smoothing). However, this conclusion must be analysed under the light of other output diagnostics NED, OT1 and OT2. Note: Recall that the input to OPTDesign[] defines a performance function Gamma that depends on the frequency and on the designable transfer function T. In this chapter, this function has the form

$$Gamma(w,T) = 1/radius(Iw)Abs[center[Iw] - T[Iw]]^2.$$

The number

$$supgamma(T) := sup_w Gamma(w,T)$$

is a performance index: a real number that measures in a specific way "how good" a designable T is. If supgamma(T) < 1, then T meets all the specifications. The number SupGamma output to OPTDesign corresponds to the value at the optimal T. In a sense, this number tells you what is the best you can do with the given specifications.

NED The error diagnostic NED is 0.02 approximately. This figure says that the internal optimization routine Anopt encountered numerical noise of this magnitude when calculating the solution to the problem. We will consider this acceptable since NED is whithin 2This numerical noise is frequently associated with lack of smoothness of the data to the problem; for example, NED is large if functions such as the radius or center have jump discontinuities or sharp corners. The automatic smoothing done by OPTDesign helped in obtaining here a figure which is not "too large". Smaller values of NED are obtained with choices of grid on the I w axis with more points, or with more smoothing. This will be discussed in a different section.

OT1, OT2. The error diagnostics OT1 = 0.006 and OT2 = 0 give (loosely speaking) an indirect measure of how close to optimal the solution is. The best result possible is OT1 = 0 and OT2 = 0. In our example this was not obtained because of a nonzero NED.

5.9 Controlling the numerical error

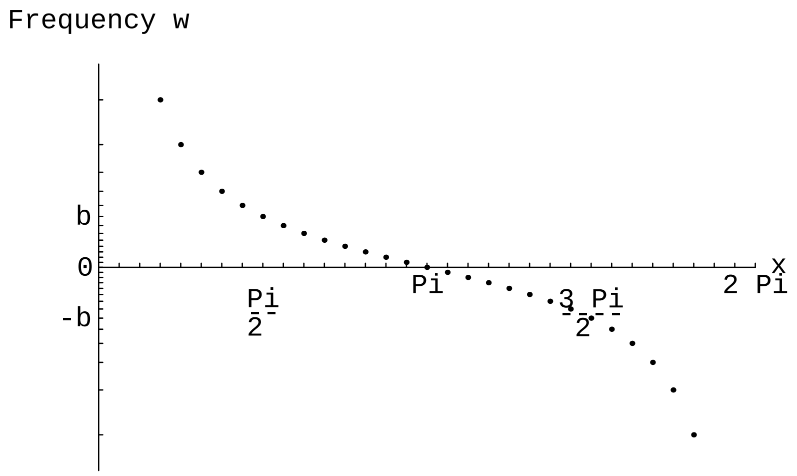
To reduce the numerical error as shown in NED, you can try two things:

1. Modify the grid, by increasing the number of points or by changing the normalization used.

2. Increase the amount of smoothing done automatically on functions.

5.9.1 Options Ngrid and GridSpread

A grid on the frequency axis is produced by OPTDesign by taking a grid of equally spaced points "x" on the interval [0,2Pi), and then calculating samples on the frequency axis via the formula w = Cot[x / 2]. The point x = 0 corresponds to infinite frequency, and "b" is a normalization constant that can be set by the user. Below is a plot of angles x vs. frequency w.



Observe how the gridpoints on the w axis accumulate near the origin. Actually, roughly half of the gridpoints will be located in the interval [-b , b], the two endpoints of the interval corresponding to x = Pi/2 and x=3 Pi/2. It is important to remember this, since good numerical behavior of computer runs depend a great deal on the choice of the grid, i.e., on the choice of the number of gridpoints and of the normalization parameter "b".

The number of gridpoints is set with the option Ngrid, and the normalization constant "b" is set with the option GridSpread.

The parameter "b" has to be chosen so that [0 , b] includes approximately what you consider low and midrange frequencies. A rule of thumb for choosing "b" is to set it equal to the bandwidth.

5.9.2 Option Nsmth

In terms of pictures, increasing the degree of smoothness corresponds to "rounding the corners" in the modified envelope of solutions (roughly speaking). But one must make sure that doing this results in an envelope that changes the original envelope by an acceptable amount.

OPTDesign has the ability to automatically "smooth out" the input data, so that the user avoids some additional work. The price paid by the user is less control on the envelope, but if smoothing is used carefully it will give the desired results. On the other hand, in some cases one must resort to setting the amount of smoothing in order to obtain good numbers.

This automatic smoothing is specified with the option **`Nsmth`**. The input **`Nsmth`**− >`1` indicates a small amount of automatic smoothing; more smoothing is achieved with **`Nsmth`**− >`5` or **`Nsmth`**− >`10`. The input **`Nsmth`**− >`0` corresponds to no automatic smoothing. Some precautions must be taken when setting this input since smoothing modifies the envelope.

Part II

Advanced Design

Chapter 6

Examples

Three examples are presented in this chapter. They illustrate the theory developed in previous chapters, as well as the use of some techniques for solving problems in practice. Each example is preceded by a chapter with material related to the example.

The design process involves extensive use of computers and algorithms, so some time spent studying their efficient use is beneficial. We discuss briefly in Section 6.1 numerical issues that arise in a wide class of algorithms for solving optimization problems. The first control example we solve, the design of a control of a slide drive, is presented in Section 6.2.

Many practical situations lead to control design problems with requirements on the time domain functions associated with the system. To use frequency domain methods, one must translate time domain information to the frequency domain. We take a first look at time domain requirements in Section 6.3. The requirements overshoot and the settling time and a method for translating information to the frequency domain are introduced in this section. The method is applied in our solution to a design problem in Section 6.4. A more sophisticated method for translating time domain information is presented in Chapter 6.

The generic problem in control, as in many other disciplines, is to achieve an objective subject to certain constraints. Achieving the objective usually involves a tradeoff: one constraint cannot be attained to perfection without the rest doing badly. These are called "competing constraints". In the design problems studied here, this situation appears when different constraints are active at the same frequency. In Section 6.5 we discuss ways to treat competing constraints with our methods; this is used to solve the design problem in Section 6.6.

6.1 Numerical Practicalities

Several difficulties may arise when OPT problems are solved by computer, because algorithms for solving OPT will not work, or will work in an unsatisfactory manner, if any of the following occur:

- The performance function Γ is not continuous. In circular performance case, this occurs when the radius function or the center function is not continuous.
- The value of the performance $\Gamma(\omega, T(j\omega))$ is extremely large at some frequencies ω, regardless of the value of T. In circular performance problems, this is the case when the radius gets very close to 0 at one or more frequencies.
- Points on the $j\omega$ axis for sampling functions are badly chosen.
- At certain frequency ω, the value of the performance $\Gamma(\omega, T(j\omega))$ depends on ω but does not depend on T.

We discuss these issues below. The emphasis is on circular performance functions, but the comments apply, for the most part, to other performance functions as well.

6.1.1 Discontinuous functions and smoothing

Depending on the way in which requirements are put together, the performance function may not be continuous. For example, suppose that you have a case in which the only two constraints active near frequency $\omega = 1$ are: a gain-phase constraint active for $\omega < 1$, and a bandwidth constraint active for $\omega \geq 1$. A circular performance problem formed with these constraints has a center function with values that drop from 1 to 0 near frequency $\omega = 1$. This is undesirable; most algorithms for solving OPT have numerical problems when dealing with discontinuous performance functions. Even if the optimal designable function T could be computed, in many cases it wouldn't be acceptable because it is not continuous.

To circumvent this problem in the circular performance case, it is necessary to "smooth" R and K before forming the performance, which modifies these functions to make them continuous and differentiable. Smoothing changes the problem being solved, but it still should produce good approximate solutions if handled with care. The trade-off is closeness to the original function vs. good numerical properties. Smoothing is usually based on convolution or spline interpolation. Figs. 6.1 and 6.2 illustrate the effect of smoothing.

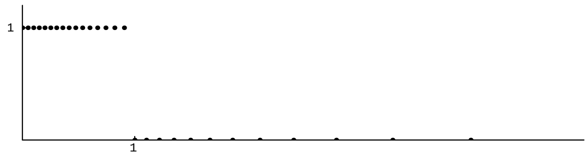


Figure 6.1: Sample values of a discontinuous function

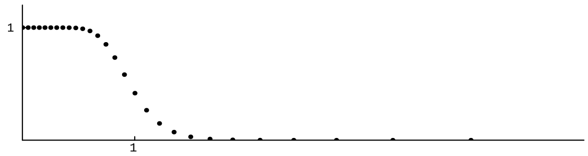


Figure 6.2: Smoothed sample values of a discontinuous function

6.1.2 Vanishing radius function

When the radius in circular performance problems vanishes, the mathematics for solving OPT break down. However, we show below that a vanishing radius is not a difficulty if the problem is set up correctly.

A radius formed with the performance measures given in Chapter 3 will vanish at infinite frequency because of the closed loop-roll off constraint. This constraint is

$$|T(j\omega)| \leq \alpha_r \frac{|P(j\omega)|}{|\omega|^{d_c}}, \qquad \text{for large } |\omega| \tag{6.1}$$

where d_c is the degree of the desired compensator, as specified by the designer. Consider now the parameterization of all T obtained from the requirement of internal stability:

$$T = A + BT_1$$

where A and B are $\mathcal{RH}^\infty$ functions that satisfy certain interpolation conditions at RHP zeros and poles of the plant (see Theorem 9). Because of the way it is chosen, B has relative degree given by $d(B) = d(P) + d_c$. Therefore the right-hand-side in the inequality

$$\left| \frac{K(j\omega) - A(j\omega)}{B(j\omega)} - T_1(j\omega) \right| \leq \frac{|P(j\omega)|}{|w|^{d_c} |B(j\omega)|} \tag{6.2}$$

does not vanish at infinite frequency. That is, when the problem is stated in terms of the new parameter T_1, the new radius function does not vanish at infinite frequency.

A similar situation occurs when the radius has a zero at certain frequency ω_0. This happens when the plant has a zero or a pole on the $j\omega$ axis, so there is a plant bound or compensator bound requirement in the design problem. We leave as an exercise for the reader to show how the technique of reparameterization can produce a circular performance function whose radius does not vanish.

6.1.3 Sampling functions on the $j\omega$ axis

To solve OPT by computer one must choose a grid of points on the $j\omega$ axis. These points are needed for sampling functions. To have an adequate discrete representation of functions, the grid must contain many points in the frequency bands where the functions being sampled change alot. For circular performance functions, the center K and the radius R determine which choices of grid points are appropriate. Figs. 6.3 and 6.4 illustrate different selections of grid.

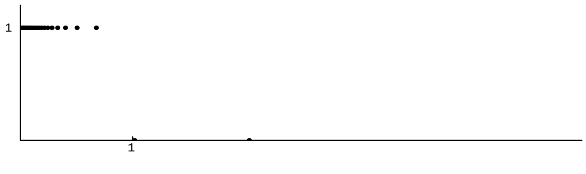


Figure 6.3: High frequency undersampled

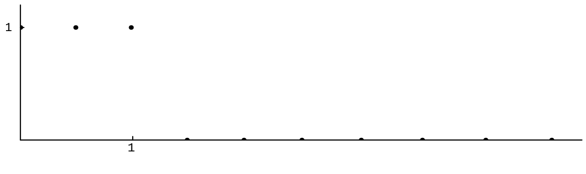


Figure 6.4: Low frequency undersampled

6.2 Design Example 1

The system is a slide drive, driven by a DC motor (see Fig. 6.5). This was studied by Brett Larson in his master's thesis [La][1], written under the guidance of Prof. Fred Bailey. The following description appears in page 3 of the thesis:

> " ...an electrical voltage $u(t)$ drives a DC motor with output speed $\omega_1(t)$. The output of the motor goes through the gear train which steps the speed down to $\omega_2(t)$. The gear train is connected to a table which slides along a rail with the load on top. "

The fundamental goal in this problem is to control the speed of the table very precisely.

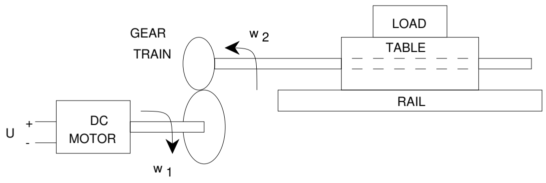


Figure 6.5: Slide drive apparatus.

6.2.1 Electro-mechanical and electrical models

Ignoring the dynamics in the gear train the problem can be represented by the equivalent circuit in Figure 6.6.

