

Introduction to Feedback Control

Introduction to Feedback Control

Clarence W. Rowley

Fall 2018

Copyright ©2018 by Clarence W. Rowley

All Rights Reserved

Published by Clarence W. Rowley
Princeton, New Jersey 08540

Third printing: September 2018

Contents

Contents	v
Preface	vii
Notation	ix
1 Introduction	1
1.1 Principles of feedback	2
1.2 Dynamics	9
1.3 Linearization	13
Exercises	21
2 Transfer functions	25
2.1 Frequency response	25
2.2 Impulse response	31
2.3 Laplace transforms	33
2.4 Zero-frequency gain	38
2.5 Block diagrams	40
2.6 Gang of four	46
Exercises	50
3 Linear dynamics	53
3.1 Poles and zeros	53
3.2 Second-order systems	57
3.3 Time-domain specifications	59
3.4 Effect of zeros	62
Exercises	64
4 Classical feedback design	67
4.1 Feedback changes dynamics	67
4.2 Root-locus diagrams	74
4.3 Bode plots	91
4.4 Loop transfer function	105
4.5 Nyquist stability criterion	110

4.6	Stability margins	118
4.7	Loopshaping	121
4.8	Frequency-domain design	129
	Exercises	136
5	State-space systems	143
5.1	Solution of state-space equations	144
5.2	Eigenvectors for linear ODEs	152
5.3	Realizations	155
	Exercises	159
6	State feedback	163
6.1	Controllability	164
6.2	Pole placement	173
6.3	Tracking with state feedback	178
6.4	Loop gain	181
6.5	Linear Quadratic Regulator (LQR)	188
	Exercises	201
7	Output feedback	207
7.1	Observability	207
7.2	Observers	214
7.3	Observer-based feedback	219
7.4	Fundamental limitations	227
	Exercises	235
A	Linear algebra	237
A.1	Eigenvectors and eigenvalues	237
A.2	Positive-definite matrices	240
A.3	Coordinate changes	241
	Exercises	243
B	Complex variables	245
B.1	Basic properties	245
B.2	Functions of a complex variable	247
B.3	The logarithm and the argument principle	252
B.4	Bode's theorems	255
B.5	Maximum modulus principle	262
	Exercises	265
C	MATLAB commands	267
	Bibliography	271
	Index	273

Preface

These notes are for an introductory course on control systems, taught at Princeton University. There are several outside references that may be helpful, to complement these notes. I recommend the book by Åström and Murray [1], which provides an excellent treatment of both classical and modern control methods, from two of the world's great experts on the subject. For classical control concepts, I recommend the book Franklin et al. [11]. Other good references for modern control topics are Friedland [12] or Bédanger [2]. For more advanced topics, see Skogestad and Postlethwaite [22]. For an elementary (but rigorous) overview of robust control, see Doyle et al. [7].

Acknowledgments

Over the years, many graduate students have served as Associate Instructors for this course, and have provided valuable feedback about these lectures, and the corresponding labs. I especially thank Mark Luchtenburg, Jonathan Tu, Imene Goumiri, and Simon LeBlanc. I am particularly indebted to Prof. Michael Littman, who has worked alongside me since my first year as a faculty member, helping me to become a better lecturer, and continually refining the laboratory portions of this course.

Clancy Rowley
Princeton University
February, 2014

Notation

Throughout these notes, vectors are denoted in boldface: for instance, a vector with n components is written $\mathbf{x} = (x_1, \dots, x_n)$. When used in matrix computations, \mathbf{x} is treated as a *column* vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

The corresponding row vector is denoted

$$\mathbf{x}^\top = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}.$$

Matrices are denoted by boldface capital letters, such as \mathbf{A} .

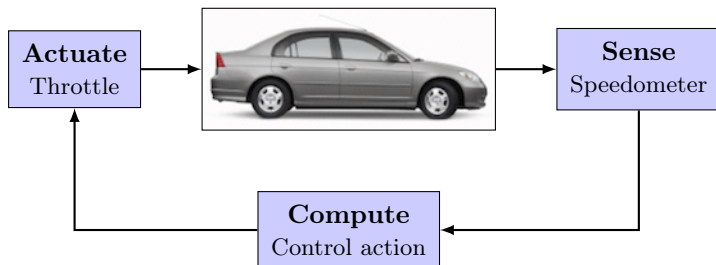
Chapter 1

Introduction

This course is about the mathematical techniques used to analyze feedback systems. What is a feedback system? This is best illustrated by an example. Consider an automobile's cruise control, which is used to maintain a desired speed even in the presence of hills, headwinds, or tailwinds. This system involves several different components:

- a speed sensor (the speedometer)
- some computation (deciding how much to open the throttle)
- an actuator (the throttle).

It is common to draw a picture as follows, called a *block diagram*:



From this picture, we can see that there is a *feedback loop*: the speed measured by the speedometer influences the throttle, which changes the speed of the car, which in turn alters the throttle, and so on. Loosely defined, a feedback system is any system that contains a feedback loop like this.

What is special about feedback systems, and why do we need a whole course in methods for analyzing them? It turns out that feedback can do some very special things, both good and bad, as we will see in the next section. Furthermore, feedback systems can behave in quite non-intuitive ways, and in trying to use intuition to analyze them, one can easily get

stuck in circular arguments that are incorrect. Instead, we must use more precise tools, namely mathematics. Fortunately, there is a wonderful theory for analyzing these systems, and this is the theory we will learn about in this course.

1.1 Principles of feedback

It turns out that feedback can do some remarkable things. For instance, using feedback, we can build *reliable components from unreliable parts*. This may sound impossible, but this principle is what enabled the development of highly accurate electrical amplifiers in the 1930s, for use in trans-continental telephone communications. You can observe this same principle today: if you order 5 identical audio amplifiers from Amazon, place them next to each other, and set their volume knobs to the same level, they will behave identically. However, inside these amplifiers are transistors whose gains vary dramatically from one box to the next. Yet somehow, when these transistors are connected together, the overall circuits behave the same. This is because these transistors are connected in feedback circuits so that the overall gain is relatively insensitive to the gain of the individual transistors. This is one of the first magical properties of feedback.

A related property of feedback is that it can make systems *insensitive to external disturbances*. For instance, in the automobile cruise control example mentioned above, the car can maintain a desired speed even in the presence of hills and winds (the external disturbances), regardless of the number of passengers in the car or how full the fuel tank is. The reasons for this are obvious: the speedometer can measure the speed of the car, and therefore one knows how to compensate, whether to step on the gas or release it. But this ability to make a system insensitive to disturbances illustrates a more general principle of feedback.

Another remarkable property of feedback is that, using feedback, one can *modify the dynamics* of a system, for instance stabilizing an unstable system. An example of this is the F-16 Fighting Falcon, which is the first production aircraft intentionally designed with unstable longitudinal dynamics. What does this mean? Imagine you trim an aircraft (e.g., set the thrust and elevators) so that it is flying straight and level. Now, tap the stick forward, so that the nose dips down a little. An aircraft with stable longitudinal dynamics will return to the original trim condition; an unstable aircraft will continue to dive down. As one might imagine, an unstable aircraft could be exceedingly difficult for a pilot to fly. However, an unstable aircraft could also be significantly more maneuverable, if one could control it. One of the major breakthroughs of the F-16 was a fly-by-wire control system that provided *stability augmentation*: the flight control system is actually providing the inputs to the control surfaces (such as elevators), and the pilot merely gives inputs to the flight control system. The resulting “closed-loop” system

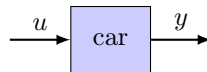
is actually stable, so the pilot feels as though she is flying a stable aircraft, while retaining the improved maneuverability of an unstable aircraft.

The examples above illustrate what we call the *two main principles of feedback*:

1. Feedback can reduce sensitivity, both to external disturbances, and to changes to internal components of a system.
2. Feedback can modify the dynamics of a system.

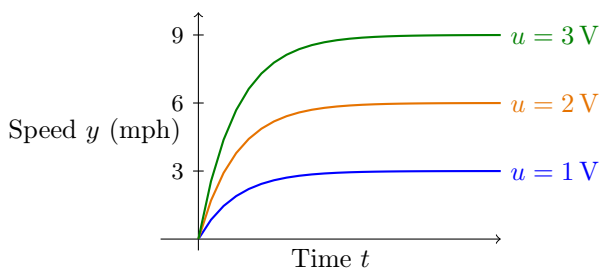
Example: cruise control

We will now illustrate these concepts in more detail, considering again the cruise control example. The goal is to design a speed controller for a car, so that it drives at a set speed, regardless of the incline, or any headwind. We will consider a simple electric car, and neglect the transmission, brakes, etc., so that our only input to the car is a voltage to the motor. We will measure the speed of the car, so our conceptual model of the system is as follows:

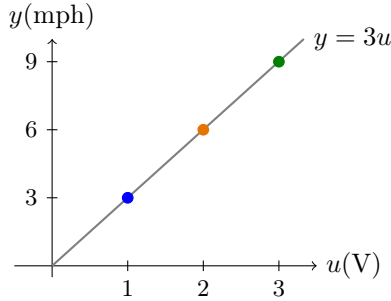


where u is the input voltage to the motor, and y is the speed, in miles per hour (mph). (It is conventional in control theory to denote the input and output of the system to be controlled by u and y , respectively.)

We begin by doing some simple experiments with the car. In particular, we apply a fixed voltage to the motor, and see how fast the car goes. Below are the results of three such experiments:



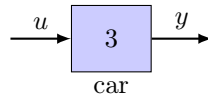
At first, we may only care about the final speed of the car, not how long it takes to ramp up to that speed. If we plot the final speed versus the applied voltage, we get a nice linear plot:



so a reasonable mathematical model for the car is

$$y = 3u. \quad (1.1)$$

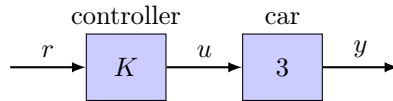
We could represent this equation in a block diagram as follows:



Now, how should we design a control system to achieve a desired speed?

Method 1: Open-loop control

One approach is as follows. Denoting the desired reference speed by r , we could choose $u = Kr$ for some constant K , as shown in the following diagram:



This block diagram is a graphical representation of the equations

$$u = Kr, \quad y = 3u. \quad (1.2)$$

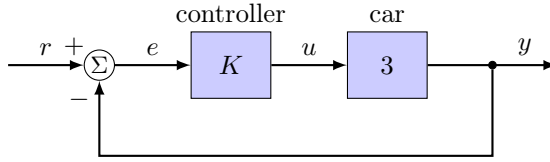
One sees immediately from the diagram that there is no feedback loop, so this approach is known as *open-loop control*. How should we choose K ? We want to design the controller so that $y = r$, and equations (1.2) give

$$y = 3u = 3Kr, \quad (1.3)$$

so we want $3K = 1$, or $K = 1/3$. With this choice of K , we get $y = r$, so it seems we perfectly track our reference speed. However, the trouble with this approach is that we will not be able to compensate for disturbances such as hills or headwinds.

Method 2: Closed-loop control

Another approach is a feedback controller, as shown in the diagram below:



This diagram represents the equations

$$e = r - y, \quad u = Ke, \quad y = 3u, \quad (1.4)$$

where $e = r - y$ is the *tracking error*, or the difference between the desired speed and the actual speed. Now how should we choose K in order for y to track the desired speed r ? We have

$$y = 3u = 3Ke = 3K(r - y). \quad (1.5)$$

This equation illustrates the “circular reasoning” that arises in feedback systems, for unlike (1.3), our expression for y now involves y on the right-hand side! Fortunately, we can get around this circular reasoning using some fancy mathematics, namely elementary algebra: solving for y , we obtain

$$y = \frac{3K}{1 + 3K}r, \quad e = r - y = \frac{1}{1 + 3K}r. \quad (1.6)$$

There is actually no finite value of K for which y identically equals r (or, equivalently, $e \equiv 0$). However, picking a large value of K gives $y \approx r$ and $e \approx 0$: for instance, for $K = 100$, we have

$$y = \frac{300}{301}r, \quad e = \frac{1}{301}r.$$

At this point, it seems as though the first method was better, since we get perfect tracking. However, the subsequent sections will reveal some major advantages of feedback.

Robustness to changes in the system

Now consider what happens if our motor changes over time. For instance, as parts wear or get dirty, the motor may slow down a bit, so that the real motor behaves as $y = 2u$ instead of $y = 3u$. This happens without our knowledge, however, so we still set $K = 1/3$ in Method 1 above, and the tracking error for Method 1 (open-loop control) now becomes

$$e = r - y = r - 2u = r - 2Kr = (1 - 2K)r = \frac{1}{3}r. \quad (1.7)$$

We now have a 33% tracking error, which seems pretty large: if our desired speed is 60 mph, our actual speed will be only 40 mph.

For Method 2 (closed-loop control), however, the tracking error with $K = 100$ is obtained by suitably modifying (1.6), to obtain

$$e = \frac{1}{1 + 2K} r = \frac{1}{201} r, \quad (1.8)$$

so less than 0.5% tracking error—much better! Now if our desired speed is 60 mph, our actual speed will be 59.7 mph. So if our car might be changing over time, or if we are not too confident in our mathematical model (1.1) (which, admittedly, was pretty crude), then the closed-loop controller of Method 2 should be far superior to the open-loop controller of Method 1.

This is precisely what is meant by the statement that

Feedback reduces sensitivity to changes in internal components.