The following equations correspond to the DC motor.

$$\begin{aligned} \tau_M &= K \\ e_G &= K\omega_1 \end{aligned} \qquad (6.3)$$

---

[1] See also [SS]

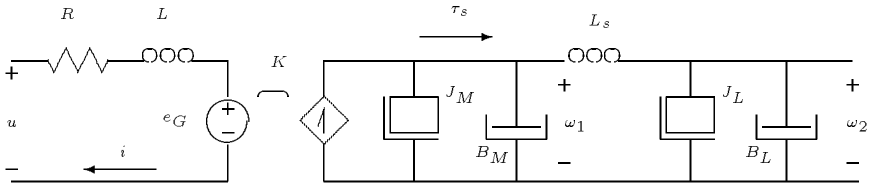


Figure 6.6: Electro-mechanical model

The parameters for this problem are

$$\begin{array}{rcl}
R & = & 1\Omega \\
L & = & 5mH \\
K & = & 0.2V/rps = 0.2N - m/amp \\
J_M & = & 2.010^{-3}Kg - m^3 \\
B_M & = & 2.010^{-3}N - m/rps \\
K_s & = & 1000N - m/rad \\
0.5J_M & \leq & J_L \leq 5.0J_M \\
J_{L_{nom}} & = & J_M \\
B_L & = & 1N - m/rps
\end{array} \tag{6.4}$$

Figure 6.7 represents this problem with the mechanical parts replaced by their electrical equivalents, where

$$\begin{array}{rcl}
C_m & = & J_M \\
R_M & = & 1/B_M \\
L_S & = & 1/K_s \\
C_L & = & J_L \\
R_L & = & 1/B_L
\end{array} \tag{6.5}$$

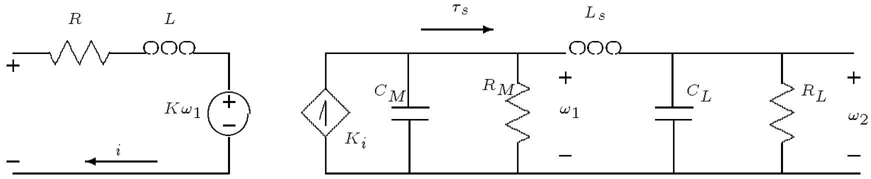


Figure 6.7: Electrical model

6.2.2 Mathematical model

Choosing the state vector as

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} i \\ \omega_1 \\ \tau_x \\ \omega_2 \end{pmatrix} \tag{6.6}$$

we have that the system of state equations for the system is

$$\begin{aligned} x'(t) &= Ax(t) + bu(t) \\ y(t) &= cx(t) + du(t) \end{aligned} \tag{6.7}$$

Take ω_1 as the output and scale frequency by 100 to obtain the following coefficient matrices, where $\alpha = 1/J_L$:

$$\begin{aligned} A &= \begin{pmatrix} -2 & -0.4 & 0 & 0 \\ 1 & -0.01 & -5 & 0 \\ 0 & 10 & 0 & -10 \\ 0 & 0 & \alpha & \alpha \end{pmatrix} \\ b &= \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ c &= (0\ 1\ 0\ 0) \\ d &= (0) \end{aligned} \tag{6.8}$$

The parameter α varies between 1 and 10. The maximum load corresponds to $\alpha = 1$. The plant transfer function from input u to output ω_1 is computed from the formula

$$P_\alpha(s) = c(sI - A)^{-1}b \tag{6.9}$$

We obtain,

$$P_\alpha(s) = \frac{2s^2+2\alpha s+20\alpha}{s^4+(2.01+\alpha)s^3+(50.42+12.01\alpha)s^2+(100+70.52\alpha)s+104.2\alpha} \tag{6.10}$$

The parameter α accounts for the variable load in this example. Here α varies between $\alpha = 10$, which is the minimum load case, and $\alpha = 1$, which is the maximum load case. The nominal case corresponds to $\alpha = 5$. The nominal plant is

$$\begin{aligned} P_5(s) &= \frac{2s^2+10s+100}{s^4+7.01s^3+110.47s^2+452.6s+521} \\ &= \frac{(s+2.5\pm j6.61437)}{(s+2.33827\pm j0.272238)(s+1.16673\pm j9.62573)} \end{aligned} \tag{6.11}$$

The nominal plant is stable (so are all the other plants P_α for all $\alpha \in [1, 10]$).

6.2.3 Statement of the problem

The set of requirements given here is taken from [La].

1. $|S(j\omega)| < -20\ dB$, for $|\omega| < 0.12\ rad/sec$.
2. $|S(j\omega)| < 10\ dB$, for $|\omega| < 1.0\ rad/sec$.
3. $|T(j\omega)| < -20\ dB$, for $|\omega| > 5.0\ rad/sec$.
4. $|T(j\omega)| < -40\ dB$, for $|\omega| > 10.0\ rad/sec$.
5. For stability, a phase margin greater than 30 deg.

According to Larson, these requirements should be met over the full range of load perturbations. We will instead do the (much easier) problem of solving *Design* for the nominal plant only [2]. The analysis of the performance of the system for loads other than the nominal will not be discussed here, except for the comparison of the closed loop transfer functions shown in Fig. 6.17.

6.2.4 Reformulation of requirements

Now we rewrite the requirements in 6.2.3 using the notation of Chapter 3. The translation is quite direct. The only constraint that does not have an obvious translation is the phase-margin requirement; to treat it we rewrite it as a gain-phase margin constraint with $m = 0.5$:

$$|1 - T(j\omega)| < \frac{1}{0.5} = 2, \qquad \forall\omega$$

---

[2] To treat this problem, Larson approximated the nominal plant with the simpler plant $P(s) = 2.2188/(s+3.4)^2$.

It is clear that the gain phase margin constraint is the only one that is binding at frequencies $1 < \omega < 5$. Also note that $m = 0.5$ insures a gain margin of 6 dB (see Fig. 6.8).

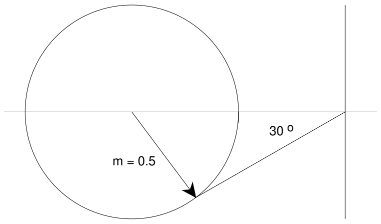


Figure 6.8: Derivation of m in Design Example 1.

An important observation is that the set of constraints in 6.2.3 lacks a closed loop roll-off constraint. We supply one by requiring

$$|T(j\omega)| < \frac{0.01}{|P(j10.)|}|P(j\omega)|, \qquad \text{for } \omega > 10.$$

This insures that constraint 4. in 6.2.3 is satisfied, since $|P(j\omega)|$ is decreasing in the band $10. < \omega$. The complete set of performance requirements is:

$$\begin{array}{rcll} |1-T(j\omega)| & < & 0.12, & 0 \leq \omega \leq 0.1 \\ |1-T(j\omega)| & < & 0.32, & 0 \leq \omega \leq 1.0 \\ |1-T(j\omega)| & < & 2, & 1 < \omega < 5.0 \\ |T(j\omega)| & < & 0.1, & 5 < \omega < 10.0 \\ |T(j\omega)| & < & 0.01, & 10 < \omega \\ |T(j\omega)| & < & 0.01|P(j\omega)|/|P(j10.)|, & \omega > 10 \end{array} \tag{6.12}$$

Now we turn to internal stability. The plant $P(s)$ has no zeros or poles in the closed RHP. It also has relative degree $d(P) = 2$. This implies that the set INT of interpolation conditions to be satisfied by T, for internal stability with proper compensator, is

$$INT \qquad rd(T) = 2$$

6.2.5 Formulation of the performance function

A circular performance function

$$\Gamma(\omega, z) = \frac{1}{R(j\omega)^2}|K(j\omega) - z|^2 \tag{6.13}$$

can be formed from the performance requirements by setting

$$K(j\omega) = \begin{cases} 1, & |\omega| < 5.0 \\ 0, & |\omega| \geq 5.0 \end{cases} \tag{6.14}$$

and

$$R(j\omega) = \begin{cases} 0.12, & |\omega| < 0.10 \\ 0.32, & 0.10 < |\omega| < 1.0 \\ 2.0, & 1.0 < |\omega| < 5.0 \\ 0.1, & 5.0 < |\omega| < 10.0 \\ 0.01|P(j\omega)|/|P(j10)|, & |\omega| > 10.0 \end{cases} \tag{6.15}$$

Fig. 6.9 shows the requirements envelope. We now have center and radius functions with large jumps, and for good numerical behavior of algorithms, we attempt smoothing by averaging near the jumps. However, this leads to an envelope which is larger than the envelope from the set of constraints (see Fig. 6.10). One way to fix this problem is to enlarge the bands on which the constraints are tighter, and then do the averaging at jumps. The envelope obtained in this fashion can be contained in the original envelope. However, there are large changes in the smooth envelope near the frequencies that previously corresponded to jumps. If possible this situation must be avoided, since *large changes in the envelope over small frequency bands correspond to large changes in the optimal designable function T over the same bands.* This occurs for rational T when T has poles near the $j\omega$ axis, that is, T is nearly unstable.

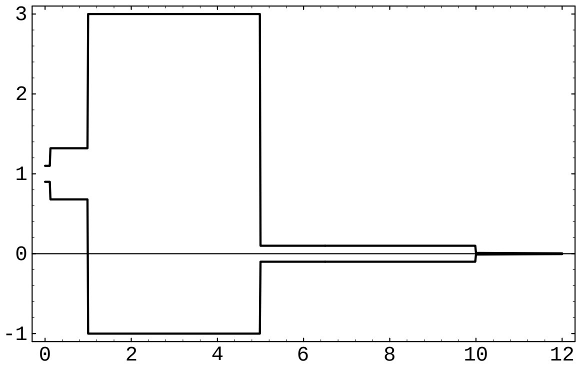


Figure 6.9: Requirements envelope.

We now try a simple modification of the envelope. The boundary of the envelope, which is mostly piecewise constant, is replaced by a continuous

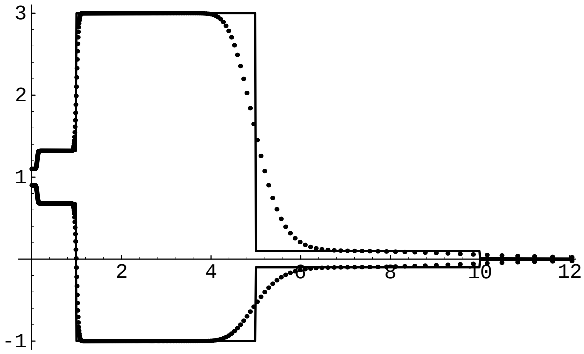


Figure 6.10: Requirement envelopes: before and after smoothing.

function which is piecewise linear at low and mid-frequency ranges. Keep in mind that the resulting envelope must be contained in the original one.

To do this modification, note that on the frequency band $1 < \omega < 5$ the radius is relatively large, and that the center drops from 1 to 0. Set the center in this band to drop linearly from 1 at $\omega = 1.0$, down to 0 at $\omega = 0.1$:

$$K(j\omega) = \begin{cases} 1 & \text{if } |\omega| < 1.0 \\ -0.25(\omega - 1) + 1 & \text{if } 1.0 < |\omega| < 5.0 \\ 0 & \text{if } |\omega| \geq 5.0 \end{cases} \tag{6.16}$$

Now pick a radius in a similar way, so that it is continuous and the resulting envelope does not become larger than the original one. Set

$$\begin{aligned} \{\omega_0, \omega_1, \ldots, \omega_8\} &= \{0., 0.12, 0.5, 1., 2., 3., 5., 10.\}, \\ \{y_0, y_1, \ldots, y_8\} &= \{0.1, 0.1, 0.32, 0.32, 1.75, 1.5, 0.1, 0.01\}, \\ m_\ell &= (y_{\ell+1} - y_\ell)/(\omega_{\ell+1} - \omega_\ell),\ \ell = 0, \ldots, 7 \end{aligned} \tag{6.17}$$

and

$$R(j\omega) = \begin{cases} m_\ell(|\omega| - \omega_\ell) + y_\ell & \text{if } \omega_\ell \leq |\omega| < \omega_{\ell+1},\ \ell = 0, \ldots, 7 \\ \\ 0.01|P(j\omega)|/|P(j10.)| & \text{if } |\omega| \geq 10.0 \end{cases} \tag{6.18}$$

The resulting requirements envelope is depicted in Fig. 6.11.

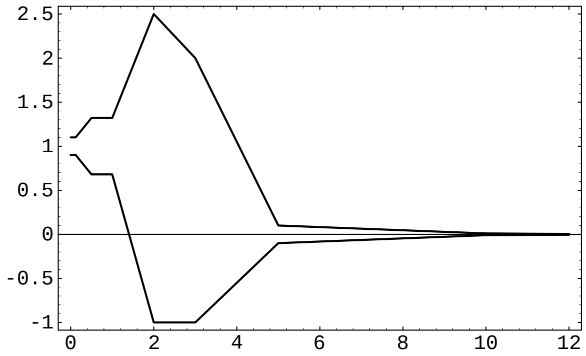


Figure 6.11: Modified envelope.

6.2.6 Optimization

We now have all the elements necessary to state and solve the optimization problem:

$$
\begin{aligned}
\gamma^* &= \inf_{T \ sat. \ INT} \sup_\omega \tfrac{1}{R(\omega)^2} |K(j\omega) - T(j\omega)|^2 \\
&= \inf_{T_1} \sup_\omega \tfrac{1}{R_1(\omega)^2} |K_1(j\omega) - T_1(j\omega)|^2
\end{aligned}
\qquad (6.19)
$$

where $K_1 = (K - A)/B$, $R_1 = R/B$ and $T = A + BT_1$.

Calculating the answer by computer [3] yields $\gamma^* = 0.98$. By Proposition 3 in Chapter 4 this means that there exist solutions to *Design*. Plots of the optimal T and optimal L are shown in Figs. 6.12 – 6.15.

6.3 Time domain performance requirements

Time domain requirements on the closed loop system $\mathcal{S}$ must be translated to the frequency domain in order to solve design problems with the methods used in this book. In this section, two time domain requirements are introduced, and a technique for treating them is given. The technique is applied later in a second design problem (Section 6.4). We warn the reader that the technique introduced in this section may not be powerful enough to solve some problems; the methods presented later may give better results.

---

[3] See Part II for the input of a computer session with the package OPTDesign

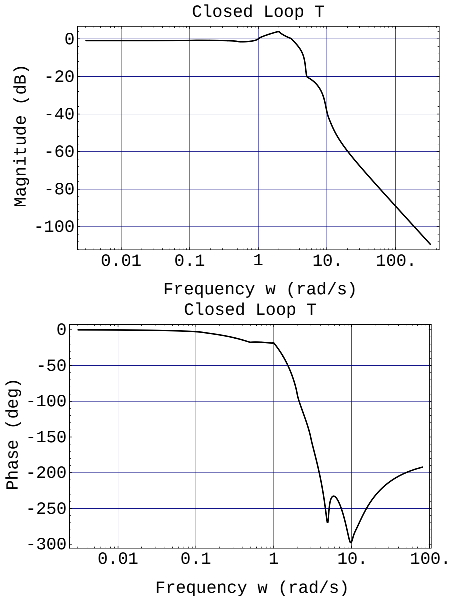


Figure 6.12: Bode plots of optimal closed loop T in design example 1.