We can precisely quantify just how much feedback reduces this sensitivity, as follows. For Method 1, if the car changes from $y = 3u$ to $y = 2u$, the overall controlled system changes by

$$\frac{\Delta y}{y} = \frac{2Kr - 3Kr}{3Kr} = -\frac{1}{3},$$

or by 33%, independent of the choice of K . For Method 2 (the feedback system), however, we see from (1.6) that the overall controlled system changes by

$$\frac{\Delta y}{y} = \frac{\frac{2K}{1+2K} - \frac{3K}{1+3K}}{\frac{3K}{1+3K}} = -\frac{1}{3} \cdot \frac{1}{1 + 2K}.$$

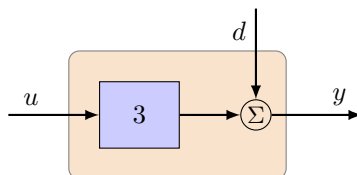
We say that feedback reduces the sensitivity by a factor of $1/(1 + 2K)$ (we will learn in Section 2.6 that this is the *sensitivity function* for this system), which for $K = 100$ gives a substantial factor of $1/201$.

Robustness to external disturbances

Next, suppose we include the effects of hills and winds, modeling both as a single disturbance d . Our model for the car's speed y is now

$$y = 3u + d, \quad (1.9)$$

which may be represented in the block diagram as



Thus, a positive value of d increases the car's speed, so may be viewed either as a downhill incline, or as a tailwind; conversely, a negative value of d represents an uphill slope or a headwind.

For Method 1 (open-loop control), we have

$$\begin{aligned} u &= Kr \\ y &= 3u + d = 3Kr + d, \end{aligned}$$

and the tracking error is

$$e = r - y = (1 - 3K)r - d.$$

Choosing $K = 1/3$ as before, we have

$$e = -d. \tag{1.10}$$

For Method 2 (feedback control), we have

$$\begin{aligned} u &= K(r - y) \\ y &= 3u + d = 3K(r - y) + d. \end{aligned}$$

Again, our expression for y involves y itself, but we can once again solve for y using the magic of algebra. We are ultimately interested in the tracking error $e = r - y$, so we will actually solve for e directly instead of y . Writing $y = r - e$ and solving for e in terms of the two inputs r and d , we have

$$e = \frac{1}{1 + 3K}r - \frac{1}{1 + 3K}d. \tag{1.11}$$

If $d = 0$, we have the same expression as (1.6), as we expect, but now we see an additional term resulting from the disturbance d . For $K = 100$, this becomes

$$e = \frac{1}{301}r - \frac{1}{301}d. \tag{1.12}$$

Contrasting this with expression (1.10) for Method 1, we see that feedback has reduced the effect of disturbances by a factor of $1/301$. This again is a dramatic effect, and this is precisely what is meant by the statement that

Feedback reduces sensitivity to external disturbances.

Note the distinction between this situation and that discussed in Section 1.1: previously, we looked at changes in *the car itself* (e.g., changing over time), while in this section, we are investigating the effects of *external* disturbances such as hills or winds. Feedback can reduce the sensitivity to both types of unknown effects.

Feedback changes dynamics

In our analysis above, we completely ignored the dynamics of the system, but we know that in reality, the car takes some time to ramp up to its final speed. Let us try to account for the dynamics of the system using a simple time delay. Suppose we sample the system at some time interval Δt , and let $y_j = y(j\Delta t)$ and $u_j = u(j\Delta t)$, for $j = 0, 1, 2, \dots$. If we change the voltage u_j , we know the speed does not respond instantaneously, so a simple model would be to assume that the speed reaches equilibrium after one time step:

$$y_{j+1} = 3u_j. \quad (1.13)$$

What does the controlled system look like in this case? For the open-loop controller (Method 1), the control law is still $u_j = Kr_j$ (with $K = 1/3$), and we have

$$y_{j+1} = 3Kr_j = r_j,$$

so the situation is essentially the same as (1.3), except the speed is delayed by one step. (Of course, this open-loop control approach will still be sensitive to changes in the system and to external disturbances, as considered previously.)

However, for the closed-loop controller (Method 2), the situation is more interesting. Now, the control law is $u_j = K(r_j - y_j)$, so (1.13) becomes

$$y_{j+1} = 3K(r_j - y_j) = -3Ky_j + 3Kr_j. \quad (1.14)$$

Now, even if r_j is a constant, y_j still changes in time according to (1.14): it has dynamics. It is enough to consider the case $r_j = 0$, for which the system becomes

$$y_{j+1} = -3Ky_j. \quad (1.15)$$

It is easy to see that, given an initial condition y_0 , the solution of (1.15) is

$$y_j = (-3K)^j y_0. \quad (1.16)$$

In particular, if $|3K| < 1$, then $|y_j|$ goes to zero as $j \rightarrow \infty$, so the speed converges (to zero) as time goes on. However, if $|3K| > 1$, then $|y_j|$ keeps increasing as j increases, and the system is *unstable*. Intuitively, if the gain K is too large, then the controller overcorrects, and then at the next step overcorrects further in the other direction, and so on, leading to an instability.

This illustrates another fundamental principle of feedback:

Feedback changes the dynamics of a system.

In particular, it can stabilize an unstable system, but as we see here, it can also destabilize a perfectly well-behaved, stable system.

What is this limiting gain for which instability occurs? The limiting gain K occurs when $3K = 1$, or for $K = 1/3$. Thus, according to the

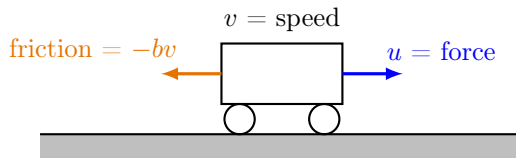
model (1.13), if $K > 1/3$, then the closed-loop system is unstable. But this is a totally unacceptable situation, since in order to achieve the benefits of feedback seen in the previous sections, we required K to be *large* (we took $K = 100$), and $K = 1/3$ is not large. For instance, for $K = 1/3$, the tracking error from (1.6) is a whopping 50%; furthermore, the effect of disturbances, calculated in (1.11), is reduced by only 50%, relative to the open-loop system.

Note also that this limiting value $K = 1/3$ is independent of the timestep Δt , so it seems that, even for an arbitrarily small delay Δt , we cannot use a feedback gain larger than $K = 1/3$. Fortunately, the situation in reality is not quite this bad, as the model (1.13) is a bit too simple: the car's inertia and natural ramp-up time actually help matters, and as long as the delay is not too large (relative to the natural timescales of the system), the real physical system will not have problems with instability, even for large K . Nevertheless, it is certainly true that feedback modifies dynamics, and can introduce instability. We will learn more sophisticated ways of representing dynamics in the next section, and will see much more in Chapter 2.

1.2 Dynamics

Most of the systems we will want to control are accurately described by ordinary differential equations (ODEs). You are probably familiar with deriving such models (for instance from MAE 206), but here we will give a quick review, by way of some examples. In each example, we will specify an *input* u , which is the quantity we have control over, and an *output* y , which is the quantity we measure with a sensor.

Example 1.1 (Cruise control). Consider a simple model of a car with mass m moving along a horizontal surface with speed $v(t)$. Suppose we can apply a force $u(t)$ (our actuator: for instance, a motor), and friction gives a resisting force $-bv$, as shown in the diagram below:

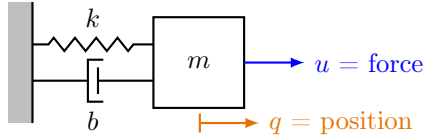


Newton's equations $F = ma$ then give us the equations of motion

$$m\dot{v} = -bv + u. \quad (1.17)$$

Thus, the dynamics of our car are described by a first-order linear ODE. If we measure the velocity (e.g., with a speedometer or tachometer), the output is then $y = v$. \diamond

Example 1.2 (Spring-mass-damper). Consider the mechanical system illustrated in the diagram below:

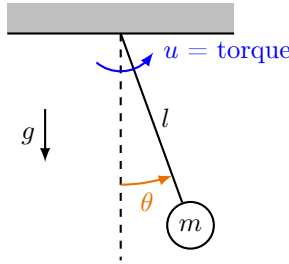


in which our input u is a force on the mass m . Newton's equations $F = ma$ then become

$$m\ddot{q} = -b\dot{q} - kq + u. \quad (1.18)$$

If our sensor measures the position of the mass, then the output is $y = q$. Note that, if desired, we may reduce the number of parameters by nondimensionalizing, as done in Section 3.2, but for now we will retain the dimensional parameters. \diamond

Example 1.3 (Simple pendulum). Consider the dynamics of a simple pendulum, a bob of mass m attached to a massless rod of length l , swinging under gravity, with an applied torque $u(t)$ (which we can control):

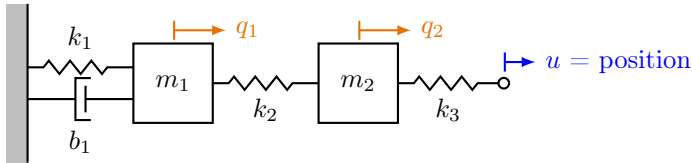


Noting that the moment arm is $l \sin \theta$, and the moment of inertia about the pivot point is ml^2 , Newton's equations give

$$ml^2\ddot{\theta} = -mgl \sin \theta + u. \quad (1.19)$$

If we measure the angle of the pendulum, the output is $y = \theta$. \diamond

Example 1.4 (Double spring-mass). Consider the dynamics of the following system, consisting of two masses connected to a wall by springs. Here, our input $u(t)$ is the *position* of the free end of the rightmost spring in the diagram:

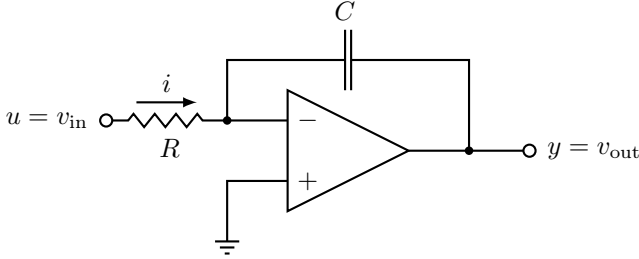


Applying Newton's equations to each of the two masses gives the coupled equations

$$\begin{aligned} m_1\ddot{q}_1 &= -k_1q_1 + k_2(q_2 - q_1) - b_1\dot{q}_1 \\ m_2\ddot{q}_2 &= -k_2(q_2 - q_1) + k_3(u - q_2). \end{aligned} \quad (1.20)$$

Here, if we measure the positions of both masses, the output is a vector $\mathbf{y} = (q_1, q_2)$. \diamond

Example 1.5 (Op-amp circuit). Consider the circuit in the diagram below:



Recall that for an ideal operational amplifier, one assumes infinite input impedance, so there is zero current into the inputs. Furthermore, one assumes an infinite gain, so the voltage across the two inputs is equal. Therefore, the current through the capacitor equals the current through the resistor, or

$$\frac{u}{R} = -C \frac{dy}{dt}. \quad (1.21)$$

Note that, if we integrate this equation, we obtain

$$y = -\frac{1}{RC} \int u \, dt,$$

so this circuit is an integrator. \diamond

State-space form

All of the above models are ordinary differential equations, but they all have slightly different forms: some are first-order, some are second-order, and some have more than one dependent variable. We would like to write these all in a standard form. It turns out we can write all of these as systems of first-order differential equations, in what is called *state-space form*:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \quad (1.22a)$$

$$y = h(\mathbf{x}, u). \quad (1.22b)$$

Here, u and y represent the input and output, as before, and $\mathbf{x} = (x_1, \dots, x_n)$ is a vector with n real components (written $\mathbf{x} \in \mathbb{R}^n$), called the *state*:

The *state* (usually denoted \mathbf{x}) is a vector of independent physical quantities that completely determine the future evolution of a system, assuming no disturbances or other inputs.

The *input* u describes any external forcing to the system (e.g., actuators and/or disturbances), and the *output* y denotes the measured quantities. The dynamics are determined by $\mathbf{f}(\mathbf{x}, u) = (f_1(\mathbf{x}, u), \dots, f_n(\mathbf{x}, u))$, a function with n components, and the measurement function h is a scalar-valued function (assuming we have only one sensor measurement).

In this course, we will be particularly interested in *linear* systems, for which \mathbf{f} has the special form

$$\mathbf{f}(\mathbf{x}, u) = \mathbf{A}\mathbf{x} + \mathbf{B}u, \quad (1.23)$$

where \mathbf{A} and \mathbf{B} are matrices of appropriate dimension (in particular, if $\mathbf{x} \in \mathbb{R}^n$ and $u \in \mathbb{R}$, then \mathbf{A} is $n \times n$ and \mathbf{B} is $n \times 1$). Similarly, for a linear system, h has the form $h(\mathbf{x}, u) = \mathbf{C}\mathbf{x} + Du$, so the state-space equations (1.22) have the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (1.24a)$$

$$y = \mathbf{C}\mathbf{x} + Du. \quad (1.24b)$$

Let us return to our examples above, and write each in the standard state-space form.