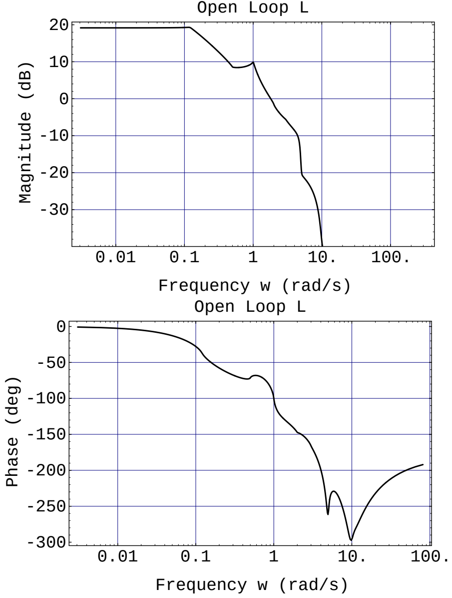


Figure 6.13: Bode plots of optimal open loop L in design example 1.

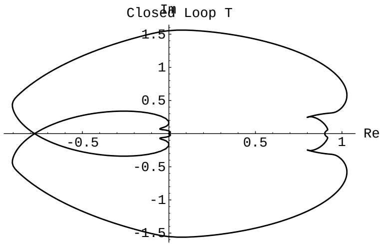


Figure 6.14: Polar plot of optimal T in design example 1.

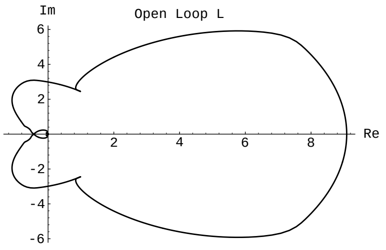


Figure 6.15: Polar plot of optimal L in design example 1.

6.3.1 Two common time domain requirements

Two common time domain requirements used for system design are the overshoot requirement and the settling time requirement. Both are defined in terms of the step response curve $c(t)$, which is the response of the system in the time domain to an input that is a unit step (see Fig. 6.18). The step response function has $T(s)/s$ as Laplace transform i.e.,

$$c(t) = \mathcal{L}^{-1}[T(s)/s](t) \tag{6.20}$$

The requirements we present in this sections are

- **Overshoot requirement**:
  Given $M > 0$,
  $$c(t) - 1 < M, \ \forall t > 0 \tag{6.21}$$
- **Settling time requirement**:
  Given $t_s > 0$ and $\Delta > 0$,
  $$|c(t) - 1| < \Delta, \qquad \forall t > t_s \tag{6.22}$$

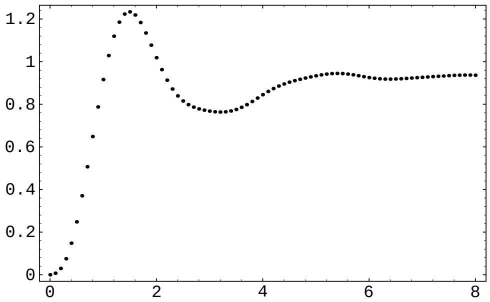


Figure 6.16: Step Response curve of optimal system in design example 1.

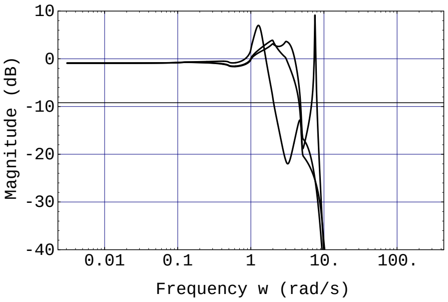


Figure 6.17: Magnitude of the closed loop transfer function for different plants in the parameter range: $\alpha = 1, 5, 10$. The compensator is optimal for $\alpha = 5$.

6.3.2 A naive method

A very common method for handling time domain requirements in the frequency domain is to use a low order rational function T_{ref}. The function T_{ref} is the transfer function of a well understood system that meets the time domain requirements. Frequency domain requirements that correspond to given time domain requirements are established in terms of the function T_{ref}.

A simple example is the closed loop system with transfer function

$$T_{ref}(s) = \frac{\omega_n^2}{s^2 + 2\zeta s\omega_n + \omega_n^2} \tag{6.23}$$

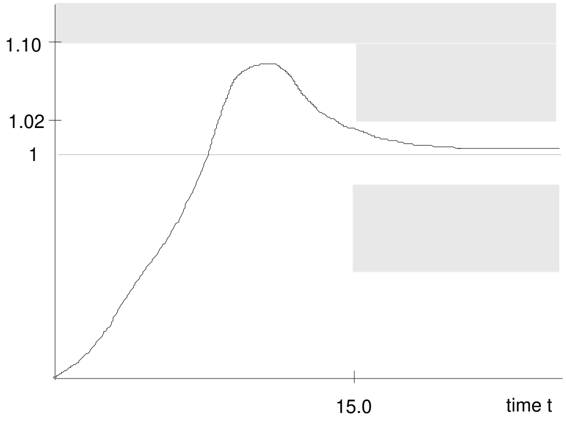


Figure 6.18: A step response function that satisfies the overshoot and settling time requirements when $M = 0.1$, $t_s = 15.0$ and $\Delta = 0.02$. In this figure, the step response curve cannot run into the shaded area without violating the constraints.

This system is known to satisfy the relations (see [FPE])

$$t_s \approx \frac{4.6}{\zeta\omega_n} \tag{6.24}$$

and

$$M_p = \mathrm{Exp}\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right) \tag{6.25}$$

Hence $T_{ref}(s)$ is completely determined by specifying t_s and M_p. Also determined is the open loop $L_{ref} = T_{ref}/(1 - T_{ref})$. Bode plots of T_{ref} and L_{ref} are then used to determine parameters such as bandwidth, peak magnitude, etc.

6.3.3 Limitations of the naive method

Relationships between time domain and frequency domain constants, like the ones described in Section 6.3.2, are commonly used in practice as rules of thumb for system design. However, they do not hold for general closed loop systems, and thus may give misleading information. Another important weakness of the naive method is that the function T_{ref} has nothing in it that accounts for the characteristics of the given plant P.

6.4 Design Example 2

The following design problem is taken from [FPE, p. 497]. *Design* is treated here with the naive method for converting time domain requirements just described in Section 6.3.2.

6.4.1 Statement of the problem

The problem is to design a satellite's attitude control. The plant is the rational function

$$P(s) = \frac{0.036(s+25)}{s^2(s+0.02+2j)(s+0.02-2j)}$$

The requirements on the step response function $c(t)$ are overshoot requirement,

$$c(t) < 1.15 \qquad (\forall t \geq 0),$$

and settling time,

$$|c(t)-1| < 0.01,\ \forall t,\ t \geq 14.0$$

We need to find a closed loop system $\mathcal{S}$ that is internally stable and meets the performance requirements.

6.4.2 Translation of time domain requirements

Set $t_s = 14.$ and $M_p = 0.15$. Substituting these into equations (6.24) and (6.25) gives $\zeta \approx 0.52$ and $\omega_n \approx 0.64$. Thus (6.23) becomes

$$T_{ref}(s) = \frac{0.404}{0.404 + 0.657s + s^2} \tag{6.26}$$

The corresponding step response function $c_{ref}(t) = \mathcal{L}^{-1}(\frac{T_{ref}(s)}{s})$ is given by

$$c_{ref}(t) = 1 - (0.6039\, sin(0.544t) + cos(0.544t))\, e^{-0.328571t} \tag{6.27}$$

Figs. 6.19 – 6.22 show plots associated with T_{ref}. Frequency domain requirements are set now from these plots.

Midrange frequencies: tracking and bandwidth

We use a combination of the plots of $|T_{ref}|$ and $|1 - T_{ref}|$ (Fig 6.22) to set frequency dependent tracking and bandwidth constraints:

$$\begin{aligned} |1 - T(j\omega)| &< |1 - T_{ref}(j\omega)|, & |\omega| < 0.8 \\ |T(j\omega)| &< |T_{ref}(j\omega)|, & |\omega| > 0.8 \end{aligned} \tag{6.28}$$

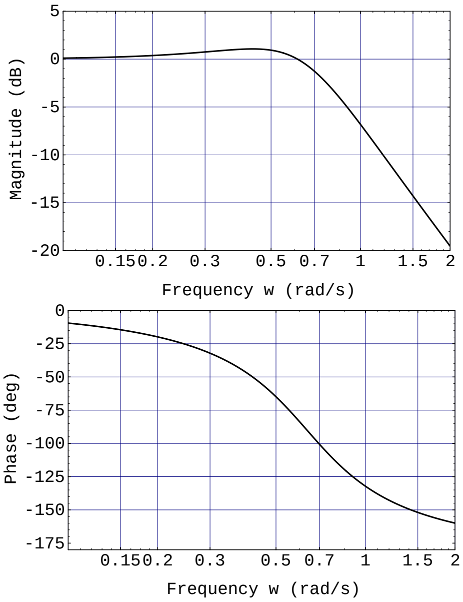


Figure 6.19: Bode plots of T_{ref}.

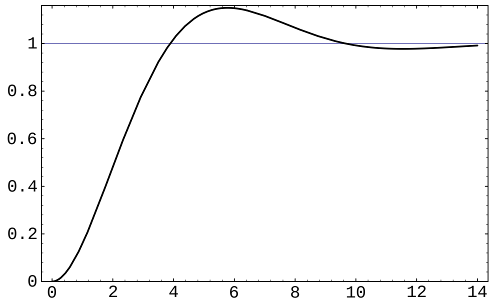


Figure 6.20: Step response for T_{ref}.

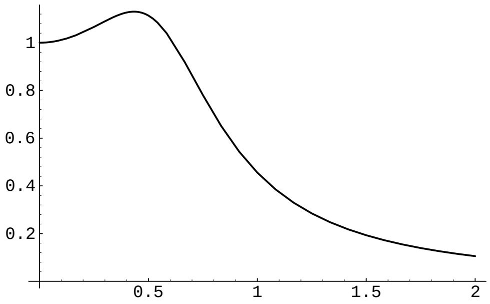


Figure 6.21: Plot of $|T_{ref}|$.

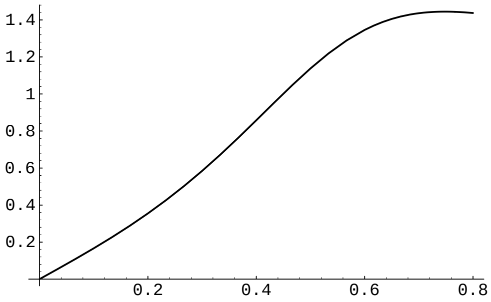


Figure 6.22: Plot of $|1 - T_{ref}|$.

High frequency: closed loop roll off

We specify high frequency behavior of the designable T with the roll-off constraint: $|T(j\omega)| < \alpha_r |P(j\omega)|$, valid for $\omega > \omega_r$. The value of α_r is chosen to have continuous envelope at the point $\omega = \omega_r$.

$$|T(j\omega)| \leq |\frac{T_{ref}(j10.)}{P(j10.)}||P(j\omega)|, \qquad |\omega| \geq 10.0 \tag{6.29}$$

Low frequency: plant bound constraint

Since the plant P has a pole at the origin, it is necessary to consider the plant bound constraint $|(1 - T(j\omega))P(j\omega)| < \alpha_p$, valid for $|\omega| < \delta_p$. Somewhat arbitrarily choose $\delta_p = 0.1$, and α_p is selected to make $|1 - T(j\delta)| \leq |1 - T_{ref}(j\delta)|$:

$$|T(j\omega) - 1| \leq |P(j0.1)(1 - T_{ref}(j0.1))| \frac{1}{|P(j\omega)|}, \quad |\omega| < 0.1 \tag{6.30}$$

6.4.3 Forming the performance function Γ

A circular performance function

$$\Gamma(\omega, z) = \frac{1}{R(j\omega)^2}|K(j\omega) - z|^2 \tag{6.31}$$

can be formed from inequalities (6.28) through (6.30). Inequalities (6.28) lead to center and radius functions with a jump discontinuity at frequency $\omega = 0.8$. To avoid numerical difficulties we compromise and set a constraint defined on the band $0.6 < \omega < 0.8$ that smooths the transition between the constraints in (6.28). Simply set the radius function K to drop linearly, from $K(j0.6) = 1.$ down to $K(j0.8) = 0.$. The radius function R is also set to be linear on the band $0.6 < \omega < 0.8$, with endpoints given by the constraints (6.28).

$$K(j\omega) = \begin{cases} 1, & |\omega| < 0.5 \\ \frac{-1}{0.2}(|w| - 0.8), & 0.6 \leq |\omega| < 0.8 \\ 0, & |w| \geq 0.8 \end{cases} \tag{6.32}$$

and

$$R(j\omega) = \begin{cases} \frac{|P(j0.1)(1-T_{ref}(j0.1))|}{|P(j\omega)|}, & |\omega| < 0.1 \\ |1 - T_{ref}(j\omega)|, & 0.1 < |\omega| < 0.6 \\ \frac{|T_{ref}(j0.8)| - |1-T_{ref}(j0.6)|}{0.2}(|\omega| - 0.6) \\ +|1 - T_{ref}(j0.6)|, & 0.6 < |\omega| < 0.8 \\ |T_{ref}(j\omega)|, & 0.8 < |\omega| < 10.0 \\ |\frac{T_{ref}(j10.)}{P(j10.)}||P(j\omega)|, & |\omega| > 10.0 \end{cases} \tag{6.33}$$

The corresponding requirements envelope is shown in Fig. 6.23.