Example 1.6 (Cruise control). From equation (1.17), we see that we may choose the state to be v , and the equations may be written

$$\begin{aligned} \dot{v} &= -\frac{b}{m}v + \frac{1}{m}u \\ y &= 1 \cdot v + 0 \cdot u. \end{aligned}$$

These are in the standard state-space form for a linear system (1.24), with $A = -b/m$, $B = 1/m$, $C = 1$, and $D = 0$. \diamond

Example 1.7 (Spring-mass-damper). The dynamics are given by (1.18), a *second-order* ODE. We might think we are in trouble, since the form (1.22a) is a *first-order* ODE. However, we can use a standard trick to transform (1.18) into a system of two first-order ODEs: if we let $v = \dot{q}$, and define $\mathbf{x} = (q, v)$, then the equations may be written

$$\begin{aligned} \dot{q} &= v \\ \dot{v} &= -\frac{k}{m}q - \frac{b}{m}v + \frac{1}{m}u \end{aligned}$$

or, in matrix form,

$$\underbrace{\frac{d}{dt} \begin{bmatrix} q \\ v \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} -\frac{k}{m}q - \frac{b}{m}v + \frac{1}{m}u \end{bmatrix}}_{\mathbf{f}(\mathbf{x}, u)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} q \\ v \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{\mathbf{B}} u,$$

which is of the form (1.24a). Similarly, the equation for the output $y = q$ can be written in the form (1.24b), as

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_C \begin{bmatrix} q \\ v \end{bmatrix} + \underbrace{0}_D u. \quad \diamond$$

Example 1.8 (Simple pendulum). Again, the dynamics are given by a second-order ODE (1.19), so we use the same trick as in the previous example to transform this to a system of two first-order ODEs. Defining $\omega = \dot{\theta}$ and choosing $\mathbf{x} = (\theta, \omega)$, we have

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= -\frac{g}{l} \sin \theta + \frac{1}{ml^2} u, \end{aligned}$$

or, in matrix form,

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} \omega \\ -\frac{g}{l} \sin \theta + \frac{1}{ml^2} u \end{bmatrix} = \mathbf{f}(\mathbf{x}, u), \quad (1.25)$$

which is in (nonlinear) state-space form (1.22a). Note that, because of the term containing $\sin \theta$, the function \mathbf{f} is truly nonlinear, so it may not be written in the linear form (1.23). \diamond

1.3 Linearization

Most of the techniques we will learn in this course apply only to *linear* systems. Linear systems obey the principle of *superposition* (i.e., if x_1 and x_2 are both solutions, then $x_1 + x_2$ is also a solution), and we can use this property to build up more sophisticated tools, such as Laplace transform methods, which make linear systems particularly easy to study.

However, almost all systems that occur in nature are *nonlinear*. So does this mean that the techniques we will learn are not useful for real systems? Fortunately not: *linearization* provides a way to obtain linear systems that closely approximate nonlinear systems. The range of validity of the approximations depends on the particular problem: some problems are severely nonlinear, and the linearization says little, or even nothing, about the full nonlinear dynamics. But very often, the range of validity is large enough to adequately describe the behavior of the system in the operating range we care about. This is particularly true for control applications, where typically we want to use control to regulate a system about a particular operating point.

Some cases are simple: for instance, as we just saw in Example 1.8, the dynamics of a simple pendulum are given by the ODE

$$ml^2 \ddot{\theta} = -mgl \sin \theta + u.$$

If the pendulum is close to vertical ($|\theta| \ll 1$), then we can use the small-angle approximation $\sin \theta \approx \theta$, and the linearized equations are

$$ml^2\ddot{\theta} = -mgl\theta + u.$$

In state-space form, with $\omega = \dot{\theta}$, these become

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -g/l & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1/(ml^2) \end{bmatrix} u. \quad (1.26)$$

However, other cases are not so obvious. For instance, a model that arises in queuing systems (see [1, Example 2.10]) is given by the nonlinear ODE

$$\dot{x} = \lambda - \mu \frac{x}{x+1}.$$

Here, it is not so obvious how to obtain an approximate linear system. There are many nonlinear systems that are much more complicated than this (for instance, the rotary pendulum apparatus we will use in the lab later in the course) that we would also like to obtain linear approximations to. Here, we illustrate a general procedure for finding such linearizations.

Linearization of a differential equation

We first consider systems of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (1.27)$$

(that is, in state-space form (1.22a) but with no control input u). We first need to define the notion of an equilibrium point:

A point \mathbf{x}_0 is called an *equilibrium point* (or *fixed point*) of the system (1.27) if $\mathbf{x}(t) = \mathbf{x}_0$ is a solution of the system. Thus, \mathbf{x}_0 is an equilibrium point if $\mathbf{f}(\mathbf{x}_0) = \mathbf{0}$.

Linearization is concerned with *approximating* the dynamics of (1.27) *near an equilibrium point*¹. The system may have many different equilibrium points, so first we must choose the one we wish to linearize about. Choose an equilibrium point \mathbf{x}_0 , and write

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{z}. \quad (1.28)$$

The new quantity \mathbf{z} is simply how far \mathbf{x} is from the equilibrium point, and \mathbf{z} will be small if \mathbf{x} is close to \mathbf{x}_0 . Since \mathbf{x}_0 is chosen beforehand, if we find how \mathbf{z} behaves, we will also know how \mathbf{x} behaves.

¹There are other notions of linearization, such as linearizing about a particular trajectory, but these lead to *time-varying* linear systems, which are more difficult to study. Here we address only linearization about equilibrium points.

Let us now obtain a linear approximation of (1.27) assuming \mathbf{z} is small (that is, assuming \mathbf{x} is close to \mathbf{x}_0). If \mathbf{f} is differentiable at \mathbf{x}_0 , then the Taylor series of \mathbf{f} is

$$\mathbf{f}(\mathbf{x}_0 + \mathbf{z}) = \mathbf{f}(\mathbf{x}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0} \cdot \mathbf{z} + \text{Higher order terms} \quad (1.29)$$

where

$$\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0} \equiv \left[\begin{array}{cccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{array} \right]_{\mathbf{x}=\mathbf{x}_0} \quad (1.30)$$

is the *Jacobian matrix* of \mathbf{f} , evaluated at the point \mathbf{x}_0 (also called the *derivative* at \mathbf{x}_0 , and often denoted $\mathbf{Df}(\mathbf{x}_0)$). Note that we evaluate this matrix of partial derivatives at the equilibrium point \mathbf{x}_0 , so this is just a constant $n \times n$ matrix.

In linearization, we neglect the higher-order terms, so the equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ becomes

$$\dot{\mathbf{x}}_0 + \dot{\mathbf{z}} = \mathbf{f}(\mathbf{x}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0} \cdot \mathbf{z}$$

Now, since \mathbf{x}_0 is an equilibrium point, $\dot{\mathbf{x}}_0 = \mathbf{f}(\mathbf{x}_0) = 0$, so defining the constant matrix

$$\mathbf{A} := \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_0},$$

the equation becomes

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z}. \quad (1.31)$$

Equation (1.31) is called the *linearization* of the system (1.27) about the point \mathbf{x}_0 . Note that the linearization gives an ordinary differential equation for the position *relative to the equilibrium point*. That is, it gives an equation for $\mathbf{z} = \mathbf{x} - \mathbf{x}_0$. Often, the equilibrium point is the origin ($\mathbf{x}_0 = 0$), in which case (1.31) becomes simply

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}. \quad (1.32)$$

(In fact, we may always choose a coordinate system in which $\mathbf{x}_0 = 0$, by translating the coordinate system so that the equilibrium point is at the origin.)

Example 1.9. Consider the one-dimensional system

$$\dot{x} = -2x + x^2 + x^3. \quad (1.33)$$

First, we find the equilibrium points of this system. Letting

$$f(x) = -2x + x^2 + x^3 = x(x-1)(x+2), \quad (1.34)$$

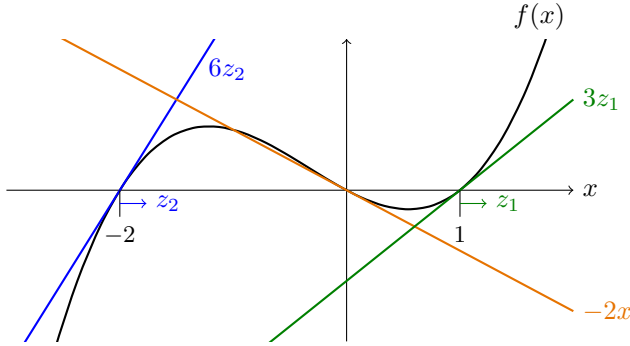


Figure 1.1: Graph of the function f from equation (1.34), along with linearizations from equations (1.35), (1.36), and (1.37).

we find that $f(x) = 0$ when $x = 0, 1, -2$. Thus, there are three equilibrium points. Linearizing the system about the first equilibrium point $x_0 = 0$, we let $x = x_0 + z = 0 + z$, and the linearized equations become

$$\dot{z} = f'(0) \cdot z.$$

Here, $f'(x) = -2 + 2x + 3x^2$, so $f'(0) = -2$, and since $x = z$, the linearized system is simply

$$\dot{x} = -2x. \quad (1.35)$$

This approximation is valid when x is close to zero (since $x_0 = 0$). The range of validity is clearly seen from Figure 1.1, which shows a graph of the function f , along with the linear approximation $f(x) \approx -2x$.

Next, we linearize about the equilibrium point $x_0 = 1$. Writing $x = 1 + z_1$ (so z_1 is the displacement from the equilibrium point at $x = 1$), the linearized system becomes

$$\dot{z}_1 = f'(1) \cdot z_1 = 3z_1, \quad \text{where } x = 1 + z_1. \quad (1.36)$$

This approximation is valid when x is close to 1 (z_1 close to zero), as is also apparent from Figure 1.1. Finally, linearizing about $x_0 = -2$ and writing $x = -2 + z_2$ gives

$$\dot{z}_2 = f'(-2) \cdot z_2 = 6z_2, \quad \text{where } x = -2 + z_2, \quad (1.37)$$

valid when x is near -2 (z_2 close to zero). We have obtained three different approximations of the full nonlinear system (1.33), each of which is a good approximation only near its corresponding equilibrium point. \diamond

Linearization of a system with inputs and outputs

We now consider a nonlinear system with an input u and output y :

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u) \quad (1.38a)$$

$$y = g(\mathbf{x}, u), \quad (1.38b)$$

and suppose $\mathbf{x} = \mathbf{x}_0$ is an equilibrium point of (1.38a) with $u = u_0$ (that is, $\mathbf{f}(\mathbf{x}_0, u_0) = \mathbf{0}$). We proceed as before: writing

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_0 + \mathbf{z} \\ u &= u_0 + v, \end{aligned}$$

we obtain an approximate linear system, assuming \mathbf{z} and v are small. Taking a Taylor series of \mathbf{f} about $\mathbf{x} = \mathbf{x}_0$, $u = u_0$, we have

$$\mathbf{f}(\mathbf{x}_0 + \mathbf{z}, u_0 + v) = \mathbf{A}\mathbf{z} + \mathbf{B}v + \text{Higher order terms}, \quad (1.39)$$

where the matrices \mathbf{A} and \mathbf{B} are given by

$$\mathbf{A} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, u_0)} \equiv \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \bigg|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ u=u_0}} \quad \mathbf{B} = \left. \frac{\partial \mathbf{f}}{\partial u} \right|_{(\mathbf{x}_0, u_0)} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \vdots \\ \frac{\partial f_n}{\partial u} \end{bmatrix} \bigg|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ u=u_0}} \quad (1.40)$$

and are both constant matrices, as before. It is common to choose the coordinates so that $\mathbf{x}_0 = \mathbf{0}$ and $u_0 = 0$, in which case $\mathbf{z} = \mathbf{x}$ and $v = u$. In this case, the linearized equations become

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (1.41)$$

and are valid when \mathbf{x} and u are close to zero. Similarly, the function g from equation (1.38b) may be linearized to give

$$g(\mathbf{x}, u) = \mathbf{C}\mathbf{x} + Du + \text{Higher order terms}, \quad (1.42)$$

where \mathbf{C} and D are given by

$$\mathbf{C} = \left. \frac{\partial g}{\partial \mathbf{x}} \right|_{(\mathbf{x}_0, u_0)} = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \cdots & \frac{\partial g}{\partial x_n} \end{bmatrix} \bigg|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ u=u_0}} \quad D = \left. \frac{\partial g}{\partial u} \right|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ u=u_0}}. \quad (1.43)$$

Note that \mathbf{C} is a $1 \times n$ matrix (or a row vector), and D is just a number.

The overall linearized equations of motion then have the form

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} + Du \end{aligned}$$

which is the standard linear state-space form for a linear system, as in (1.24).

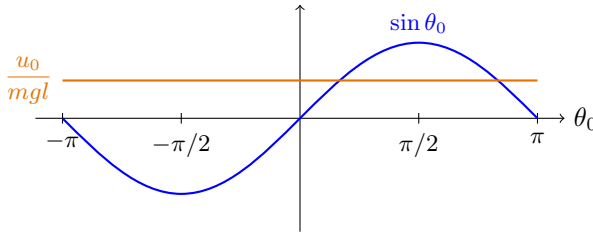
Example 1.10 (Simple pendulum). Consider the dynamics of a simple pendulum, as in Example 1.8. The dynamics are given by equation (1.25):

$$\frac{d}{dt} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} \omega \\ -\frac{g}{l} \sin \theta + \frac{1}{ml^2} u \end{bmatrix} = \mathbf{f}(\mathbf{x}, u),$$

where θ is the angle of the pendulum from the vertical, u is a torque on the pendulum, $\omega = \dot{\theta}$, and $\mathbf{x} = (\theta, \omega)$. First, we find the equilibrium points: letting $\mathbf{f}(\mathbf{x}_0, u_0) = 0$, we obtain $\omega_0 = 0$ and

$$-\frac{g}{l} \sin \theta_0 + \frac{1}{ml^2} u_0 = 0 \implies \sin \theta_0 = \frac{1}{mgl} u_0. \quad (1.44)$$

With no torque ($u_0 = 0$), this is simply $\sin \theta_0 = 0$, which gives the familiar equilibrium points $\theta_0 = 0$ (pendulum down) or $\theta_0 = \pi$ (pendulum up). With a nonzero torque, there may be two, one, or zero solutions to (1.44), as illustrated in the following plot:



If $|u_0| < mgl$, then there are two equilibria; if $|u_0| = mgl$, then there is a single equilibrium at $\theta_0 = \pm\pi/2$, and if $|u_0| > mgl$, there are no equilibria.

Let us consider the case where $u_0 = 0$, so that the equilibria are $\mathbf{x}_0 = (0, 0)$ (pendulum down) and $(\pi, 0)$ (pendulum up). We calculate the derivatives of \mathbf{f} as follows:

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} &= \begin{bmatrix} \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \omega} \\ \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} \cos \theta & 0 \end{bmatrix} \\ \frac{\partial \mathbf{f}}{\partial u} &= \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}. \end{aligned}$$

Evaluating at the equilibrium point $\mathbf{x}_0 = (0, 0)$, we find the equations linearized about the pendulum-down position to be

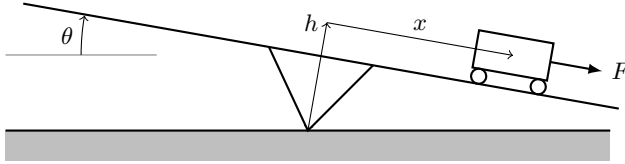
$$\frac{d}{dt} \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -g/l & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1/(ml^2) \end{bmatrix} u.$$

This agrees with our earlier expression (1.26), obtained by simply using the small-angle approximation $\sin \theta \approx \theta$.

Evaluating at the equilibrium point $\mathbf{x}_0 = (\pi, 0)$, and defining $\varphi = \theta - \pi$ (so that (φ, ω) is the displacement from the equilibrium point $(\pi, 0)$), the equations linearized about the pendulum-up position are then

$$\frac{d}{dt} \begin{bmatrix} \varphi \\ \omega \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ +g/l & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1/(ml^2) \end{bmatrix} u. \quad \diamond$$

Example 1.11 (Cart on a seesaw). Consider a cart moving on the surface of a seesaw, as shown in the following diagram:



We derive the equations of motion using a Lagrangian approach. (For readers unfamiliar with Lagrangian mechanics, see standard references such as [13, 14, 17].) The position of the center of mass of the cart is

$$\mathbf{q} = (h \sin \theta + x \cos \theta, h \cos \theta - x \sin \theta),$$

from which it is straightforward to calculate

$$|\dot{\mathbf{q}}|^2 = (\dot{x} + h\dot{\theta})^2 + x^2\dot{\theta}^2.$$

Treating the cart as a point mass with mass $m > 0$, and letting the moment of inertia of the seesaw about the pivot be $J > 0$, the overall kinetic energy is given by

$$T = \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m[(\dot{x} + h\dot{\theta})^2 + x^2\dot{\theta}^2].$$

To determine the potential energy of the system (cart plus seesaw), let M be the mass of the seesaw, and assume that its center of mass (without the cart) is located a distance l above the pivot. Then the overall potential energy is

$$V = Mgl \cos \theta + mg(h \cos \theta - x \sin \theta).$$

Altogether, the Lagrangian is a function of $(x, \theta, \dot{x}, \dot{\theta})$ and is given by $L = T - V$, and the equations of motion are given by the Euler-Lagrange equations

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = \frac{\partial L}{\partial x} - d\dot{x} + F, \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} = \frac{\partial L}{\partial \theta},$$

where F denotes an external force on the cart (the control input) and $d\dot{x}$ is a damping term. Evaluating the partial derivatives of L , these equations then become

$$\begin{aligned} \frac{d}{dt}(m(\dot{x} + h\dot{\theta})) &= m\ddot{x} + m\dot{\theta}^2 + mg \sin \theta - d\dot{x} + F \\ \frac{d}{dt}(J\dot{\theta} + mh(\dot{x} + h\dot{\theta}) + mx^2\dot{\theta}) &= (Mgl + mgh) \sin \theta + mgx \cos \theta. \end{aligned}$$

These equations involve seven parameters (J, m, M, h, l, g, d) , but if we nondimensionalize, we can reduce the number of parameters to four. In particular, a natural length scale is $L = \sqrt{J/m}$ (note that this is preferable to using h or l as length scales, since either of these may be zero, while J and m are never zero). Introducing new dimensionless variables

$$\tilde{x} = \frac{x}{L}, \quad \tilde{t} = t\sqrt{g/L}, \quad u = \frac{F}{mg},$$

and defining dimensionless parameters

$$a := \frac{h}{L}, \quad b := \frac{l}{L}, \quad \mu := \frac{M}{m}, \quad \delta := \frac{d}{m\sqrt{g/L}},$$

and now letting $(\dot{})$ denote $d/d\tilde{t}$, the dimensionless equations become

$$\begin{aligned} \frac{d}{d\tilde{t}}(\dot{\tilde{x}} + a\dot{\theta}) &= \tilde{x}\dot{\theta}^2 + \sin\theta - \delta\dot{\tilde{x}} + u \\ \frac{d}{d\tilde{t}}(a\dot{\tilde{x}} + (1+a^2)\dot{\theta} + \tilde{x}^2\dot{\theta}) &= (a+\mu b)\sin\theta + \tilde{x}\cos\theta. \end{aligned}$$

(Henceforth, we will drop the $\tilde{}$, and write simply x for \tilde{x} .) There is an equilibrium for $(x, \theta, \dot{x}, \dot{\theta}) = (0, 0, 0, 0)$ and $u = 0$, corresponding to the seesaw balanced with the cart in the center. The linearization about this equilibrium point may then be found by approximating $\sin\theta \approx \theta$, $\cos\theta \approx 1$, and neglecting the cubic terms, to obtain

$$\begin{aligned} \ddot{x} + a\ddot{\theta} &= \theta - \delta\dot{x} + u \\ a\ddot{x} + (1+a^2)\ddot{\theta} &= (a+\mu b)\theta + x. \end{aligned}$$

Solving for \ddot{x} and $\ddot{\theta}$ (for instance, by subtracting a times the second equation from $1+a^2$ times the first equation; and by subtracting a times the first equation from the second equation), we obtain

$$\begin{aligned} \ddot{x} &= -ax + (1-\mu ab)\theta - (1+a^2)\delta\dot{x} + (1+a^2)u \\ \ddot{\theta} &= x + \mu b\theta + a\delta\dot{x} - au. \end{aligned}$$

Finally, we may write these equations in matrix form, as in (1.24a):

$$\frac{d}{d\tilde{t}} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a & 1-\mu ab & -(1+a^2)\delta & 0 \\ 1 & \mu b & a\delta & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1+a^2 \\ -a \end{bmatrix} u. \quad (1.45)$$

We will see that this linear system is unstable, and in Example 6.15 we will design a feedback controller to stabilize it. \diamond

Exercises

- 1.1.** Describe a feedback system that you have encountered in your everyday environment. Identify the sensing mechanism, the actuation mechanism, and control law. Describe the role that feedback serves: for instance, the uncertainty to which the feedback system provides robustness, and/or the dynamics that the feedback changes.
- 1.2.** The equation (1.14) for the dynamics with a pure time delay has the form

$$y_{j+1} = \lambda(y_j - r_j).$$

Show that, if r_j is a constant r , then the solution is

$$y_j = \lambda^j y_0 - \frac{\lambda - \lambda^{j+1}}{1 - \lambda} r.$$

For what values of λ does y_j reach a steady state (as $j \rightarrow \infty$)? Investigate the behavior of these solutions for the cruise control example from Section 1.1, for $r = 1$, and various different values of K .

- 1.3.** Write the state-space form of the system described in Example 1.4.
- 1.4.** The following system of second-order differential equations describes the dynamics of two masses coupled by springs and dampers:

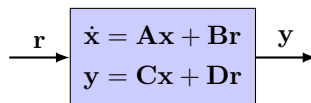
$$\begin{aligned} m_1 \ddot{q}_1 &= -k_1 q_1 + k_2 (q_2 - q_1) - b_1 \dot{q}_1 + b_2 (\dot{q}_2 - \dot{q}_1) \\ m_2 \ddot{q}_2 &= -k_2 (q_2 - q_1) - b_2 (\dot{q}_2 - \dot{q}_1) + u \end{aligned} \quad (1.46)$$

where q_1 and q_2 are the positions of the two masses, m_1 and m_2 , k_1 and k_2 are spring constants, b_1 and b_2 are damping coefficients, and u is the force applied to the second mass.

Letting $\mathbf{u} = u$ and $\mathbf{y} = (q_1, q_2)$, write the above equations in state-space form (1.24) for each of the two choices of state vector \mathbf{x} :

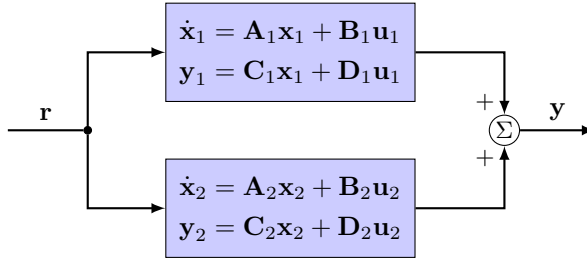
- (a) $\mathbf{x} = (q_1, q_2, \dot{q}_1, \dot{q}_2)$
 (b) $\mathbf{x} = (q_1, q_2 - q_1, \dot{q}_1, \dot{q}_2 - \dot{q}_1)$

- 1.5.** For each of the system interconnections below, rewrite the interconnected system as a single state-space system as in the diagram below:

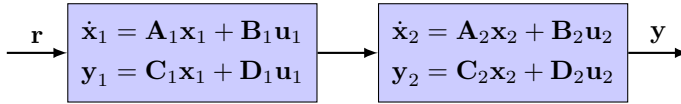


Use the state vector $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$.

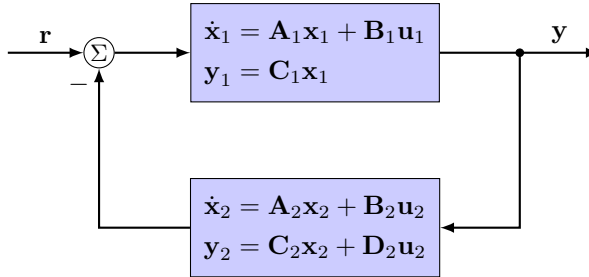
(a) Parallel interconnection



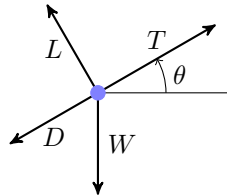
(b) Series interconnection.



(c) Feedback interconnection.



1.6. Consider a simple model of the dynamics of an aircraft, treating it as a point-mass in two dimensions. A free body diagram of the forces acting on the aircraft is shown below:



Here, T is thrust, L is lift, D is drag, W is weight, and θ is the aircraft's pitch angle. If we assume that $D = \delta v^2$ and $L = C v^2$ (where v is the speed and δ, C are positive constants), then the equations of motion

are

$$\begin{aligned}\dot{v} &= -\frac{\delta v^2}{m} - g \sin \theta + \frac{T}{m} \\ \dot{\theta} &= \frac{C}{m}v - \frac{g}{v} \cos \theta.\end{aligned}$$

Here, m and g are positive constants (mass and gravitational acceleration). Take the state vector to be $\mathbf{x} = (v, \theta)$ and the control to be T .

- (a) Assume that we wish to track the equilibrium $(v, \theta) = (v_e, 0)$, for some equilibrium thrust $T = T_e$. Find conditions on v_e and T_e that give an equilibrium.
- (b) Find the linearization $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u$ about this equilibrium, where $\mathbf{z} = (v - v_e, \theta)$ and $u = T - T_e$.

1.7. Consider the nonlinear differential equation

$$\ddot{x} + x(1 - x) = u,$$

where u is a control input.

- (a) Write the equation in first-order form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u)$, and find all equilibrium points of the system, for $u = 0$.
- (b) Linearize the system about each equilibrium point, expressing the result in the standard state-space form

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{A}\mathbf{z} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{z} + Du\end{aligned}$$

where the output y is $x - x_0$, where x_0 is the value of x at the equilibrium point, and \mathbf{z} is a state you are free to choose.

1.8. Consider a system described by the following equations:

$$\begin{aligned}\dot{x}_1 &= x_1 - 2x_1x_2 + u, \\ \dot{x}_2 &= x_1x_2 - x_2,\end{aligned}$$

where $\mathbf{x} = (x_1, x_2)$ is the state and u is an input.

- (a) Find all equilibrium points for $u = 0$.
- (b) For each equilibrium point $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2)$, find the linearization of the system about the equilibrium. Express your results in state-space form, $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u$, where $\mathbf{z} = \mathbf{x} - \bar{\mathbf{x}}$. Also give the output equation $y = \mathbf{C}\mathbf{z} + \mathbf{D}u$, where the output is $y = x_2 - \bar{x}_2$ (that is, the displacement of coordinate x_2 from its equilibrium value).

Chapter 2

Transfer functions

We now focus our attention on *linear* systems. One of the many advantages of working with linear systems is that there is a convenient way of representing them, called the transfer function. We will define the transfer function in several different ways in this chapter. First, in Section 2.1, we define the transfer function through the frequency response, which gives a nice intuitive picture of how to interpret a transfer function and why it is useful. Next, in Sections 2.2–2.3, we give a more standard, mathematical definition in terms of the Laplace transform of the impulse response.

2.1 Frequency response

The most important property of linear systems is the *principle of superposition*. There are several different versions of this principle for linear input-output systems (see, for instance, Exercise 5.1), but the common idea is that if we have two solutions of a linear system, then we may add them to obtain a new solution.

Principle of superposition: If an input $u_1(t)$ produces the output $y_1(t)$, and an input $u_2(t)$ produces the output $y_2(t)$, then an input $u_1(t) + u_2(t)$ produces the output $y_1(t) + y_2(t)$.

This simple property of linear systems enables the use of many powerful techniques, such as Laplace transforms, and transfer functions.