6.4.4 Optimization

Proceed as in Section 6.2.6 to fomulate OPT. Solving by computer produces an optimum $\gamma^* = 1.269$, indicating that there are no solutions to *Design* for the given set of constraints.

Since the frequency domain requirements were obtained from T_{ref}, there is a certain amount of flexibility in resetting them, which we now do. Empirical observations play a role here: for example, it is known that in practice large bandwidth is associated with short rise time. We relax the low frequency constraints below to see if the new constraints lead to a satisfactory answer.

Consider the following modified constraints obtained by increasing the

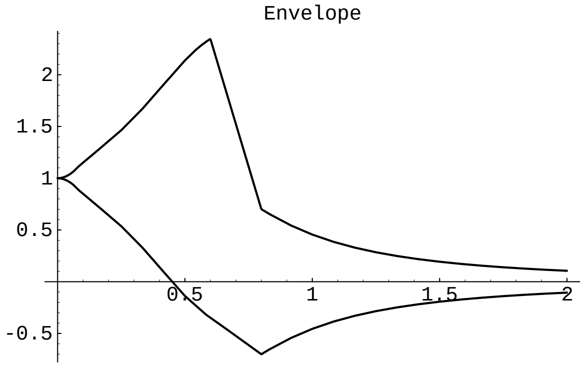


Figure 6.23: Requirements envelope.

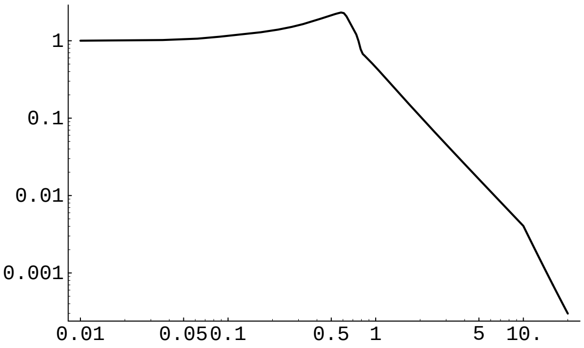


Figure 6.24: Log-log plot of the function $K + R$.

radius function on the low frequency range:

$$R(j\omega) = \begin{cases} 1.5\frac{|P(j0.1)(1-T_{ref}(j0.1))|}{|P(j\omega)|}, & |\omega| < 0.1 \\ 1.5|1-T_{ref}(j\omega)|, & 0.1 < |\omega| < 0.6 \\ \frac{|T_{ref}(j0.6)|-1.5|1-T_{ref}(j0.8)|}{0.2}(|\omega|-0.6) & \\ +|1-T_{ref}(j0.6)|, & 0.6 < |\omega| < 0.8 \\ |T_{ref}(j\omega)|, & 0.8 < |\omega| < 10.0 \\ |\frac{T_{ref}(j10.)}{P(j10.)}||P(j\omega)|, & |\omega| > 10.0 \end{cases} \tag{6.34}$$

The envelope corresponding to the center function K and the radius R in (6.34) is shown in Fig. 6.25. Solving by computer with the modified constraints gives $\gamma^* = 0.73$, which indicates that there are solutions to the modified optimization problem OPT. Plots 6.26 – 6.27 correspond to the optimal system $\mathcal{S}$. Observe that the time domain constraints are not met; overshoot is 0.25, and $c(14.0) \approx 0.05$.

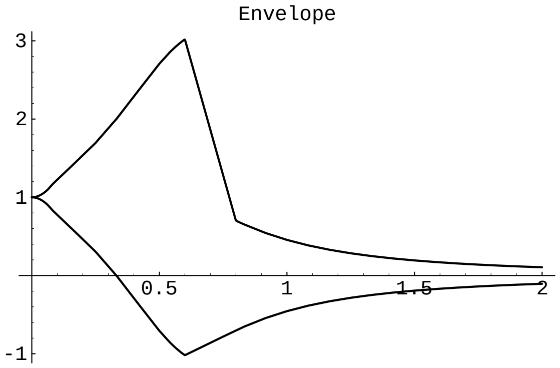


Figure 6.25: Modified envelope.

6.5 Design problems with competing constraints

6.5.1 Competing constraints

Design problems can easily lead to situations in which several inequalities, binding at the same frequency, are expected by the designer to be satisfied simultaneously by the functions of the system. In this case, the inequalities are called competing constraints.

In general, a set of competing constraints at frequency ω can be stated in the form

$$\begin{array}{rcl} \Gamma_1(\omega, T(j\omega)) & \leq & 1 \\ \Gamma_2(\omega, T(j\omega)) & \leq & 1 \\ & \vdots & \\ \Gamma_m(\omega, T(j\omega)) & \leq & 1 \end{array} \tag{6.35}$$

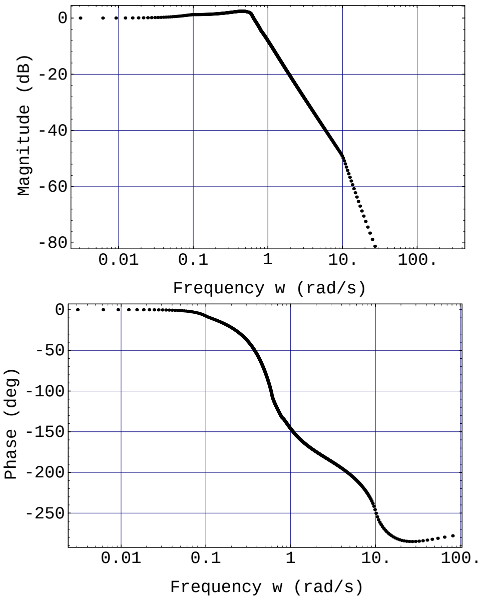


Figure 6.26: BodePlots of optimal T for Design Problem 2.

The inequalities in (6.35) can be written as a single inequality as follows:

$$\Gamma(\omega, T(j\omega)) := \max_{1 \leq k \leq m} \Gamma_k(\omega, T(j\omega)) \leq 1 \tag{6.36}$$

Example

Suppose a system $\mathcal{S}$ is to be designed, such that at frequency ω the designable transfer function T satisfies

$$|T(j\omega) - 1| \leq 1/2 \tag{6.37}$$

and

$$|T(j\omega)| \leq 6/5 \tag{6.38}$$

These inequalities can be written as

$$\begin{aligned} 2|T(j\omega) - 1| &\leq 1 \qquad (6.39) \\ \frac{5}{6}|T(j\omega)| &\leq 1 \qquad (6.40) \end{aligned}$$

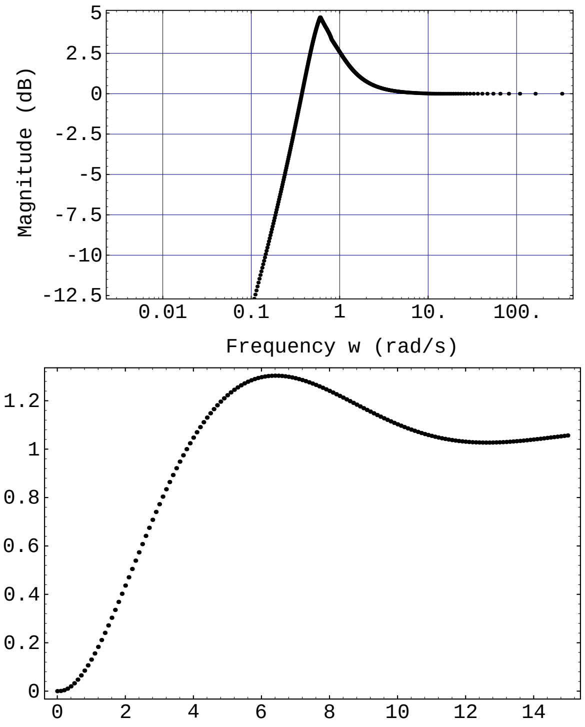


Figure 6.27: Plots of magnitude of optimal sensitivity S and of step response for design problem 2.

The set S_ω of all possible $T(j\omega)$ that satisfy both (6.39) and (6.40) is a subset of the complex plane. This set can be described as follows:

$$S_\omega = \{ \ s : \ max(\ 2|s-1| \ , \ \frac{5}{6}|s| \ ) \leq 1 \ \}$$

Note that S_ω is the intersection of two disks. Another important feature of S_ω is that its boundary has " corners ".

6.5.2 Performance from competing constraints

The function Γ in (6.36) is not smooth even if the functions Γ_k are smooth. This can be seen from the *sublevel sets*

$$S_\omega(\gamma) = \{ \ s : \ \Gamma(\omega, s) \leq \gamma \ \}, \qquad (6.41)$$

which in many cases contain corners on their boundary (see Fig. 6.28). Such functions $\Gamma(\omega, z)$ are not differentiable in z. This is important since

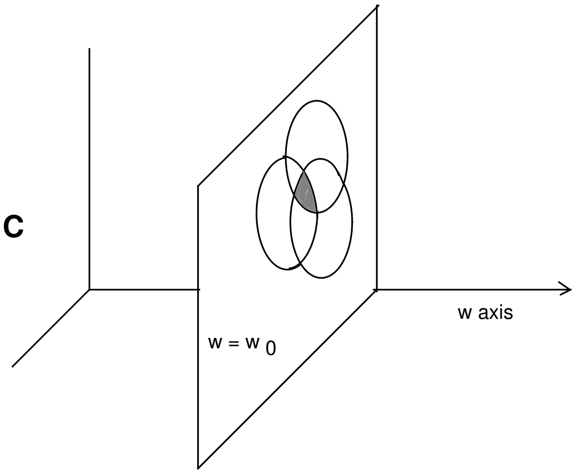


Figure 6.28: Sublevel sets S_ω from competing constraints

some algorithms for solving *OPT* require smooth functions Γ. Thus to solve *OPT* with these algorithms one must compromise (see Section 6.5.3).

Note that S_ω is convex if it is the intersection of convex sets, such as disks. Algorithms do exist that handle functions Γ with convex sublevel sets, even if these have corners. See [BB] for more details on these algorithms.

6.5.3 Solving *OPT*

To solve *OPT* for smooth Γ, gradient-descent/Newton type of iterative algorithms can be used[4] (see [HMer]). These algorithms do not apply directly when Γ is of the form (6.36). However, functions Γ from competing constraints can be handled when certain compromises are made.

To simplify, the discussion we take $m = 2$ in (6.36) to obtain the performance function

$$\Gamma(\omega, z) = \max(\Gamma_1(\omega, z), \Gamma_2(\omega, z)) \tag{6.42}$$

We describe in Subsections 6.5.4 and 6.5.5 functions $\Gamma'(\omega, z)$ that are *smooth* approximations to Γ. *OPT* for Γ' can be solved, and then the optimal T is

---

[4] The software OPTDesign is an implementation of these algorithms.

evaluated with the original requirements. In many cases this procedure is good enough to produce systems that satisfy all the constraints.

6.5.4 Smoothing the performance function

One method to obtain a smooth performance function Γ from (6.42), if Γ_1 and Γ_2 are smooth, is based on optimizing the following performance function:

$$\Gamma'(\omega, T(j\omega)) = \Gamma_1(\omega, T(j\omega))^2 + \Gamma_2(\omega, T(j\omega))^2 \tag{6.43}$$

Suppose that T is such that

$$\Gamma'(\omega, T(j\omega)) \leq 1 \tag{6.44}$$

Since for any positive numbers x and y,

$$\max(x, y) \leq \sqrt{x^2 + y^2}$$

it follows that both

$$\Gamma_1(\omega, T(j\omega)) \leq 1 \tag{6.45}$$

and

$$\Gamma_2(\omega, T(j\omega)) \leq 1 \tag{6.46}$$

hold. Therefore, if solving OPT with the function Γ' as performance yields an optimal value $\gamma^* \leq 1$, then the optimal T satisfies both (6.45) and (6.46). One can try variations on the formula (6.43), such as[5] using an exponent higher than 2 in 6.43.

6.5.5 Weight functions

Choose weight functions of frequency W_1 and W_2, and form

$$\Gamma'(\omega, z) = W_1(j\omega)\Gamma_1(\omega, z) + W_2(j\omega)\Gamma_2(\omega, z) \tag{6.47}$$

These weight functions can be chosen to emphasize either Γ_1 or Γ_2 at each frequency ω. A performance function of the type (6.47) is used in Design Example 3 below.

---

[5] This is more demanding computationally; on the other hand, $\sqrt[n]{x^n + y^n} \approx max(x, y)$ if n is large.

6.6 Design Example 3

6.7 References

[La] Brett Larson. Siso Robust Controller Design via the H^∞ Method. Master Thesis, under the direction of Prof. F. Bailey. University of Minnessota, April 1989.

[HMer] J. W. Helton and O. Merino.

[BB] Stephen P. Boyd and Craig P. Barrat, *Linear Controller Design*, Prentice Hall, 1991.

[FPE] G. F. Franklin, J. D. Powell and A. Emami-Naeini. *Feedback Control of Dynamic Systems.* Addison-Wesley, 1986.

Chapter 7

Time domain requirements

This chapter discusses time domain requirements in more detail than was discussed in Chapter 5 (Section 7.1), and presents mathematical results on translating these requirements to the frequency domain (Section 7.2). These results are used in a solved example (Section 7.3).

7.1 Requirements on the step response function

In the time domain, the performance of systems is usually described in terms of output functions, produced by specific input functions. Typical input functions used for this are the unit impulse, unit step, ramp, and sinusoidal functions. We concentrate our study on the step response function $c(t)$, that is, the output produced when the input is the unit step function.

7.1.1 Step response indices

A step response function $c(t)$ is usually described in terms of a set of numbers, or *indices*, that depend on the function. These include the overshoot, undershoot, settling time, and rise time [1]. The definitions below are taken

---

[1] Another common index is the time delay. It is usually defined as the time it takes the step response curve $c(t)$ to reach 0.5. This definition is ambiguous unless $c(t)$ is a priori known to be extremely well behaved.

from [BB, Chapter 8]. Here $c(t)$ represents the step response function of a *given* system $\mathcal{S}$.

- **Overshoot**: $M_{os} = \sup_{t \geq 0} c(t) - 1$.
- **Undershoot**:$M_{us} = \inf_{t \geq 0} c(t)$.
- **Rise time**: $t_{rise} = \inf\{t_0 : c(t) \geq 0.8 \ for \ t \geq t_0\}$.
- **Settling time**: $t_{settle} = \inf\{t_0 : |c(t) - 1| \leq 0.05 \ for \ t \geq t_0\}$.

The indices above are depicted in Fig 7.1.

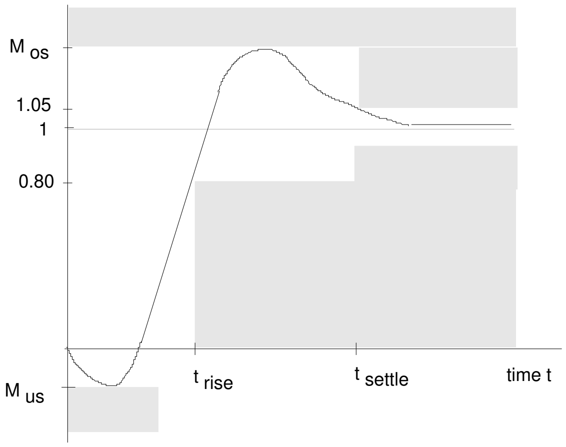


Figure 7.1: The step response function indices

7.1.2 Time domain performance requirements

The time domain indices listed above give information about a particular curve $c(t)$. To express goals of the design process in the time domain we need the definition below.

Definition 2 *A time domain performance requirement on the step response function $c(t)$ is an inequality involving $c(t)$ and an interval on which the inequality is defined.*

Time domain performance requirements for other functions, such as the impulse response, ramp response, etc., are defined in similar fashion. Observe that time domain performance requirement and index are two different concepts; while the performance requirement is an inequality stated *in advance* of choosing a system, the performance index applies to a *given* system.

7.1.3 Classical time domain requirements

We now give a set of time domain performance requirements, which corresponds to the indices typically used.

Overshoot requirement: Given $M > 0$,

$$c(t) - 1 < M, \ \forall t > 0 : \tag{7.1}$$

Undershoot requirement: Given $M > 0$,

$$c(t) > M, \ \forall t > 0 : \tag{7.2}$$

Rise time requirement: Given $t_r > 0$, $\Delta_r > 0$,

$$c(t) > \Delta_r, \ \forall t > t_r : \tag{7.3}$$

Settling time requirement: Given $t_s > 0$ and $\Delta_s > 0$,

$$|c(t) - 1| < \Delta_s, \qquad \forall t > t_s \tag{7.4}$$

7.1.4 Envelope-type requirements

A mathematically natural way to establish requirements on $c(t)$ is to give a set of curves that act as bounds for $c(t)$.

General step response envelope requirement: Given functions of time $g_1(t)$ and $g_2(t)$,

$$g_1(t) \leq c(t) \leq g_2(t) \ (\forall t \geq 0) \tag{7.5}$$

We refer to g_1 and g_2 as bounds for $c(t)$. See Fig. 7.2.

Another constraint can be defined on the slope of the curve $c(t)$. In fact the rise time can be used to give an indirect measure of this slope. To restrict the behavior of $c'(t)$, functions g_3 and g_4 can be chosen to require that

$$g_3(t) \leq c'(t) \leq g_4(t) \tag{7.6}$$

The corresponding picture is given in Fig. 7.3.

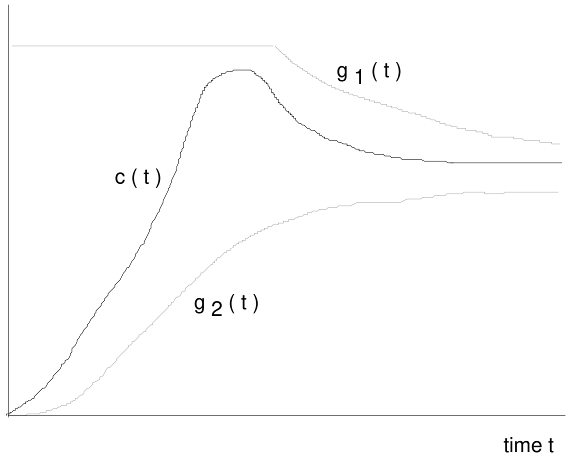


Figure 7.2: Bounds for $c(t)$.

7.2 Translation of time-domain requirements

In this section we look at the question of how the inequality $f(t) \geq 0$ translates to the frequency domain.

7.2.1 Completely monotonic functions

Let F be the Laplace transform of a function of time $f(t)$:

$$F(s) = \mathcal{L}[f](s) = \int_0^\infty e^{-st} f(t) dt \tag{7.7}$$

Below we find criteria that indicate when the function $F(s)$ arises from a nonnegative function $f(t)$.

Suppose that the function f in (7.7) satisfies $f(t) \geq 0$. To gain some insight, we formally differentiate both sides of (7.7). By differentiating under the integral sign we obtain

$$F^{(n)}(s) = \int_0^\infty (-t)^n e^{-st} f(t) dt \tag{7.8}$$

Thus for s any positive real number,

$$(-1)^n F^{(n)}(s) = \int_0^\infty t^n e^{-st} f(t) dt \geq 0 \tag{7.9}$$

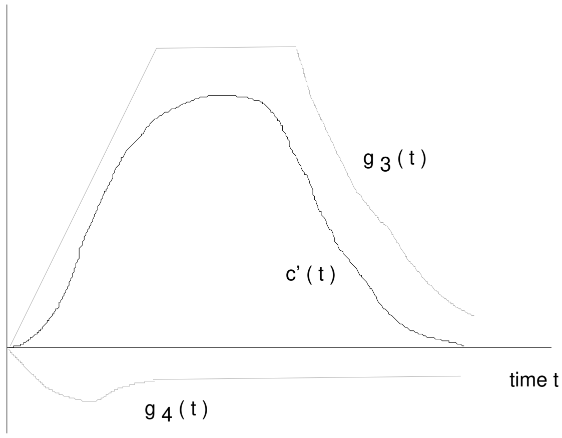


Figure 7.3: Bounds for $c'(t)$.

Theorem 2 formalizes this and gives a converse statement.

Theorem 2 *Let* $F \in \mathcal{RH}^\infty$ *be a strictly proper function, and let* $f : (0, \infty) \rightarrow R$ *be such that* $\mathcal{L}[f](s) = F(s)$.

(i) If $f(t) \geq 0$ *for all* $t \geq 0$, *then*

$$(-1)^n F^{(n)}(s) \geq 0 \ (\forall n \in N) \ (\forall s > 0) \tag{7.10}$$

(ii) If there exists a real number $s_0 > 0$ *such that*

$$(-1)^n F^{(n)}(s_0) \geq 0, \tag{7.11}$$

then $f(t) \geq 0$ *for all* $t \geq 0$.

Definition 3 *A function* $F(s)$ *is completely monotonic if it is continuous on* $0 \leq s < \infty$, *has derivatives of all orders on* $0 < s < \infty$, *and satisfies (7.10).*

Thus Theorem 2 says that strictly proper $\mathcal{RH}^\infty$ functions F that are completely monotonic are precisely those that are the Laplace transform of nonnegative functions. Furthermore, to check that F is completely monotonic it suffices to verify (7.10) at only one point $s = s_0$. Theorem 2, in more general form, is due to D. V. Widder and to S. Bernstein (circa 1930). See [W] for details.

Example 1. The function $F(s) = 1/(s+1)^3$ is a strictly proper $\mathcal{RH}^\infty$ function, thus it has an inverse transform $f(t) = \mathcal{L}[F](t)$. Since

$$F^{(n)}(1) = (-1)^n \frac{(n+2)!}{2^{n+4}}$$

it follows from Theorem 2 that the function F is completely monotonic, and that $f(t)$ is nonnegative.

7.2.2 Known boundary values

Let F be a strictly proper $\mathcal{RH}^\infty$ function, and suppose that its values $F(j\omega)$ on the $j\omega$ axis are known. The question posed now is, when is F the Laplace transform of a nonnegative function $f(t)$? Theorem 3 below gives a partial answer to this question.

We need the following notation. For $F \in \mathcal{RH}^\infty$ and $\omega \in R$ set

$$\Delta_n(F;\omega) = \mathbf{Det}\begin{pmatrix} F(0) & F(j\omega) & \cdots & F(j(n-1)\omega) \\ \overline{F(j\omega)} & F(0) & \cdots & F(j(n-2)\omega) \\ \overline{F(j2w)} & \overline{F(j\omega)} & \cdots & F(j(n-3)\omega) \\ \vdots & \vdots & & \vdots \\ \overline{F(j(n-1)\omega)} & \overline{F(j(n-2)\omega)} & \cdots & F(0) \end{pmatrix} \tag{7.12}$$

Let us compute the determinant (7.12) for $n = 2$.

$$\Delta_2(F;\omega) = \mathbf{Det}\begin{pmatrix} F(0) & F(j\omega) \\ \overline{F(j\omega)} & F(0) \end{pmatrix} = F(0)^2 - |F(j\omega)|^2. \tag{7.13}$$

Some calculations give the following formula for $n = 3$.

$$\Delta_3(F;\omega) = (F(0)^2 - |F(jw)|^2)^2 - |F(jw)^2 - F(0)F(j2w)|^2. \tag{7.14}$$

Theorem 3 *Let F be a strictly proper $\mathcal{RH}^\infty$ function, and let $f = \mathcal{L}^{-1}[F]$. If the function f in nonnegative then $\Delta_n(F;\omega) \geq 0, \forall \omega \in R, \forall n \in N$.*

For emphasis we state Theorem 3 above in a different form. If there exist n and ω such that $\Delta_n(F,\omega) < 0$, then $\mathcal{L}^{-1}[F](t)$ changes sign at least once on the interval $0 < t < \infty$.

7.2.3 Application to system design

Consider the closed loop system $\mathcal{S}$ given in Fig. 1.1. The step response curve $c(t)$ has as its Laplace transform the function

$$\mathcal{L}[c](s) = T(s)/s \tag{7.15}$$

From the assumption that the functions c and c' are both of exponential type so that their transforms are well defined, and from the properties of Laplace transform we have

$$\mathcal{L}[\frac{dc}{dt}](s) = T(s) \tag{7.16}$$

Now it is possible to characterize in terms of the closed loop transfer function T all systems $\mathcal{S}$ with non-decreasing step response function.

Theorem 4 *If the system $\mathcal{S}$ has a strictly proper closed loop transfer function $T \in \mathcal{RH}^{\infty}$, then the step response curve $c(t)$ is nondecreasing if and only if the function T is completely monotonic.*

Proof. The function $c(t)$ is nondecreasing if and only if its derivative $c'(t)$ is nonnegative for all t, which in turn holds if and only if T is completely monotonic. ■

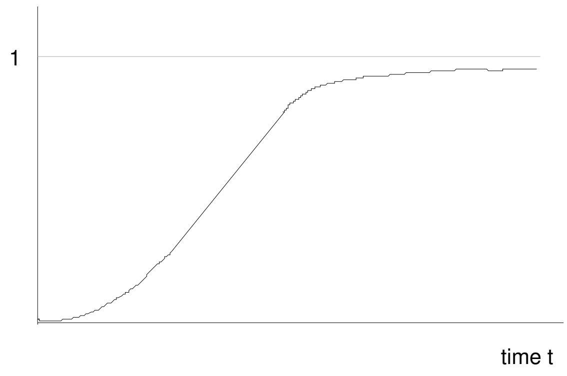


Figure 7.4: A nondecreasing step response function.

Time domain performance requirements on the step response, such as

$$c(t) \leq g(t), \qquad t \geq 0,$$

can be translated to the frequency domain as the statement,

$$G(s) - \frac{T(s)}{s} \text{ is a completely monotonic function.} \qquad (7.17)$$

Note that there is a singularity at the point $s = 0$ in the expression (7.17). However, this is a removable singularity under the appropriate hypotheses.

Theorem 5 *Let $\mathcal{S}$ be an internally stable system with plant $P(s)$ having a pole at $s = 0$. Let $g(t)$ be a function whose Laplace transform is of the form $G(s) = 1/s + G_0(s)$, with $G_0(s) \in \mathcal{RH}^\infty$. Then the function $G(s) - \frac{T(s)}{s}$ is in $\mathcal{RH}^\infty$, and the inequality $c(t) \leq g(t)$ holds for all $t \geq 0$ if and only if $G(s) - \frac{T(s)}{s}$ is completely monotonic.*

Proof. Since $G(s) - \frac{T(s)}{s}$ is the Laplace transform of $g(t) - c(t)$, the only thing to prove is the statement $G(s) - \frac{T(s)}{s}$ is in $\mathcal{RH}^\infty$. Note that $s = 0$ is the only possible source of trouble. By internal stability and since there is a pole of P at $s = 0$, the closed loop transfer function T satisfies the interpolation condition $T(0) = 1$. This says that $\frac{(1-T(s))}{s}$ is in $\mathcal{RH}^\infty$. The same is true then for the function

$$G(s) - \frac{T(s)}{s} = G_0(s) - \frac{T(s) - 1}{s}.$$

■

An application of Theorem 5 and Theorem 3 is presented next in Section 7.3.