Sinusoidal inputs

The next most important property of a linear system is the key idea behind a frequency response:

Gain	0.1	0.5	1	2	10	100	1000
dB	-20	-6	0	6	20	40	60

Table 2.1: Common conversions between magnitude and decibels.

If the input to a linear system is $\sin \omega t$, then the output is $A \sin(\omega t + \varphi)$, after transients have decayed.

That is, if the input is sinusoidal, then the output is also perfectly sinusoidal, with the same frequency (ω), but with a possible change in amplitude (A , usually called the *gain*) and phase (φ). This property actually follows directly from the principle of superposition, as we will see in Section 2.2.

Note that sinusoidal inputs are quite special, and the above property does *not* hold for most other inputs: for instance, if the input to a linear system is a square wave, the output will usually not be a square wave. The frequency response summarizes these effects on a sinusoidal input $u = \sin \omega t$ as the frequency ω is varied.

Definition 2.1. The *frequency response* of a linear system is the relationship between the gain and phase of a sinusoidal input and the corresponding steady-state (sinusoidal) output.

The frequency response is thus given by the gain A and the phase φ , both of which depend on the frequency ω of the sinusoidal input. It is often convenient to plot the gain on a log-log scale and the phase on a semilog scale. The resulting figure is called a *Bode plot*, in honor of Hendrik Bode [3]. Note that the Bode plot is really two plots (one for gain, one for phase).

Another convention is to specify the gain in units called *decibels* (dB), which are defined as follows:

$$A \text{ (dB)} = 20 \log_{10} A.$$

For instance, a unity gain (no amplification or attenuation) corresponds to 0 dB, and a gain of 10 corresponds to 20 dB. Another handy value to know is that a gain of 2 is about 6 dB. Since decibels are a logarithmic scale, if the gains multiply, the values in decibels add. Thus, a factor of 100 is $20+20 = 40$ dB, and a factor of 200 is $40+6 = 46$ dB. Some useful conversions from amplification to dB are given in Table 2.1.

Example 2.2. Consider a spring-mass system, as in Example 1.2, where now we will apply a sinusoidal force $u(t) = \sin \omega t$. Here, we will choose $m = 0.5$ kg, $b = 0.12$ N/(m/s), and $k = 0.5$ N/m. The undamped natural frequency is then $\sqrt{k/m} = 1$ rad/sec.

We first try forcing at a frequency $\omega = 0.33$ rad/sec, well below the natural frequency. The results are shown in the leftmost plot in Figure 2.1. Note that

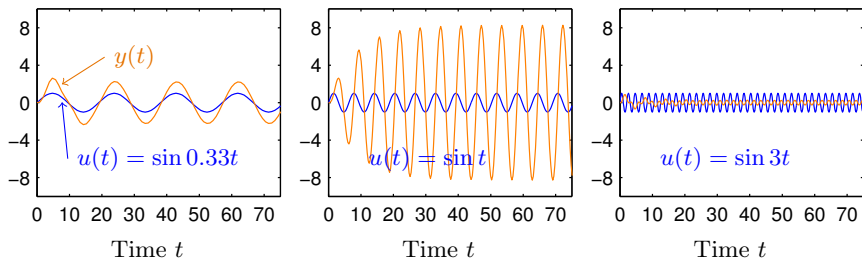


Figure 2.1: Input force $u(t)$ and output displacement $y(t)$ for Example 2.2, for three different sinusoidal forcing frequencies. Note that the output is always sinusoidal, once transients have decayed, but the amplification factor and phase shift depend on frequency.

the output $y(t)$ (the position of the mass) is also sinusoidal, with the same frequency as the input. Here, the amplitude of the output is about twice that of the input (a gain of about 6dB), and the two signals are almost perfectly in phase with one another: their peaks occur at the same point in time. We can measure this amplitude and phase and plot the corresponding point on a frequency response plot (Bode plot), shown in Figure 2.2, to obtain the leftmost orange points (one on the gain plot, and one on the phase plot).

If we increase the forcing frequency to $\omega = 1$ rad/sec, exactly at the undamped natural frequency, we see the response shown in the middle plot of Figure 2.1. Now the amplitude of the output $y(t)$ is much greater, because we are forcing near resonance, and there is a noticeable phase shift: the output y lags the input u by about a quarter of a period (90°). The corresponding points on the Bode magnitude and phase plots are also shown as orange circles in Figure 2.2.

If we increase the forcing frequency beyond the natural frequency, to $\omega = 3$ rad/sec, we see the amplitude of the output become much smaller, and the two signals are nearly perfectly out of phase (180° phase shift). The reason for this is that the forcing is oscillating so fast that the mass does not have time to accelerate before the direction of the force changes, so the mass does not move much. The higher the frequency, the more prominent the effect. This effect of the amplitude decreasing more and more at high frequencies is a characteristic of virtually all real systems, and we say that the gain “rolls off” for high frequencies.

The full frequency response for this system is shown in Figure 2.2. To obtain this plot, one could do many many numerical experiments similar to the three we have shown here, forcing at different frequencies, and plotting the resulting amplitude and phase. However, we will learn a much more efficient way to plot this frequency response once we learn about transfer functions. \diamond

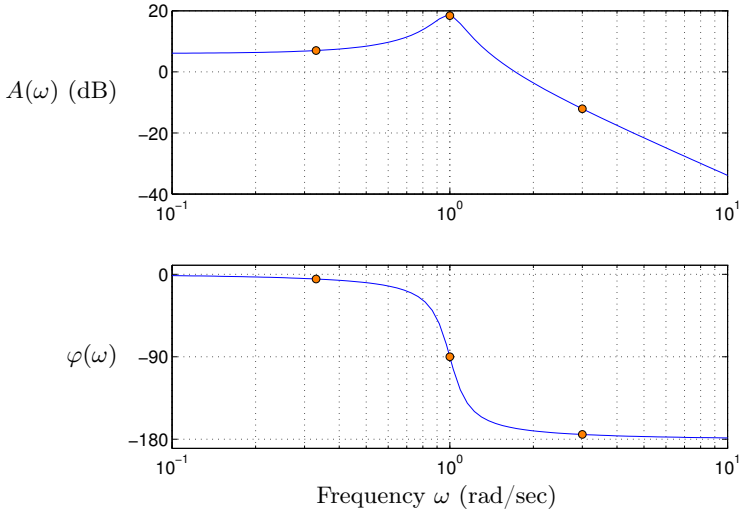


Figure 2.2: Frequency response (Bode plot) for Example 2.2, showing the amplification (gain) A and phase φ as a function of frequency. The values corresponding to the three frequencies shown in Figure 2.1 are shown as circles.

Complex inputs

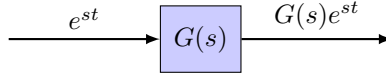
In order to define the transfer function, we will generalize things slightly, and consider inputs $u(t)$ that are complex. This might seem strange at first, for if $u(t)$ is some physical quantity such as a voltage or a force, how can a voltage or a force possibly be a complex number? Well, using a complex input is just a mathematical convenience, but it makes sense precisely because of superposition: for a complex input $u(t) = u_r(t) + iu_i(t)$ (where u_r and u_i are both real), if the output is $y(t) = y_r(t) + iy_i(t)$, then because of superposition¹, this means that the response to the real input $u_r(t)$ is $y_r(t)$, and similarly the response to the real input $u_i(t)$ is $y_i(t)$. In essence, when we consider complex inputs, we are solving two problems at once: one for the real part and one for the imaginary part.

We mentioned earlier that if the input to a linear system is sinusoidal, the output is also sinusoidal. It turns out we can generalize this statement to complex exponential inputs: if the input to a linear system is e^{st} , where s is complex (denoted $s \in \mathbb{C}$), then the output is Ge^{st} , where $G \in \mathbb{C}$. We will show this explicitly in Section 2.2, but for now let us just accept this as being true.

¹For instance, consider the superposition $u + \bar{u} = 2u_r$, for which the response is $y + \bar{y} = 2y_r$. Here, $\bar{u} = u_r - iu_i$ denotes the complex conjugate of u , and similarly for y .

We are now ready for our first definition of the transfer function.

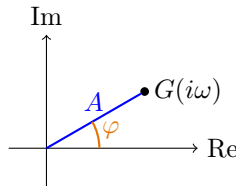
Definition 2.3. The *transfer function* of a linear system is a function $G(s)$, for $s \in \mathbb{C}$, such that an input $u(t) = e^{st}$ gives an output $y(t) = G(s)e^{st}$.



If these complex exponential inputs still seem strange, note that sinusoidal inputs are simply a special case of these, with $s = i\omega$. Suppose the input is

$$u(t) = e^{i\omega t} = \cos \omega t + i \sin \omega t,$$

and suppose $G(i\omega) = Ae^{i\varphi}$. That is, A and φ are simply the magnitude and phase of the complex number $G(i\omega)$, as shown in the following diagram:



Then from Definition 2.3, the output is

$$\begin{aligned} y(t) &= G(i\omega)e^{i\omega t} = Ae^{i\varphi}e^{i\omega t} \\ &= Ae^{i(\omega t + \varphi)} \\ &= A \cos(\omega t + \varphi) + iA \sin(\omega t + \varphi). \end{aligned}$$

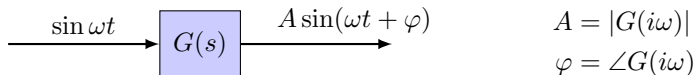
Considering only the real part, we see that the response to the input $\cos \omega t$ is $A \cos(\omega t + \varphi)$, and similarly, considering the imaginary part, we see that the response to an input $\sin \omega t$ is $A \sin(\omega t + \varphi)$. Both have an amplification factor (gain) of A , and a phase shift of φ , where

$$A = |G(i\omega)|, \quad \varphi = \angle G(i\omega),$$

where $|z|$ denotes the absolute value (magnitude) of a complex number z , and $\angle z$ denotes the argument (angle) of z .

Thus, we could define an equivalent definition of the transfer function as follows, that relates more directly to the idea of a frequency response.

Definition 2.4. The *transfer function* of a linear system is a function $G(s)$ such that $G(i\omega)$ gives the gain and phase of the response to a sinusoid at frequency ω .



Example 2.5 (Integrator). Consider an integrator, given by

$$y(t) = \int_0^t u(\tau) d\tau.$$

We claim that the transfer function for an integrator is

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

To check this, consider a sinusoidal input:

$$u(t) = \cos \omega t \implies y(t) = \frac{1}{\omega} \sin \omega t = \frac{1}{\omega} \cos(\omega t - \pi/2). \quad (2.1)$$

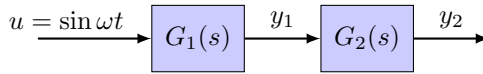
Thus, at frequency ω , the gain is $1/\omega$, and the phase is $-\pi/2$. Furthermore,

$$G(i\omega) = \frac{1}{i\omega} = \frac{1}{\omega}(-i) = \frac{1}{\omega}e^{-i\pi/2},$$

which has magnitude $1/\omega$ and phase $-\pi/2$. Thus, according to Definition 2.4, $G(s) = 1/s$ is indeed the transfer function for an integrator. \diamond

Interconnection

One of the wonderful properties of transfer functions is how they behave when connecting different input-output systems together. Consider a series interconnection of two systems, as in the diagram below:



Since the input to G_1 is $u = \sin \omega t$, we see from Definition 2.4 that

$$y_1(t) = A_1 \sin(\omega t + \varphi_1), \quad G_1(i\omega) = A_1 e^{i\varphi_1}.$$

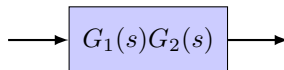
This is then a sinusoidal input to G_2 , which applies another gain and phase shift:

$$y_2(t) = A_2 A_1 \sin(\omega t + \varphi_1 + \varphi_2), \quad G_2(i\omega) = A_2 e^{i\varphi_2}.$$

Altogether, we see the gains A_1 and A_2 *multiply*, and the phases φ_1 and φ_2 *add*. This is precisely how complex numbers behave when multiplied:

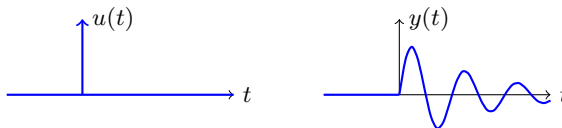
$$\begin{aligned} z_1 &= r_1 e^{i\theta_1} \\ z_2 &= r_2 e^{i\theta_2} \implies z_1 z_2 = r_1 r_2 e^{i(\theta_1 + \theta_2)}. \end{aligned}$$

(For a review of basic properties of complex numbers, see Appendix B.1.) So the transfer functions simply multiply, and the interconnected system is equivalent to



2.2 Impulse response

Consider the response of a linear system to a unit impulse $\delta(t)$. This is analogous to “hitting the system with a hammer”: the instantaneous force is very large, but the duration is very short, so a finite amount of work is done. If we imagine a spring-mass system, the impulse response might look like the following:



This impulsive input is the *Dirac delta function* $\delta(t)$, defined by the property

$$\int_a^b \delta(t) dt = \begin{cases} 0 & 0 \notin (a, b) \\ 1 & 0 \in (a, b). \end{cases} \quad (2.2)$$

Thus, $\delta(t)$ is zero everywhere except at $t = 0$, and yet the “area” under the curve $\delta(t)$ is 1. The value of $\delta(0)$ is not actually defined: intuitively, one can think of this as $\delta(0) = +\infty$, although of course this is not a proper mathematical definition of a function. In fact, $\delta(t)$ is *not* an ordinary function: it is a type of generalized function called a *distribution*. The way to think about such an object is that it makes sense only inside an integral. Another intuitive way to think about $\delta(t)$ is as the derivative of the unit *step function*

$$\mathbb{1}(t) := \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

(which is of course not differentiable at $t = 0$). We will not be concerned with how precisely to define $\delta(t)$; just understanding its properties, such as (2.2), will be enough for us. Another important property of the Dirac delta function is its *sifting property*: if a function $f(t)$ is continuous at t_0 , then

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0). \quad (2.3)$$

That is, all other values of f at points other than t_0 are irrelevant (since $\delta(t - t_0) = 0$ unless $t = t_0$), and the integral “sifts out” $f(t_0)$. This property actually follows from (2.2), and this is the main property we will use.