7.3 Design Example 4

We now use the results given in this chapter to improve upon the answer obtained in Section 4 of Chapter 5.

7.3.1 Statement of the problem

Solve *Design* for the plant

$$P(s) = \frac{0.036(s + 25)}{s^2(s + 0.02 + 2j)(s + 0.02 - 2j)}$$

The step response function has requirements step response overshoot

$$c(t) < 1.15 \qquad (\forall t \geq 0)$$

and settling time

$$|c(t) - 1| < 0.01, \qquad (t \geq 14.0)$$

We need to find a closed loop system $\mathcal{S}$ that is internally stable and meets the performance requirements.

7.3.2 Strategy for defining the performance function

The optimal system obtained in Subsection 6.4 does not satisfy the settling time requirement. Now the performance function is redefined to obtain an improved solution. To do this, we set the performance to be of the form

$$\Gamma'(\omega, z) = W_1(j\omega)\Gamma_1(\omega, z) + W_2(j\omega)\Gamma_2(\omega, z) \tag{7.18}$$

where Γ_1 is the performance function in Subsection 6.4. The function Γ_2 and the weight functions W_1 and W_2 are selected in the following way:

a. Choose a function g for the step response envelope requirement

$$c(t) \leq g(t), \qquad \forall t \geq 0 \tag{7.19}$$

The idea is to force the settling time and overshoot requirements upon our design by the choice of a suitable function g.

b. The envelope constraint (7.19) is replaced with a (weaker) statement in the frequency domain of the form

$$|G(j\omega) - T(j\omega)/(j\omega)|^2 \leq \gamma_0^2 \quad \forall \omega \tag{7.20}$$

where $\mathcal{L}[g] = G$ (based on Theorems 3 and 5). Call $\Gamma_2(\omega, T(j\omega))$ the left-hand-side of (7.20).

c. Substitute the optimal T_1^* obtained in Subsection 6.4 into (7.20). Determine frequency bands where (7.20) is not satisfied. Then choose W_1 and W_2 to enhance Γ_2 at these frequency bands.

7.3.3 Generating an envelope requirement

The function g must have a maximum value of 1.15 approximately, and it must satisfy $g(14.0) \approx 0.01$. One way to do this is to choose nonnegative parameters α_1, α_2, α_3 and κ in

$$g(t) = 1 + \alpha_1\, e^{-\kappa t} + \alpha_2\, t\, e^{-\kappa t} + \alpha_3\, t^2\, e^{-\kappa t} + \alpha_4\, t^3\, e^{-\kappa t}$$

After some experimentation we obtain the following bound for the step response:

$$g(t) = 1 + 0.05\, t\; e^{-0.65\, t} + 0.005\, t^2\; e^{-0.65\, t} + 0.02\, t^3\; e^{-0.65\, t} \tag{7.21}$$

Fig. 7.5 shows a plot of the function g(t). Thus we expect to satisfy the requirements on the step response and settling time by requiring (7.19) to hold for the step response function and the function g in (7.21).

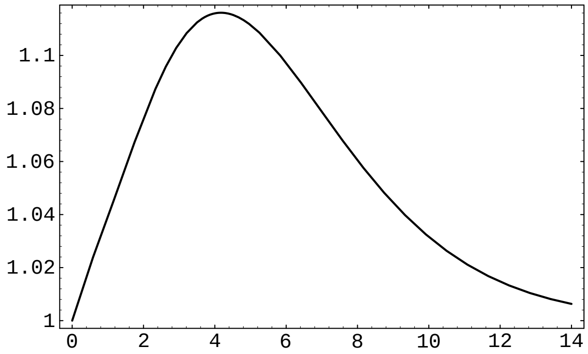


Figure 7.5: Bound g for envelope requirement in Design Example 4.

7.3.4 Weak form of requirements envelope

The internal stability requirement on the system $\mathcal{S}$ and the order 2 pole of the plant at $s = 0$ imply that designable closed loop transfer functions satisfy

$$\lim_{s \to 0} \frac{1 - T(s)}{s} = 0 \tag{7.22}$$

Then the function

$$\mathcal{L}[g - c](s) = \frac{1}{s} + \frac{0.05}{(s + 0.65)^2} + 2\frac{0.005}{(s + 0.65)^3} + 6\frac{0.02}{(s + 0.65)^4} - \frac{T(s)}{s} \tag{7.23}$$

is in $\mathcal{RH}^\infty$, and the value of this function at $s = 0$ is

$$\mathcal{L}[g - c](0) = \frac{0.05}{(0.65)^2} + 2\frac{0.005}{(0.65)^3} + 6\frac{0.02}{(0.65)^4} =: \gamma_0 \approx 0.827 \tag{7.24}$$

The function $g - c$ is nonnegative if and only if $\mathcal{L}[g - c](s)$ is completely monotonic, by Theorem 5. A necessary condition for this, by Theorem 3,

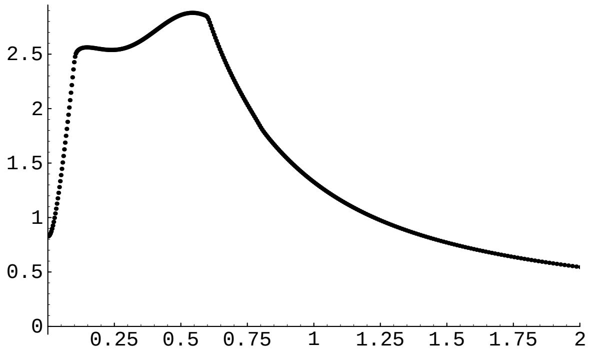


Figure 7.6: Plot of $\Gamma_2(\omega, T_1^*(j\omega))$.

is that at all frequencies ω,

$$\left| \frac{1}{j\omega} + \frac{0.05}{(j\omega + 0.65)^2} + 2\frac{0.005}{(j\omega + 0.65)^3} + 6\frac{0.02}{(j\omega + 0.65)^4} - \frac{T(j\omega)}{j\omega} \right|^2 \leq \gamma_0^2 \tag{7.25}$$

Now set $\Gamma_2(\omega, T(j\omega))$ to be the left-hand-side of (7.25).

7.3.5 Choosing weight functions

To choose W_2 we look at the plot of $\Gamma_2(\omega, T_1^*(j\omega))$ depicted in Fig. 7.6. It is clear from this figure that inequality 7.25 is false over the band $0 < \omega < 1.25$. This implies that Γ_2 should be emphasized on this frequency band. Note that for any choice of strictly proper T the function values $\Gamma_2(\omega, T(j\omega))$ become very small at high frequency. We select constant weights

$$W_1(j\omega) = 1 \qquad (\forall \omega) \tag{7.26}$$

and

$$W_2(j\omega) = 9 \qquad (\forall \omega) \tag{7.27}$$

7.3.6 Results of the optimization

A computer run[2] produces an optimal closed loop T with Bode plots shown in Fig. 7.7. The following is a rational approximation to T^*:

$$\begin{aligned} T_1(s) \quad = \quad & \frac{5.52947+855.26s+4867.54s^2+7480.54s^3+\cdots}{5.52947+855.26s+5303.79s^2+13156.5s^3+17071.8s^4+\cdots} \cdots \\ & \frac{+4972.77s^4+1303.81s^5+197.695s^6+5.26629s^7}{+12858.6s^5+5832.42s^6+1582.24s^7+246.287s^8+21.0034s^9+1.\,s^{10}} \end{aligned} \tag{7.28}$$

We can see in Fig. 7.8 that the rational function T_1 satisfies $\max_\omega |T^*(j\omega) - T_1(j\omega)| < 0.02$. The compensator associated with T_1 is

$$\begin{aligned} C(s) \quad = \quad & \frac{146.286(s+0.02\pm j\,1.)(s+30.3597)(s+2.50175\pm j\,3.28011)\cdots}{(s+25.)(s+4.8979\pm j\,7.8766)(s+4.62512)\cdots} \\ & \frac{\cdots(s+0.956961\pm j\,0.515942)(s+0.255874)(s+0.00671959)}{\cdots(s+2.25488\pm j\,2.03546)(s+0.989326\pm j\,0.532551)(s+0.0941209)} \end{aligned}$$

The plot of the step response curve associated with T_1 (or with T^*) shown in Fig. 7.9 shows that the original design goals are satisfied.

7.4 References

[BB] Stephen P. Boyd and Craig P. Barrat, *Linear Controller Design*, Prentice Hall, 1991.

[W] D. V. Widder. The Laplace Transform. Princeton University Press. 1941.

---

[2] with the software package OPTDesign.

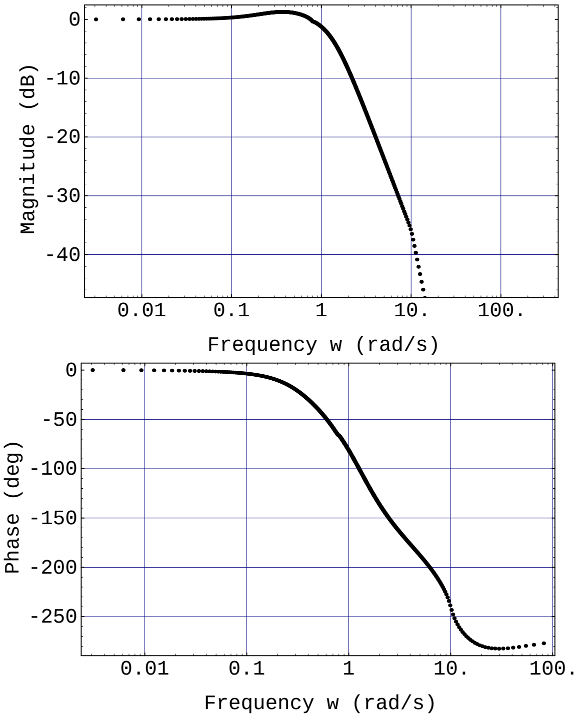


Figure 7.7: Bode plots of optimal closed loop T^* in Design Example 4.

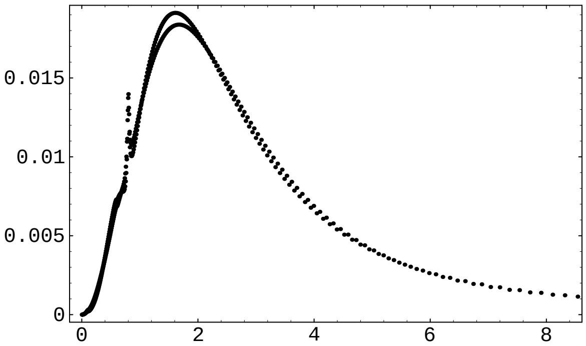


Figure 7.8: Plot of $|T^*(j\omega) - T_1(j\omega)|$.

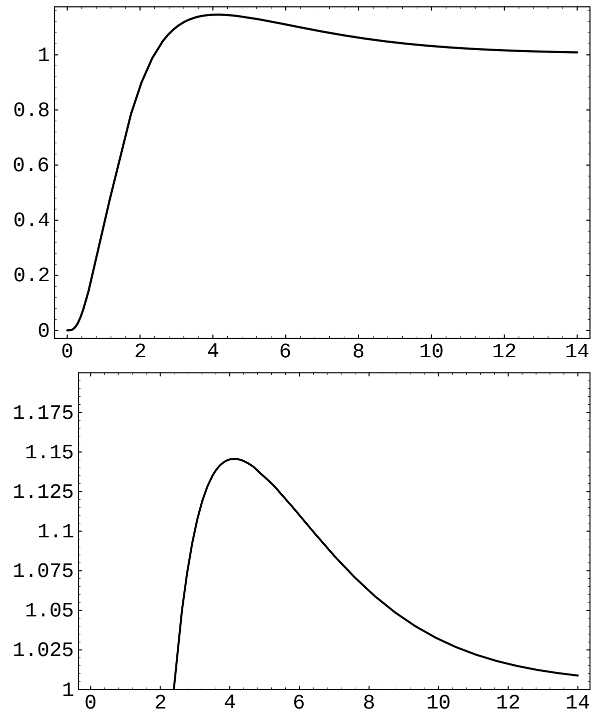


Figure 7.9: Step response function corresponding to T_1.

Chapter 8

Internal Stability II

Internal Stability was introduced in Chapter 2. This chapter continues the discussion to obtain theorems that precisely characterize internally stable systems, either in terms of zeros and poles of the plant or in terms of a formula for the closed loop transfer function. Section 8.1 develops the mathematical tools for interpolation with rational functions. In Section 8.2 a characterization of internally stable systems is given in terms of interpolation conditions on the closed loop transfer function, when the plant has simple RHP zeros and poles. The case of higher multiplicity is treated in Section 8.3.

8.1 Calculating interpolants

Systems $\mathcal{S}$ whose plant has RHP poles or RHP zeros have closed loop transfer functions that can be described by a formula. We shall see that this formula depends on those RHP poles and RHP zeros of the plant and nothing else. To derive the formula and related results, we concentrate first on the case where all of these zeros and poles have multiplicity one (called *simple* poles and zeros).

Suppose that the following sets of complex numbers are given:

$$s_1, \ldots, s_n \in RHP \quad \text{(data points)} \tag{8.1}$$

and

$$v_1, \ldots, v_n \in C \quad \text{(data values)}\ . \tag{8.2}$$

For $T(s)$ a rational function, consider the set of equalities

$$(INT) \qquad T(s_1) = v_1,\ T(s_2) = v_2,\ \ldots,\ T(s_n) = v_n$$

called an *interpolation condition on* T. If $T \in \mathcal{RH}^\infty$ satisfies *INT*, then T is called an *interpolant* (for the data given in (8.1) and (8.2)).

The main objective of this section is to present ways to generate interpolants. More precisely, given a set of equalities *INT* we show how to

- Produce one interpolant.
- Produce a parameterization (formula) of all interpolants.