We now come to the main definition of this section:

The *impulse response* $h(t)$ is the output $y(t)$ for an input $u(t) = \delta(t)$ (with zero initial conditions).

Why is this useful? It turns out we can build up solutions for *arbitrary* inputs from the impulse response, using superposition. We can write any function as a sum of impulses. Then any output can be written as a sum of impulse responses, by superposition.

From the sifting property of the delta function (2.3), we have, for any input $u(t)$,

$$u(t) = \int_{-\infty}^{\infty} u(\tau) \delta(t - \tau) d\tau.$$

One can think of this integral as a sum of an infinite number of spikes (delta functions) at time τ , each scaled by $u(\tau)$. Now, the response to a single delta function $\delta(t - \tau)$ is simply $h(t - \tau)$, so by superposition, the output will be

$$y(t) = \int_{-\infty}^{\infty} u(\tau) h(t - \tau) d\tau. \quad (2.4)$$

This integral is called the *convolution* of u and h , and is often denoted $y(t) = (u \star h)(t)$. Note that, changing the variable of integration to $t - \tau$, the integral in (2.4) may be written equivalently as

$$y(t) = \int_{-\infty}^{\infty} h(\tau) u(t - \tau) d\tau. \quad (2.5)$$

Causality

Note that for a real physical system, the impulse response $h(t)$ must be zero for $t < 0$. This is intuitively obvious, if you think of the impulse response as “hitting the system with a hammer” at $t = 0$: the system does not know that it is about to be hit by the hammer until it is actually hit, so it cannot respond before $t = 0$. This property is called *causality*.

Definition 2.6. A *causal system* is one for which the impulse response $h(t) = 0$ for $t < 0$.

Thus, from (2.5), we see that for a causal system, the response $y(t)$ to an arbitrary input $u(t)$ is given by

$$y(t) = \int_0^{\infty} h(\tau) u(t - \tau) d\tau. \quad (2.6)$$

Note that the response depends only on *past* inputs ($u(t)$ for $\tau \leq t$), not on future inputs. All of the systems we will be interested in will be causal.



One might wonder when non-causal systems could possibly arise. In some applications, such as signal processing, they can be quite useful. For instance, one can define a low-pass filter or band-pass filter that is non-causal, so the output of the filter at any one time depends on the input signal at all times. This can be used to eliminate phase distortions, for instance. (See the MATLAB command

`filtfilt` for an example.) Such a filter makes sense if the entire input signal is known beforehand, e.g., if we are doing “post-processing,” although of course such a filter cannot be used to filter a signal in real time, when we do not know the input signal in the future.

Response to sinusoidal inputs

We can use the solution (2.6) to verify the claim we made in Section 2.1, in particular that for a linear system, the response to a sinusoidal input is sinusoidal. We will consider more generally the complex exponential input

$$u(t) = e^{st}, \quad s \in \mathbb{C},$$

recognizing that sinusoidal inputs are a special case of this, for $s = i\omega$ (as explained in Section 2.1, page 28). From (2.6), the response is then

$$\begin{aligned} y(t) &= \int_0^\infty h(\tau) e^{s(t-\tau)} d\tau \\ &= e^{st} \int_0^\infty e^{-s\tau} h(\tau) d\tau \\ &= e^{st} H(s), \end{aligned}$$

where $H(s)$ is a complex number defined by

$$H(s) = \int_0^\infty e^{-s\tau} h(\tau) d\tau. \quad (2.7)$$

This function $H(s)$ is thus the transfer function, as defined in Definition 2.3.

2.3 Laplace transforms

The Laplace transform is the main tool we use to transform a system from the *time domain* to the *frequency domain*. Most of the tools used in classical control theory (e.g., Bode plots, Nyquist plots) are used in the frequency domain, so it is important to become comfortable with this notion. We have already discussed the intuitive idea behind the frequency domain in Section 2.1, and the ideas here are not fundamentally different. But the Laplace transform lets us understand how transfer functions describe the response to *general* (non-sinusoidal or non-exponential) inputs. Furthermore, it gives us an easy way to obtain transfer functions directly from (linear) differential equations.

You have probably encountered Laplace transforms before, for instance in a course on differential equations, but here we give a brief review.

Definition 2.7. The *Laplace transform* of a function $f(t)$ is a new function $F(s)$ given by

$$F(s) = \int_0^\infty e^{-st} f(t) dt, \quad s \in \mathbb{C}. \quad (2.8)$$


Name	$f(t)$	$F(s)$
impulse	$\delta(t)$	1
step	$\mathbb{1}(t)$	$\frac{1}{s}$
exponential	e^{-at}	$\frac{1}{s+a}$

Table 2.2: Laplace transforms of some common functions.

We sometimes write $F(s) = \mathcal{L}\{f(t)\}$ for the Laplace transform. It transforms us from the “time domain” (a function of time t) into the “frequency domain” (a function of the complex variable s).

One thing we might notice right away is the following: the transfer function $H(s)$, given by (2.7), is simply the Laplace transform of the impulse response $h(t)$. This, in fact, is how the transfer function is typically defined. In Section 2.1, we took a more intuitive approach to the transfer function, based on the frequency response, but we now see that the two are equivalent.

Laplace transforms of some common functions are shown in Table 2.2.

 One subtlety here is computing the Laplace transform of $\delta(t)$. Since the integral is from 0 to ∞ , is 0 included in the interval or not? Strictly speaking, by the definition (2.2), it seems the Laplace transform of $\delta(t)$ should be 0, since 0 is not included in the open interval $(0, \infty)$. But we wish to include zero, so the Laplace transform is sometimes written

$$F(s) = \int_{0-}^{\infty} e^{-st} f(t) dt := \lim_{\varepsilon \rightarrow 0-} \int_{\varepsilon}^{\infty} e^{-st} f(t) dt,$$

which means that we actually start a little before $t = 0$, so we include the spike when taking the Laplace transform of $\delta(t)$. This subtlety can usually be ignored, but it does arise when taking the Laplace transform of an impulse $\delta(t)$, or when computing the transfer function for a system with an impulse response that is discontinuous at $t = 0$.

Properties of the Laplace transform

Let $F(s) = \mathcal{L}\{f(t)\}$ and $G(s) = \mathcal{L}\{g(t)\}$. Then the Laplace transform satisfies the following properties:

- Linearity

$$\mathcal{L}\{\alpha f(t) + \beta g(t)\} = \alpha F(s) + \beta G(s).$$

- Differentiation

$$\mathcal{L}\left\{\frac{df}{dt}\right\} = sF(s) - f(0^-).$$

(Here, $f(0^-)$ means the limit as $t \rightarrow 0$ from below—see the note on page 34.)

- Integration

$$\mathcal{L} \left\{ \int_0^t f(\tau) d\tau \right\} = \frac{1}{s} F(s).$$

- Time delay: if $g(t) = f(t - \lambda)$, then

$$G(s) = e^{-s\lambda} F(s).$$

- Convolution: if $f(t)$ and $g(t)$ are both zero for $t < 0$, then

$$\mathcal{L} \{f \star g\} = F(s) \cdot G(s),$$

where $f \star g$ denotes the convolution defined by (2.4).

This last property, convolution, has an important (and somewhat amazing) consequence. If $h(t)$ is the impulse response, then we know from (2.6) that

$$y(t) = \int_0^\infty h(\tau) u(t - \tau) d\tau = (h \star u)(t).$$

Thus, taking the Laplace transform of this expression and using the convolution property, we see

$$Y(s) = H(s) \cdot U(s). \quad (2.9)$$

That is, the transfer function $H(s)$ relates the Laplace transform of the input $U(s)$ to the Laplace transform of the output $Y(s)$.

We saw before (Definition 2.3) that, if the input has the special form $u(t) = e^{st}$, then $y(t) = H(s)e^{st}$, where $H(s)$ is the transfer function. Equation (2.9) shows that the transfer function can be used in exactly the same way for *arbitrary* inputs $u(t)$, as long as we first transform to the frequency domain. This is one of the main reasons why transfer functions are such a useful description of a linear system.

Computing transfer functions

If we know the dynamics of a system in the form of some ODEs, it is easy to compute the transfer function using the Laplace transform. We illustrate here with some examples.

The key here is to remember the differentiation property of Laplace transforms: in particular, $\mathcal{L} \{df/dt\} = sF(s) - f(0^-)$. To determine the transfer

function, we assume that both $u(t)$ and $y(t)$ are zero for $t < 0$, so we can safely ignore the initial condition at $t = 0$, and simply replace d/dt by s . We then solve for $Y(s)$ in terms of $U(s)$ to determine the transfer function.

Example 2.8. Consider the cruise control problem considered in Example 1.1, with dynamics given by

$$m\dot{y} + by = u.$$

Assuming $y(0) = 0$ and taking Laplace transforms, we obtain

$$msY(s) + bY(s) = U(s).$$

Solving for $Y(s)$, we have

$$Y(s) = \underbrace{\frac{1}{ms + b}}_{H(s)} U(s),$$

so the transfer function is $H(s) = 1/(ms + b)$. \diamond

Example 2.9. Consider the spring-mass system from Example 1.2, whose dynamics are given by

$$m\ddot{y} + b\dot{y} + ky = u.$$

Assuming $y(0) = \dot{y}(0) = 0$ and taking Laplace transforms, we obtain

$$ms^2Y(s) + bsY(s) + kY(s) = U(s).$$

Solving for $Y(s)$, we have

$$Y(s) = \underbrace{\frac{1}{ms^2 + bs + k}}_{H(s)} U(s). \quad \diamond$$

Some common transfer functions are shown in Table 2.3, along with their corresponding impulse responses.

Transfer function for a state-space system

Suppose we have a linear state-space system of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (2.10a)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \quad (2.10b)$$

Using the Laplace transform, we can determine an explicit formula for the transfer function in terms of the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$. Taking the Laplace transform of (2.10a), with $\mathbf{x}(0) = \mathbf{0}$, we have

$$\begin{aligned} sX(s) &= \mathbf{A}X(s) + \mathbf{B}U(s) \\ \implies (s\mathbf{I} - \mathbf{A})X(s) &= \mathbf{B}U(s) \\ \implies X(s) &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s). \end{aligned}$$

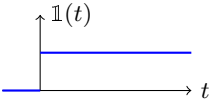
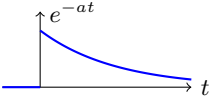
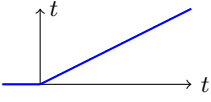
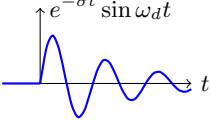
System	Transfer function	Impulse response
Integrator $\dot{y} = u$	$\frac{1}{s}$	
Differentiator $y = \dot{u}$	s	$\delta'(t)$
First-order system $\dot{y} + ay = u$	$\frac{1}{s + a}$	
Double integrator $\ddot{y} = u$	$\frac{1}{s^2}$	
Damped oscillator $\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2y = u$	$\frac{1}{s^2 + 2\zeta\omega_0s + \omega_0^2}$	
Time delay $y(t) = u(t - \tau)$	$e^{-s\tau}$	$\delta(t - \tau)$
PID controller $y = k_p u + k_i \int u + k_d \dot{u}$	$k_p + \frac{k_i}{s} + k_d s$	

Table 2.3: Common transfer functions. For the damped oscillator, $\sigma = \zeta\omega_0$ and $\omega_d = \omega_0\sqrt{1 - \zeta^2}$.

(Note that in the second equality we needed to insert an identity matrix on the left-hand side, because $s - \mathbf{A}$ does not make sense: one cannot subtract a matrix from a scalar.) Taking the Laplace transform of the output equation (2.10b) then gives

$$Y(s) = \mathbf{C}X(s) + \mathbf{D}U(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}]U(s).$$

Comparing with (2.9), we see that the transfer function of the state-space system (2.10) is

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \quad (2.11)$$

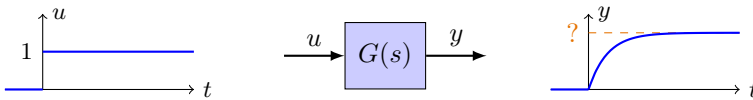
This is a useful formula, and is worth committing to memory (or at least remembering how to derive it). Let us think about the dimensions of these matrices in the common case in which both the input and the output are scalars. In this case, the \mathbf{B} matrix is a column vector (since \mathbf{B} must multiply the scalar u), and \mathbf{C} is a row vector (since $\mathbf{C}\mathbf{x}$ must also be a scalar). Thus, the dimensions of these matrices are as depicted below:

$$\boxed{H} = \boxed{\mathbf{C}} \begin{array}{c} \boxed{(s\mathbf{I} - \mathbf{A})^{-1}} \end{array} \boxed{\mathbf{B}} + \boxed{D}$$

Thus, the product $(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is a column vector, and multiplying this on the left by the row vector \mathbf{C} , one obtains a scalar. Thus, in the single-input, single-output case, the transfer function is just a scalar, which of course is what we expect, for instance from (2.9). If there are multiple inputs or multiple outputs, however, then the transfer function given by the formula (2.11) will be a matrix (as we would also expect from (2.9)).

2.4 Zero-frequency gain

It is common to want to know the final value for a step input to a system, as shown in the following diagram:



The final value to a step input is called the *zero-frequency gain* (or *DC gain*). It is useful to be able to calculate it quickly, for looking at things like the steady-state error for a step in the reference value, for a tracking problem. It turns out that it is easy to find this value directly from the transfer function.

To determine the zero-frequency gain, one uses the following theorem, which relates the steady-state behavior of a function $y(t)$ to its Laplace transform $Y(s)$:

Theorem 2.10 (Final value theorem). *If all the poles of $sY(s)$ are in the left-half plane, then*

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s). \quad (2.12)$$

Proof. From the differentiation property of Laplace transforms, we know

$$\mathcal{L} \left\{ \frac{dy}{dt} \right\} = sY(s) - y(0),$$

and thus

$$\int_0^{\infty} e^{-st} \frac{dy}{dt} dt = sY(s) - y(0).$$

Taking the limit as $s \rightarrow 0$ then gives

$$\begin{aligned} \int_0^{\infty} \frac{dy}{dt} dt &= \lim_{s \rightarrow 0} sY(s) - y(0) \\ \implies y(\infty) - y(0) &= \lim_{s \rightarrow 0} sY(s) - y(0). \end{aligned}$$

Cancelling the terms $y(0)$ gives the result. \square

Here, the *poles* of a function are values of s where the denominator goes to zero. We shall see in Section 3.1 that a linear system is stable if and only if all its poles are in the left half of the complex plane, so the condition that the poles lie in the left half-plane is needed to make sure $y(t)$ actually converges to a limit as $t \rightarrow \infty$.

We will now use the final value theorem to determine the zero-frequency gain (the final value to a unit step input). If $u(t)$ is a step function, then $U(s) = 1/s$ (see Table 2.2). Thus,

$$Y(s) = G(s) \cdot U(s) = \frac{1}{s} G(s).$$

The final value theorem then states that, as long as all the poles of $sY(s) = G(s)$ are in the left half-plane, then

$$y(\infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s \left(\frac{1}{s} G(s) \right) = \lim_{s \rightarrow 0} G(s) = G(0).$$

We have therefore shown the following:

If a transfer function $G(s)$ is stable (all poles in the left half-plane), then the zero-frequency gain is $G(0)$.

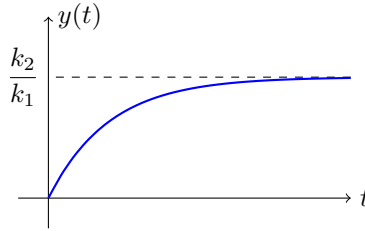
Example 2.11. Consider the dynamics of a DC motor, given by

$$\dot{y} + \frac{k_1}{J_m}y = \frac{k_2}{J_m}u,$$

where u is the voltage applied to the motor, y is the speed, and the constants k_1 , k_2 , and J_m are all positive. Taking the Laplace transform of this equation gives

$$\begin{aligned} sY + \frac{k_1}{J_m}Y &= \frac{k_2}{J_m}U \\ \implies Y &= \underbrace{\frac{k_2}{J_ms + k_1}}_{G(s)} U. \end{aligned}$$

Since $k_1, J_m > 0$, the only pole is at $s = -k_1/J_m < 0$, so is in the left half-plane. Thus, the zero-frequency gain is well defined, and has the value $G(0) = k_2/k_1$. From this, we can deduce that the step response looks something like the following:



◇

Note that setting $s = 0$ makes intuitive sense, if you recall that Laplace transforms convert derivatives d/dt into multiplication by s : so setting $s = 0$ in the frequency domain corresponds to setting derivatives equal to zero in the time domain, which is what one does to find a steady state.

If one has a Bode plot of a transfer function, then the zero-frequency gain can be seen immediately from the plot: one simply looks at the magnitude where the frequency goes to zero, as shown in Figure 2.3. Remember that on the Bode plot, frequency is plotted on a log scale, so zero frequency is actually infinitely far to the left on the plot.

2.5 Block diagrams

We will now investigate how to manipulate transfer functions when several different systems are connected together. We typically use block diagrams to illustrate the interconnection, as discussed in Section 1.1. Here we will learn some rules for manipulating and simplifying block diagrams.

We already saw in Section 2.1 how transfer functions behave when connected in series, as in the diagram below:

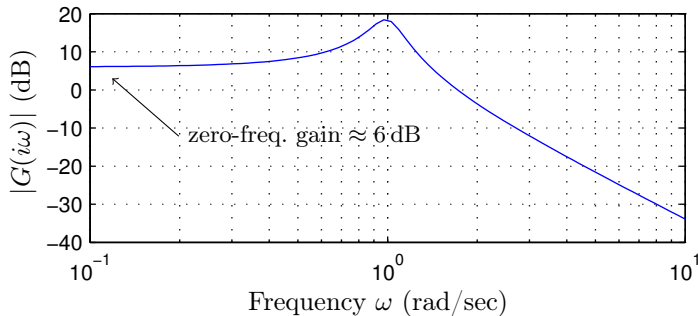
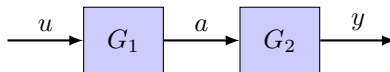


Figure 2.3: Bode magnitude plot, showing value of zero-frequency gain.



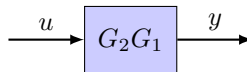
Let us have another look, now that we know about Laplace transforms, and the rule (2.9). Because $G_1(s)$ is the transfer function for the block on the left, we have

$$A(s) = G_1(s)U(s),$$

where $U(s)$ is the Laplace transform of the input $u(t)$, and $A(s)$ is the Laplace transform of $a(t)$, the output of the leftmost block. Similarly,

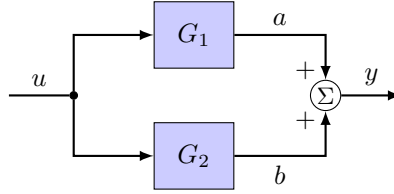
$$Y(s) = G_2(s)A(s) = G_2(s)G_1(s)U(s).$$

Thus, the transfer function of the overall system is simply $G_2(s)G_1(s)$ so the diagram is equivalent to the following:



Note that the order of G_1 and G_2 is reversed: G_1 is on the left in the original diagram, but the transfer function of the interconnection is G_2G_1 , with G_1 on the right. For single-input, single-output systems, the order does not matter, as the transfer functions are scalars, and multiplication of scalars is commutative. But if there are multiple inputs and/or outputs, these transfer functions are *matrices*, and do not commute. For these systems, one needs to be careful to keep track of the order. Fortunately, we will be concerned almost entirely with single-input single-output systems, so will not have to worry about the order. Usually.

Next, consider a parallel interconnection, as in the diagram below:



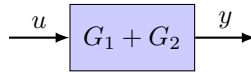
This diagram implies the following equations:

$$\begin{aligned} A(s) &= G_1(s)U(s) \\ B(s) &= G_2(s)U(s) \\ Y(s) &= A(s) + B(s). \end{aligned}$$

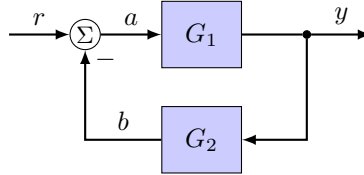
Combining, we have

$$Y(s) = (G_1(s) + G_2(s))U(s),$$

so this diagram is equivalent to the following:



The next case we will consider is the most important one of this section. Consider the feedback interconnection below:



We wish to determine the overall transfer function from r to y . The diagram implies the following equations:

$$\begin{aligned} A(s) &= R(s) - B(s) \\ B(s) &= G_2(s)Y(s) \\ Y(s) &= G_1(s)A(s). \end{aligned}$$

Starting with the last equation and combining, we have

$$Y(s) = G_1(s)A(s) = G_1(s)(R(s) - B(s)) = G_1(s)(R(s) - G_2(s)Y(s)).$$

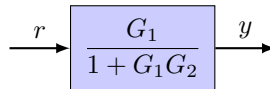
Now, we must do a little algebra to solve for Y in terms of R :

$$(1 + G_1G_2)Y = G_1R,$$

so

$$Y(s) = \frac{G_1(s)}{1 + G_1(s)G_2(s)}R(s),$$


and the overall diagram is equivalent to the following:



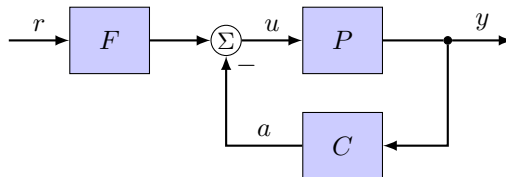
We will be using feedback interconnections a lot in this course, so it is worth remembering this formula for the transfer function of a feedback system. The quantity G_1 is called the *forward gain*, as it is the gain in the forward path from the input (here, r) to the output (here, y). The quantity G_1G_2 is called the *loop gain*, as it is the gain just around the feedback loop. A good way to remember this rule is as follows:

The transfer function for a (negative) feedback loop is given by

$$\frac{\text{Forward gain}}{1 + \text{Loop gain}}. \quad (2.13)$$

 The system in the diagram above is a *negative feedback* loop, because there is a minus sign coming into the summing junction (so the overall loop gain also has a minus sign). This is the usual convention for feedback systems, but beware that different conventions are used in different contexts (even in different MATLAB commands!), so this is something to watch out for. One can easily verify that the general rule for a positive feedback loop differs only by a sign: Forward gain/(1 – Loop gain).

Example 2.12. Let us find the overall transfer function from r to y for the following diagram:



The forward gain from r to y is FP : we just take a straight path from r to y , without going around the loop. The loop gain is PC : here, we ignore F , as it is not part of the feedback loop. We have a negative feedback loop, because of the minus sign going into the summing junction, so we can use formula (2.13) to obtain the transfer function

$$\frac{Y}{R} = \frac{FP}{1 + PC}.$$

It is probably a good idea to check this result, since we are new at applying these rules: to do this, just go back to plain old algebra. The diagram implies

$$u = Fr - a$$

$$a = Cy$$

$$y = Pu = P(Fr - a) = P(Fr - Cy).$$

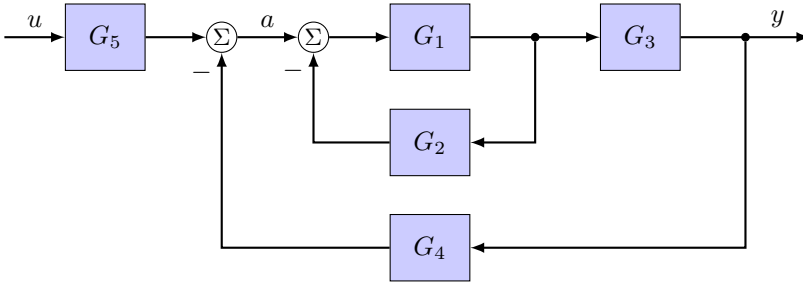
(Here, we are using a little shorthand, taking y to mean $Y(s)$, etc., in order to better distinguish between signals, which are lower case, and transfer functions, which are upper case.) Solving for y , we have

$$(1 + PC)y = PF r$$

$$\Rightarrow y = \frac{PF}{1 + PC} r,$$

which agrees with the expression we obtained using the rule (2.13). \diamond

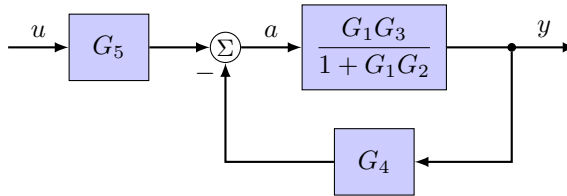
Example 2.13. Now, consider a more complicated block diagram:



What is the overall transfer function from u to y ? This is too complicated to apply our rule (2.13) all in one shot, but we can break it down one piece at a time. Note that there is an “inner loop” consisting of transfer functions G_1 , G_2 , and G_3 . We can use (2.13) to determine the overall transfer function from a to y : the forward gain is $G_1 G_3$, and the loop gain is $G_1 G_2$, so (2.13) gives

$$y = \frac{G_1 G_3}{1 + G_1 G_2} a,$$

and we may simplify the diagram to the following:



Now, this is simple enough to apply our rule (2.13) to the overall system:

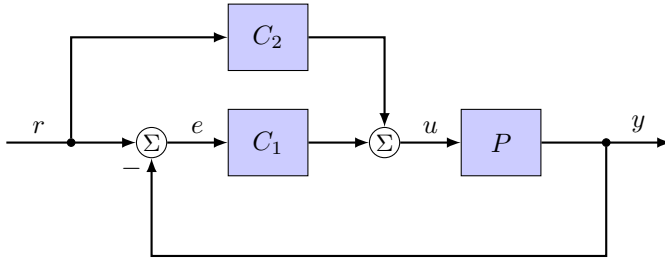
$$\text{Forward gain} = G_5 \frac{G_1 G_3}{1 + G_1 G_2}, \quad \text{Loop gain} = G_4 \frac{G_1 G_3}{1 + G_1 G_2},$$

so altogether

$$\frac{y}{u} = \frac{G_5 \frac{G_1 G_3}{1 + G_1 G_2}}{1 + G_4 \frac{G_1 G_3}{1 + G_1 G_2}} = \frac{G_1 G_3 G_5}{1 + G_1 G_2 + G_1 G_3 G_4}.$$

\diamond

Example 2.14 (2-DOF control design). Consider the following block diagram:



This is referred to as a two-degree-of-freedom (2DOF) control design. Here, P represents the *plant*, a generic term for the system we wish to control. Its input is u and its output is y , as is the usual convention. The control designer can choose C_1 and C_2 to achieve some desired behavior, and here the goal is to make the output y track the reference input r . In the diagram, $e = r - y$ represents the tracking error.