First, we present an elementary way to produce one interpolant (Subsection 8.1.1) and to parameterize all interpolants (8.1.2). This is then generalized to problems with an interpolation condition at infinity (Subsection 8.1.3).

Interpolation with polynomials, rather than rational functions, is a standard subject in numerical analysis (see [DC]). The treatment here for rational functions is conceptually the same, and proofs are similar.

8.1.1 Calculating one interpolant

Let *INT* and a real number $a < 0$ be given. We will show how to produce an interpolant T in $\mathcal{RH}^\infty$ such that all its poles are located at $s = a$.

Let T be a proper rational function of s with a single pole location at $s = a$. One can think of T as being a sum of rational functions whose denominators are of the form $(s-a)^k$, with $k = 0, \ldots, n-1$. Since T is proper, for each factor $(s - s_0)$ that appears in a numerator there is a corresponding factor $(s-a)$ in the denominator. Thus T must have the form

$$T(s) = c_0 + c_1 \frac{s-s_1}{s-a} + c_2 \frac{(s-s_1)(s-s_2)}{(s-a)^2} + \cdots + c_{n-1} \frac{(s-s_1)\cdots(s-s_{n-1})}{(s-a)^{n-1}} \tag{8.3}$$

for some constants $c_0,\ c_1,\ \cdots,\ c_{n-1}$. The right-hand-side of (8.3) is called *Newton's representation for the function* $T(s)$.

Now we seek $T \in \mathcal{RH}^\infty$ with the form (8.3) that satisfies *INT*. Set $s = s_1,\ s = s_2,\ \ldots,\ s = s_n$ in (8.3) and combine with *INT* to obtain the system of equations

$$\begin{array}{rl} c_0 & = v_1 \\ c_0 + c_1 \frac{s_2 - s_1}{s_2 - a} \cdots & = v_2 \\ & \vdots \\ c_0 + \cdots + c_{n-1} \frac{(s_n - s_1)\cdots(s_n - s_{n-1})}{(s_n - a)^{n-1}} & = v_n \end{array} \tag{8.4}$$

This system of equations in the unknown c_k's has a unique solution. It can be solved easily by back substitution.

Example. Find $T \in \mathcal{RH}^\infty$ with $a = -1$ as its only pole location, such that

$$T(0) = 1, \qquad T(1) = -2 \tag{8.5}$$

Solution. Note that $n = 2$ (number of points in *INT*), so Newton's representation for an interpolant is

$$T(s) = c_0 + c_1 \frac{s}{s+1} \tag{8.6}$$

Combining (8.5) and (8.6) we see that (8.3) becomes

$$\begin{aligned} c_0 &= v_1 \\ c_0 + c_1 \frac{1}{2} &= v_2 \end{aligned} \tag{8.7}$$

and we obtain $c_0 = 1$ and $c_1 = -6$.

A special situation occurs when the interpolation values $v_1, \ldots, v_n$ are all zero. In this case, system (8.4) gives $c_0 = c_1 = \ldots = c_{n-1} = 0$, so the function T given by Newton's representation method is the zero function. A nontrivial interpolant for this case is easily obtained by setting

$$T_0(s) = \gamma \frac{(s - s_1) \cdots (s - s_n)}{(s + a)^n} \tag{8.8}$$

where γ is any nonzero real constant. Observe that the function T_0 in (8.8) has no zeros in the *RHP* other than $s_1, \ldots, s_n$.

8.1.2 Parameterization of all interpolants

Given a set *INT* of interpolation conditions, by INT_0 we denote the set of homogeneous interpolation conditions

$$INT_0 \qquad T(s_1) = 0,\ T(s_2) = 0,\ \ldots,\ T(s_m) = 0$$

obtained from *INT* by setting the right hand sides to 0. The set of conditions INT_0 plays a fundamental role in parameterizing all functions $T \in \mathcal{RH}^\infty$ that satisfy *INT*.

Theorem 6 *Let a set INT of interpolating conditions be given. Also given are functions T_1, T_0 in $\mathcal{RH}^\infty$ such that T_1 satisfies INT, T_0 satisfies INT_0, $s_0, \ldots, s_n$ are the only zeros of T_0, and T_0 has relative degree 0.*

Then every T in $\mathcal{RH}^\infty$ that satisfies INT has the form

$$T(s) = T_1(s) + T_0(s)F(s) \tag{8.9}$$

for some F in $\mathcal{RH}^\infty$. Conversely, if a function T is given by the formula (8.9), then T belongs to $\mathcal{RH}^\infty$ and satisfies INT.

Proof. Let T be a function in $\mathcal{RH}^\infty$ that satisfies INT. Then the difference $T(s) - T_1(s)$ has zeros at the locations $s = s_1, \ldots, s = s_n$. Therefore the rational function $F(s) = (T(s) - T_1(s))/T_0(s)$ has no poles in the closed RHP, and is proper since T_0 has relative degree 0. This says that $F(s)$ belongs to $\mathcal{RH}^\infty$, and we have proved the first part of the theorem. To prove the second part, suppose that T in $\mathcal{RH}^\infty$ satisfies (8.9). By direct substitution of the values $s = s_0, s_1, \ldots, s_n$ in the formula (8.9) we see that T satisfies INT. ∎

It follows from Theorem 6 that it is enough to find particular functions T_0 and T_1, as in formula (8.9), to determine all $\mathcal{RH}^\infty$ functions that satisfy *INT*.

Example. We will find a formula for all functions $T \in \mathcal{RH}^\infty$ that satisfy

$$T(1) = 2, \quad T(3) = 4 \tag{8.10}$$

We can choose any negative number as the location of the pole of the rational functions T_0 and T_1 in formula (8.9). We select $a = -2$;

$$T_0(s) = \frac{(s-1)(s-3)}{(s+2)^2} \tag{8.11}$$

and

$$T_1(s) = c_0 + c_1\frac{s-1}{s+2} \tag{8.12}$$

By taking $s = 1$ and $s = 3$ in (8.12) and combining with (8.10), we obtain $c_0 = 2$ and $c_1 = 5$. We know from Theorem 1 that all functions $T \in \mathcal{RH}^\infty$ that satisfy (8.10) are of the form

$$T(s) = 2 + 5\frac{s-1}{s+1} + \frac{(s-1)(s-3)}{(s+1)^2}F(s) \tag{8.13}$$

where $F \in \mathcal{RH}^\infty$.

8.1.3 Interpolation with a relative degree condition

If a rational function T is strictly proper, then as $|s|$ gets large, $|T(s)|$ approaches 0. One can think of T as having a zero at $s = \infty$. The order we associate with this "zero" is the same number as the *relative degree of* T, $d(T)$ (the difference of degrees between denominator and numerator). One way to incorporate this special zero in interpolation problems is to specify the relative degree of the interpolant together with INT.

Given INT and an integer $r \geq 0$, we seek T in $\mathcal{RH}^\infty$ with relative degree $d(T) = r$ that satisfies INT. To produce one interpolant, modify the representation (8.3) to include the requirement on the relative degree of T:

$$\begin{aligned} T(s) = \ & \tfrac{1}{(s-a)^r}\Big(c_0 + c_1 \tfrac{s-s_1}{s-a} + c_2 \tfrac{(s-s_1)(s-s_2)}{(s-a)^2} + \cdots \\ & \cdots + c_{n-1} \tfrac{(s-s_1)\cdots(s-s_{n-1})}{(s-a)^{n-1}}\Big) \end{aligned} \qquad (8.14)$$

To find $c_0, c_1, \ldots, c_{n-1}$, proceed as before by forming a triangular system of equations and solving it.

A parameterization of all interpolants that satisfy both INT and the relative degree condition $d(T) = r$ is obtained after a small modification to Theorem 6. Details of the proof of Theorem 7 below are left to the reader.

Theorem 7 *Assume the hypotheses of Theorem* 1*, with the additional assumptions that* $d(T_1) \leq r$ *and* $d(T_0) = r$. *Then a rational function* T *is a function in* $\mathcal{RH}^\infty$ *that has relative degree* r *and satisfies* INT *if and only if there exists a rational* $F \in \mathcal{RH}^\infty$ *such that*

$$T(s) = T_1(s) + T_0(s)F(s) \qquad (8.15)$$

8.2 Plants with simple RHP zeros and poles

Internal stability can be expressed as the statement that certain functions associated with $\mathcal{S}$ are bounded and have no poles in the closed RHP. This fact is presented below as a lemma needed for the main theorems of this section, but it is interesting in itself. In fact, it can be used as an alternative definition of internal stability that generalizes to MIMO systems (see [BB], [FDT], and [Fr]).

Lemma 1 *The closed loop system of Figure* 2.1 *is internally stable if and only if*

$$Q,\ T,\ S,\ SP \quad \textit{belong to} \quad \mathcal{RH}^\infty \qquad (8.16)$$

Proof. Key relations are $S = 1 - T$ and $T = QP = PQ$. These imply that T is in $\mathcal{RH}^\infty$ if and only if S and $PQ = QP$ are in $\mathcal{RH}^\infty$. If the system is internally stable, we only have to check that $Q = CS$ and SP belong to $\mathcal{RH}^\infty$. Note that

$$S(s_p) = \frac{1}{(1+PC)}(s_p) = 0 \tag{8.17}$$

at the poles s_p of C. Thus $Q = CS$ is in $\mathcal{RH}^\infty$. Similar reasoning implies that

$$SP = S\frac{1}{1+PC} \tag{8.18}$$

is in $\mathcal{RH}^\infty$. This proves one half of the lemma. Now suppose that the relations in (8.16) hold. If there is pole-zero cancellation in PC at a point $s = s_0$ in the *RHP*, then either $Q = C/(1+PC)$ or $SP = P/(1+PC)$ has a pole at $s = s_0$, both of which contradict the hypothesis. ■

To illustrate the use of this lemma, we verify (8.16) for the system $\mathcal{S}_1$ with plant $P(s) = 1/(s-1)$ and compensator $C_1(s) = (s-1)/(s+1)$. In Section 2.3 we saw that $T_1(s) = 1/(s+2)$, which belongs to $\mathcal{RH}^\infty$. We also have that

$$P(s)S_1(s) = P(s)(1 - T_1(s)) = \frac{1}{s-1}\left(1 - \frac{1}{s+2}\right) = \frac{s+1}{(s-1)(s+2)} \tag{8.19}$$

We see that the closed loop plant PS_1 has a pole at $s = 1$. By Lemma 1 we can assert that the system $\mathcal{S}_1$ is not internally stable.

Now consider the system $\mathcal{S}_2$ with the same plant and compensator $C_2(s) = 2$. Recall that $T_2(s) = 2/(s+1)$, so

$$\begin{array}{lll} P(s)S_2(s) & = \frac{1}{s-1}\frac{s-1}{s+1} & = \frac{1}{s+1} \\ Q_2(s) & = C_2(s)S_2(s) & = 2\frac{s-1}{s+1} \end{array} \tag{8.20}$$

Therefore (8.16) is satisfied, and the system $\mathcal{S}_2$ is internally stable.

The description of internal stability in terms of interpolation conditions is given in the following theorem.

Theorem 8 *Consider the closed loop system $\mathcal{S}$, where the plant P is given. Let ρ_ℓ $(\ell = 1, \ldots, n)$ denote the poles and z_ℓ $(\ell = 1, \ldots, m)$ denote the zeros of the plant P in the closed* RHP. *Suppose that all unstable zeros and poles of the plant are simple. If $\mathcal{S}$ is internally stable, then there exists an integer $d_c \geq 0$ such that the closed loop transfer function $T = PC(1+PC)^{-1}$ must satisfy interpolation conditions*

$$\left\{ \begin{array}{ll} T(\rho_\ell) = 1 & (\ell = 1, \ldots, n) \\ T(z_\ell) = 0 & (\ell = 1, \ldots, m) \\ d(T) = d(P) + d_c & \end{array} \right. \tag{8.21}$$

Conversely, if T *is any function* $\mathcal{RH}^\infty$ *satisfying (8.21) for some* $d_c \geq 0$*, then the closed loop system* $\mathcal{S}$ *associated with* T *and* P *is internally stable.*

Proof. Suppose the system $\mathcal{S}$ is internally stable. Then $P^{-1}T = C(1+PC)^{-1} = Q \in \mathcal{RH}^\infty$. Clearly, if $P(z_\ell) = 0$ then z_ℓ is a pole of P^{-1}. Thus $T(z_\ell) = 0$. Also $SP = (1+PC)^{-1}P \in \mathcal{RH}^\infty$, so poles of P are zeros of $(1+PC)^{-1} = 1-T$. Then $T(\rho_\ell) = 1$. We now set $d_c = d(C)$ in equation (8.21) to finish the first part of the proof.

Now suppose that $T \in \mathcal{RH}^\infty$ satisfies (8.21). The relation $d(T)-d(P) = d_c \geq 0$ combined with (8.21) insures that the compensator is proper. This fact, and both P and T having the same RHP zeros imply that $Q = TP^{-1}$ is proper and has no RHP poles, i.e., $P^{-1}T = Q \in \mathcal{RH}^\infty$. Also $S = 1-T \in \mathcal{RH}^\infty$. Again, (8.21) says that poles of P are zeros of $1-T$, thus $(1-T)P = SP \in \mathcal{RH}^\infty$. Therefore $\mathcal{S}$ is internally stable. ∎

Theorem 8 can be combined with Theorem 7 to obtain a formula describing all possible internally stable systems $\mathcal{S}$ associated with a fixed plant. The proof of Theorem 9 is left to the reader.

Theorem 9 *Let* P *be a proper real rational function, with simple RHP zeros* $z_1, \ldots, z_n$ *and RHP poles* $\rho_1, \ldots, \rho_m$*, and let* d_c *be a nonnegative integer. Let* T_0*,* T_1 *be* $\mathcal{RH}^\infty$ *functions such that*

$$\left\{ \begin{array}{ll} T_1(\rho_\ell) = 1, & (\ell = 1, \ldots, n) \\ T_1(z_\ell) = 0, & (\ell = 1, \ldots, m) \\ d(T_1) \geq d(P) + d_c & \end{array} \right. \tag{8.22}$$

and

$$\left\{ \begin{array}{ll} T_0(\rho_\ell) = 0, & (\ell = 1, \ldots, n) \\ T_0(z_\ell) = 0, & (\ell = 1, \ldots, m) \\ d(T_0) = d(P) + d_c & \end{array} \right. \tag{8.23}$$

and suppose that all the RHP zeros of T_0 *are listed in (8.23).*

If $T \in \mathcal{RH}^\infty$ *is the closed loop transfer function of an internally stable system* $\mathcal{S}$ *with plant* P *and* $d(C) = d_c$*, then there exist* $H \in \mathcal{RH}^\infty$ *such that*

$$T(s) = T_1(s) + T_0(s)H(s) \tag{8.24}$$

Conversely, if (8.24) holds for some $H \in \mathcal{RH}^\infty$*, then the system* $\mathcal{S}$ *associated with* P *and* T *is internally stable, and the degree of its compensator is* $d(C) = d_c$*.*

Example. Consider the plant $P(s) = (s-1)/(s-3)$, and set $d_c = 0$. We now find a formula for all T that yields an internally stable system $\mathcal{S}$ with degree 0 compensator and plant $P(s) = (s-1)/(s-3)$.

The plant P has RHP zero $s = 1$ and RHP pole $s = 3$. Set $T_1(s) = 2(s-1)/(s+1)$ and $T_0(s) = (s-1)(s-3)/(s+1)^2$. Then T_1 and T_0 satisfy (8.22) and (8.23) respectively, and by Theorem 9 we have

$$T(s) = \frac{2(s-1)}{s+1} + \frac{(s-1)(s-3)}{(s+1)^2}H(s), \ (H \in \mathcal{H}^\infty).$$

8.3 Parameterization: the general case

In this section, the results already obtained on parameterization of internally stable systems is generalized to the case where the plant may have RHP zeros or poles with multiplicity greater than 1. Results in this section are stated without proof.

8.3.1 Higher order interpolation

The interpolation condition we now consider specifies values of the derivatives of the function at selected points.

$$INT_h \begin{cases} T(s_1) = v_{1,1}, & T^{(1)}(s_1) = v_{1,2} \quad \ldots, & T^{(m_1-1)}(s_1) = v_{1,m_1} \\ T(s_2) = v_{2,1}, & T^{(1)}(s_2) = v_{2,2} \quad \ldots, & T^{(m_2-1)}(s_2) = v_{2,m_2} \\ T(s_n) = v_{n,1}, & T^{(1)}(s_n) = v_{n,2} \quad \ldots, & T^{(m_n-1)}(s_n) = v_{n,m_n} \\ d(T) = r & & \end{cases} \tag{8.25}$$

where[1] $r \geq 0$, $\{s_1, s_2, \ldots, \ s_n\} \subset RHP$ and

$$\{v_{11}, \ldots, v_{1m_1}, v_{21}, \ldots, v_{2m_2}, \ldots, v_{n1}, \ldots, \ v_{nm_n}\} \subset C$$

are given. As in previous sections, we consider the problems of producing one interpolant T_1 for INT_h, and of obtaining a parameterization of all interpolants for INT_h.

We use two $\mathcal{RH}^\infty$ functions, T_0 and T_1. The assumptions on these are that T_1 satisfies INT_h, and that T_0 satisfies the corresponding set of homogeneous conditions INT_h^0. Furthermore, suppose that all the RHP zeros of T_0 are listed in INT_0^h, and that $T_0^{(m_\ell)}(s_\ell) \neq 0$, $(\ell = 1, \ldots, n)$.

A variation of the method we have been using works for higher order interpolation problems as well. The difference here is that the formula for T now contains summands with numerators of the form 1, $(s-s_1)$, $(s-s_1)^2$, …, $(s-s_1)^{m_1-1}$, $(s-s_2)(s-s_1)^{m_2-1}$, ….

---

[1] The inclusion $\{s_1, \ldots, s_n\} \subset RHP$ is not required for interpolation. However, it is essential for internal stability.

Theorem 10 *Assume the hypotheses of Theorem 2. Then a function T is in $\mathcal{RH}^\infty$, has relative degree r and satisfies INT^h if and only if there exists $H \in \mathcal{RH}^\infty$ such that*

$$T(s) = T_1(s) + T_0(s)H(s). \tag{8.26}$$

Example. Find $T \in \mathcal{RH}^\infty$ with relative degree $d(T) = 0$ such that

$$T(0) = 1 \ , \ T^{(1)}(0) = 2 \ , \ T(3) = 4 \ , \ T(4) = 0 \tag{8.27}$$

Solution. The expression for T is

$$T(s) = c_0 + c_1 \frac{s}{s+1} + c_2 \frac{s^2}{(s+1)^2} + c_3 \frac{s^2(s-3)}{(s+1)^3}. \tag{8.28}$$

We also need the derivative of T,

$$T^{(1)}(s) = \frac{c_1}{(s+1)^2} + \frac{2c_2 s}{(s+1)^3} + \frac{c_3(-6s+6s^2)}{(s+1)^4}. \tag{8.29}$$

Setting $s = 0$ in (8.28) and in (8.29) gives $c_0 = 1$ and $c_1 = 2$. Now $s = 3$ in (8.28) gives

$$4 = c_0 + c_1 \frac{3}{4} + c_2 \frac{9}{16} = 1 + 2 \frac{3}{4} + c_2 \frac{9}{16}. \tag{8.30}$$

Solving for c_2 in (8.30) yields $c_2 = \frac{8}{3}$. Finally $s = 4$ in (8.28) leads to

$$0 = c_0 + c_1 \frac{4}{5} + c_2 \frac{16}{25} + c_3 \frac{32}{125} \ , \tag{8.31}$$

and this implies that $c_3 = \frac{-1615}{48}$.

8.3.2 Plants with multiple RHP zeros and poles

A RHP pole or zero of the plant with multiplicity higher than 1 in an internally stable system can be translated as an interpolation condition on derivatives of T at that location. More precisely:

Theorem 11 *Consider the closed loop system $\mathcal{S}$. Let ρ_ℓ ($\ell = 1, \ldots, n$) denote the poles and z_ℓ ($\ell = 1, \ldots, m$) denote the zeros of the plant P in the closed* RHP, *so that these poles and zeros have multiplicity n_ℓ, $\ell = 1, \ldots, n$ and m_ℓ, $\ell = 1, \ldots, m$, respectively. If $\mathcal{S}$ is internally stable, then for some* *integer $d_c \geq 0$ the closed loop transfer function $T = PC(1+PC)^{-1}$ satisfies the interpolation conditions*

$$INT^h \begin{cases} T(\rho_\ell) = 1, & T^{(1)}(\rho_\ell) = 0 \quad \ldots, \quad T^{(n_\ell - 1)}(\rho_\ell) = 0 \quad (\ell = 1, \ldots, n) \\ T(z_\ell) = 0, & T^{(1)}(z_\ell) = 0 \quad \ldots, \quad T^{(m_\ell - 1)}(z_\ell) = 0 \quad (\ell = 1, \ldots, m) \\ d(T) = & d(P) + d_c \end{cases} \tag{8.32}$$

Conversely, if T is any function $\mathcal{RH}^\infty$ that satisfies INT^h for some $d_c \geq 0$, then the closed loop system $\mathcal{S}$ associated with T is internally stable and its compensator has relative degree d_c.

We now state the result with the parameterization of internally stable systems.

Theorem 12 *Assume the hypotheses of Theorem 11, and let T_1, T_0 in $\mathcal{RH}^\infty$ be such that T_1 satisfies INT^h, and T_0 satisfies INT_0^h. Furthermore, suppose that all the RHP zeros of T_0 are listed in INT_0^h, and that $T_0^{n_\ell}(\rho_\ell) \neq 0$, $(\ell = 1, \ldots, n)$, $T_0^{m_\ell}(z_\ell) \neq 0$, $(\ell = 1, \ldots, m)$.*

If $T \in \mathcal{RH}^\infty$ is the closed loop transfer function of an internally stable system $\mathcal{S}$ with plant P and $d(C) = d_c$, then there exist $H \in \mathcal{RH}^\infty$ such that

$$T(s) = T_1(s) + T_0(s)H(s) \tag{8.33}$$

Conversely, if (8.33) holds for some $H \in \mathcal{RH}^\infty$, then the system $\mathcal{S}$ associated with P and T is internally stable, and the degree of its compensator is $d(C) = d_c$.

Example. If $P(s) = \frac{(s+3)(s-2)}{s^2}$ and $d_c = 1$, then the interpolation conditions on any $T \in \mathcal{RH}^\infty$ for internal stability of the system $\mathcal{S}$ are given by

$$INT \qquad T(0) = 1,\ T'(0) = 0,\ T(2) = 0,\ d(T) = 1.$$

If $a = -1$ is selected as the location of the poles of T, then the technique described in Subsection 8.3.1 yields the interpolant

$$\frac{-15s^2}{4(1+s)^3} + \frac{s}{(1+s)^2} + \frac{1}{1+s} \tag{8.34}$$

Thus we have the parameterization

$$T(s) = \frac{-15s^2}{4(1+s)^3} + \frac{s}{(1+s)^2} + \frac{1}{1+s} + \frac{s^2(s-2)}{(1+s)^4}\, T_1(s), \qquad \forall T_1 \in \mathcal{RH}^\infty. \tag{8.35}$$

8.4 Exercises

1. 1. Find an interpolant for each set of conditions below:

      a. $T(1) = 0$, pole location $s = -1$.
      b. $T(2+j) = 3$, $T(2-j) - 3$, pole location $s = -10$.
      c. $T(0) = 0$, $T(1) = 0$, $T(2) = 0$, pole location $s = -5$.
      d. $T(1) = 2$, $T(3+j) = 0$, $T(3-j) = 0$, pole location $s = -1$.

   2. Prove that the system of equations (8.4) does have a solution in $c_0, \ldots, c_{n-1}$; and that this solution is unique.

   3. Prove that if the function $T \in \mathcal{RH}^\infty$ satisfies $T(s_1) = 0, \ldots, T(s_n) = 0$, then either T is identically zero or degree $\{\text{denominator}(T)\} \geq n$.

   4. Prove that given *INT* with n data points, *there exists a unique* $T \in \mathcal{RH}^\infty$ that satisfies *INT*, with the following property: if p and q are polynomials such that $T = p/q$, then $degree(\ q\ ) < n$.

2. 1. Find *all* interpolants satisfying conditions stated in Exercises 1.a–d.

   2. Prove Theorem 6.

3. 1. Find an interpolant for each set of conditions below:

      a. $T(1) = 0$, pole location $s = -1$, relative degree $d(T) = 2$.
      b. $T(2+j) = 3$, $T(2-j) = 3$, pole location $s = -10$, $d(T) = 1$.
      c. $T(0) = 2$, $T(j) = 1 + 2j$, $T(-j) = 1 - 2j$, pole location $s = -1$, $d(T) = 1$.

   2. Prove the statements below:

      a. If both $T_1, T_2 \in \mathcal{RH}^\infty$ have relative degree r, then $T_1 + T_2$ has relative degree $\leq r$. Give an example where strict inequality occurs.
      b. If $T_1, T_2 \in \mathcal{RH}^\infty$ have relative degrees r_1, r_2 respectively, then the product T_1T_2 has relative degree $r_1 + r_2$.

   3. Find *all* functions $T \in \mathcal{RH}^\infty$ that satisfy the conditions stated in Exercise 1a–c.

   4. Prove Theorem 2.

4. Consider the following system:

Define the "closed loop compensator" Q to be $Q = C(1+CP)^{-1}$.

a. Check that Q is the $e_1 \rightarrow e_2$ transfer function.

b. What interpolation conditions must Q satisfy for the system to be internally stable?

c. Do all Q meeting these interpolation conditions correspond to an internally stable system?

5. Find a formula for all closed loop transfer functions T that correspond to an internally stable system $\mathcal{S}$, if the relative degree of the compensator is 0 and the plant $P(s)$ is given by

    a. $1/(s-3)$
    b. $(s^2-4)/(s^2-1)$
    c. $(s-5)/(s^2+1)$
    d. $(s+2)/(s+4)$

6. Do 6a – d, but now with $d(C) = 1$.

7. Use the results in Section 8.3 to find the closed loop T that arise from internally stable systems with degree of compensator $d_c = 0$, and plant

    a. $(s^2-2s+1)/(s^2-4)$
    b. $1/(s-2)^2$
    c. $(s+2)/(s-2)^2$
    d. Do 8a – c with $d(C) = 1$.

8. Supply a proof for the results stated in Section 8.3.

8.5 References and further reading

References to amplify our presentation of internal stability, interpolation and control (but with a different point of view) are

[BB] Stephen P. Boyd and Craig P. Barrat, *Linear Controller Design*, Prentice Hall, 1991.

[DC] Carl DeBoor and Richard Conte, *Elementary Numerical Analysis: an algorithmic approach.* McGraw Hill, New York, 1980.

[F] Bruce Francis. *A Course in $\mathcal{RH}^\infty$-Control.* Springer Verlag, 1985.

[V] M. Vidyasagar. *Control Systems Synthesis.* The MIT Press, 1984.

[BGR] J. Ball, I. Gohberg, L. Rodman. *Interpolation of Rational Matrix Functions.* Operator Theory Advances and Applications vol. 45, Birkhauser, New York, 1990.