We wish to find the transfer function from r to y . The forward gain is now $PC_1 + PC_2$, and the loop gain is PC_1 , so applying our rule (2.13), we find the overall transfer function

$$\frac{y}{r} = \frac{P(C_1 + C_2)}{1 + PC_1}.$$

Again, we might not be so confident that the rule applies here, since the output of C_2 enters into the middle of the loop. We can once again check our result by going back to the algebra. From the diagram,

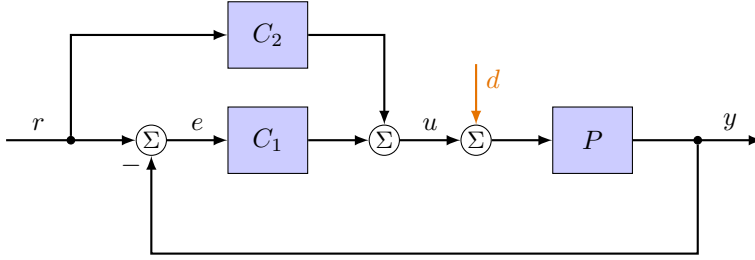
$$\begin{aligned} u &= C_2 r + C_1 e \\ e &= r - y \\ y &= P u = P(C_2 r + C_1 e) = P(C_2 r + C_1 r - C_1 y). \end{aligned}$$

So

$$\begin{aligned} (1 + PC_1)y &= P(C_1 + C_2)r \\ \implies y &= \frac{P(C_1 + C_2)}{1 + PC_1}r, \end{aligned} \tag{2.14}$$

which agrees with our previous expression. \diamond

Example 2.15. Now, let us add a disturbance d to the previous diagram:



What is the transfer function from d to y ? (For now, we will ignore r , assuming $r = 0$.) The loop gain is the same as before, PC_1 , and now the forward gain is P , so applying (2.13) we have

$$y = \frac{P}{1 + PC_1} d. \quad (2.15)$$

What if we have *both* a reference input r and a disturbance d ? It turns out we can consider these separately, because of superposition: the response to a reference input r and a disturbance d is simply the sum of the response to an input r with no disturbance, and the response to a disturbance d with no reference input. Again, we can verify this with algebra: from the diagram,

$$u = C_2 r + C_1 e$$

$$e = r - y$$

$$y = P(u + d) = P(C_2 r + C_1 e + d) = P(C_2 r + C_1 r - C_1 y + d).$$

Solving for y as before then gives

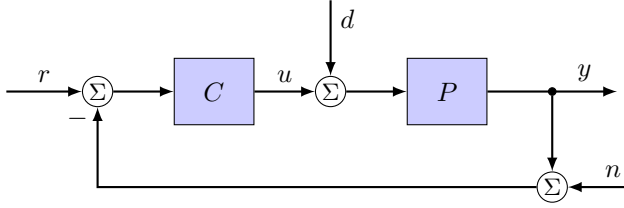
$$y = \frac{P(C_1 + C_2)}{1 + PC_1} r + \frac{P}{1 + PC_1} d.$$

This is just the sum of the responses (2.14) (for $d = 0$) and (2.15) (for $r = 0$). This property simplifies the task of finding transfer functions for systems with more than one input: it is okay to consider only one input at a time, setting the others to zero.

Note that C_2 does not appear in the disturbance term: disturbances are affected only by the loop gain (PC_1). The usual design procedure for a 2DOF controller is therefore to first design C_1 for good disturbance rejection, then design C_2 for good reference tracking. \diamond

2.6 Gang of four

Consider the diagram below, which is typical of a reference tracking control system:



As before, P represents the plant, the system to be controlled. The inputs to the plant are a disturbance d (which we cannot control), and a control input u (which we can control), and the output of the plant is y . The task of the control engineer is to design a controller C (also a transfer function) to achieve some design objective. Here, the objective is to make the output y track a reference input r , which the operator specifies: it is not known in advance, although we may know something about it, such as the typical frequencies that we are expected to track well. We have also added an input n , which represents sensor noise.

We are interested in determining the relevant transfer functions for the closed-loop system. The system has 3 different inputs (r , d , and n), and there are 3 different output quantities we might be interested in (y , u , and the tracking error $e = r - y$). Altogether these make 9 different transfer functions: from r to e , from r to u , from r to y , from d to e , etc. From the previous section, we now know how to find all of these transfer functions. The loop gain is PC (for any input/output pair), but the forward gain depends on the input and output.

As in Example 2.15, we can determine each transfer function separately (i.e., assuming all of the other inputs are zero), and then combine them using superposition. For instance, the forward gain from r to y is PC , from d to y is P , and from n to y is $-PC$, so applying (2.13), we have

$$y = \frac{PC}{1+PC}r + \frac{P}{1+PC}d + \frac{-PC}{1+PC}n. \quad (2.16a)$$

Similarly, we can see how each of the inputs affects the signal u our controller gives to the plant: the forward gain from r to u is C , from n to u is $-C$, and from d to u is $-PC$, so altogether,

$$u = \frac{C}{1+PC}r + \frac{-PC}{1+PC}d + \frac{-C}{1+PC}n. \quad (2.16b)$$

The tracking error is then given by $e = r - y$, which becomes

$$e = \frac{1}{1+PC}r + \frac{-P}{1+PC}d + \frac{PC}{1+PC}n. \quad (2.16c)$$

We see that there are four different transfer functions that appear here (apart from sign differences). The most important of these are given special names:

$$\begin{aligned}
 S &= \frac{1}{1+PC} && \text{Sensitivity function} \\
 T &= \frac{PC}{1+PC} && \text{Complementary sensitivity function.}
 \end{aligned}
 \tag{2.17}$$

The *sensitivity function* S is the closed-loop transfer function from the reference input to the tracking error, so we would like it to be small, at least over the range of frequencies we are interested in tracking. (More precisely, this means we would like $|S(i\omega)|$ to be small for frequencies ω we are interested in.)

The *complementary sensitivity function* T is the closed-loop transfer function from r to y . Note that $S + T = 1$ (this is the sense in which S and T are complementary). Normally we would like T to be close to 1, for good reference tracking. However, note that the transfer function from n to e is also T , so at high frequencies, where we may be concerned about sensor noise, we typically want T to be *small*, and not close to 1 (since we do not want to track the sensor noise).

Let us take a short digression to explain why S called the sensitivity function. It turns out S measures how sensitive the closed-loop system is to variations in the plant (or, more precisely, the overall loop gain), relative to the open-loop system. To see this, let $L = PC$ be the loop gain, so $S = 1/(1 + L)$ and $T = L/(1 + L)$. Suppose L depends on some parameter μ , and let $L' = dL/d\mu$. Then the closed-loop transfer function T will also depend on μ , and we would like to determine how sensitive T is to a change in μ . Differentiating $T(1 + L) = L$ with respect to μ , and letting $T' = dT/d\mu$, we have

$$\begin{aligned}
 T'(1 + L) + TL' &= L' \\
 \stackrel{\div L}{\implies} \frac{T'}{T} &= (1 - T) \frac{L'}{L}.
 \end{aligned}$$

Since $S + T = 1$, this expression becomes

$$\frac{T'}{T} = S \frac{L'}{L}. \tag{2.18}$$

The quantity T'/T can be thought of as the fractional change in the closed-loop system T as the parameter μ varies (infinitesimally), while L'/L is the fractional change in the open-loop transfer function L . Thus, equation (2.18) shows that the change in the closed-loop system is just S times the change in the open-loop system. For instance, if the fractional change in the loop gain is $L'/L = 10\%$ and $S = 0.1$, then the fractional change in the closed-loop system is $T'/T = 1\%$. If $|S| < 1$, feedback has a *beneficial* effect: it reduces the sensitivity to changes in the plant P (or our controller C). However, if

$|S| > 1$, then feedback has a *harmful* effect: the closed-loop system is more sensitive to changes in the plant than the open-loop system. This explains one of the main principles of feedback discussed in Section 1.1: namely, feedback can reduce the overall sensitivity of a system to changes in components:

Feedback modifies the sensitivity of a system by a factor $S = \frac{1}{1+L}$. That is, if $|S| < 1$, then the closed-loop system is less sensitive to a change in L than the open-loop system, by a factor $|S|$.

Returning to our previous discussion, we see that there are two other transfer functions that appear in our expressions (2.16):

$$\begin{aligned} PS &= \frac{P}{1+PC} && \text{Load sensitivity function} \\ CS &= \frac{C}{1+PC} && \text{Noise sensitivity function.} \end{aligned} \tag{2.19}$$

The *load sensitivity function* PS is the transfer function from d to y , and represents the effect of disturbances on the closed-loop system. The *noise sensitivity function* CS is (minus) the transfer function from n to u , and represents how much sensor noise is transmitted to our control signal.

Åström and Murray [1] refer to the four transfer functions given by (2.17) and (2.19) as the *gang of four*. They will play an important role in designing controllers, as we will see in Chapter 4.

To give a taste for what lies ahead, let us briefly consider the tradeoffs between these. In order to make $|S|$ small, it makes sense to choose our controller C to be large, so that PC will be large, and $S = 1/(1+PC)$ will be small. This is precisely analogous to the cruise control example considered in Section 1.1, in which we wanted to make the gain K large for good performance (see equation (1.6)).

But now let us consider what this does to the other transfer functions. In particular, consider the noise sensitivity function, which describes the transfer function from n to u . We mentioned in Section 2.1 that all real systems “roll off” at high frequencies: this means that $|P(i\omega)| \rightarrow 0$ as $\omega \rightarrow \infty$. This means that $S = 1/(1+PC) \rightarrow 1$ as $\omega \rightarrow \infty$, so the noise sensitivity function $CS \approx C$ for high frequencies. If C is large, then we are greatly amplifying our sensor noise, and sending this signal to our plant. To put this in perspective, suppose the units of y and u are both volts, and suppose our sensor has a noise of amplitude 1 mV. If we choose $C = 10^6$ (a nice large value), then we may get very good tracking, but at high frequencies, the value u we are sending to the plant will have a high-frequency noise level of 1000 V! This is totally unacceptable in most applications: in fact, even if we had a power

supply large enough to supply 1000 V, large noise levels in u are likely to wear out our actuator, so it is important to keep these small.

The solution is to choose C such that its gain *depends on frequency*. In particular, we typically choose C such that $|C(i\omega)|$ is *large* over some frequency range over which we wish to have good tracking (e.g., low frequencies), and *small* for other frequencies (e.g., high frequencies) where we do not wish to amplify sensor noise. This sounds like a simple task, but there is a hitch: we must also consider *dynamics*, so that we do not inadvertently destabilize our system with feedback. In order to do this, we first need to learn a bit more about dynamics of linear systems.

Exercises

- 2.1.** Consider a slightly more general form of the principle of superposition: for the linear system given in state-space form by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u, & \mathbf{x}(0) &= \mathbf{x}_0 \\ y &= \mathbf{C}\mathbf{x} + Du,\end{aligned}$$

show that, if $y_1(t)$ is the solution for an input $u_1(t)$ and initial condition $\mathbf{x}_0 = \mathbf{a}$, and $y_2(t)$ is the solution for an input $u_2(t)$ and initial condition $\mathbf{x}_0 = \mathbf{b}$, then $y_1 + y_2$ is the solution for an input $u_1 + u_2$ and initial condition $\mathbf{x}_0 = \mathbf{a} + \mathbf{b}$.

- 2.2.** The *step response* is the response $H(t)$ of a system to a step input

$$u(t) = \begin{cases} 0, & t < 0; \\ 1, & t \geq 0. \end{cases}$$

Use the convolution integral to show that for a causal system, the step response is the integral of the impulse response $h(t)$:

$$H(t) = \int_0^t h(\tau) d\tau.$$

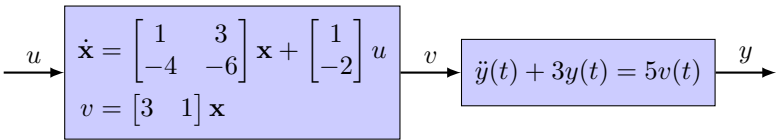
- 2.3.** Prove the convolution property for the Laplace transform, given on page 35.
- 2.4.** Compute the transfer function from the input u to the output y , for the system described by the ODE

$$a\ddot{y} + b\dot{y} + cy = d\dot{u} + eu.$$

- 2.5.** Find the transfer function from the input u to the output y , for the integro-differential equation

$$y(t) = k_p u(t) + k_i \int_0^t u(\tau) d\tau + k_d \dot{u}(t).$$

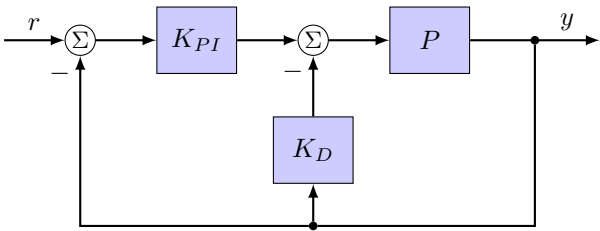
- 2.6. Find the transfer function from the input u to the output y , for the system described by the series interconnection depicted in the following diagram:



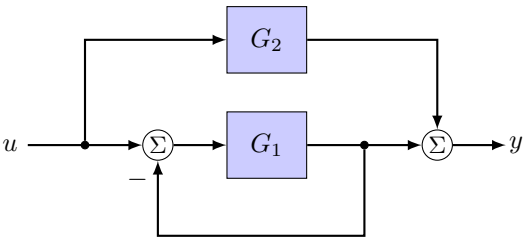
- 2.7. Find an ordinary differential equation with the transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{2s^2 - s + 1}{s^3 + 3s - 2}.$$

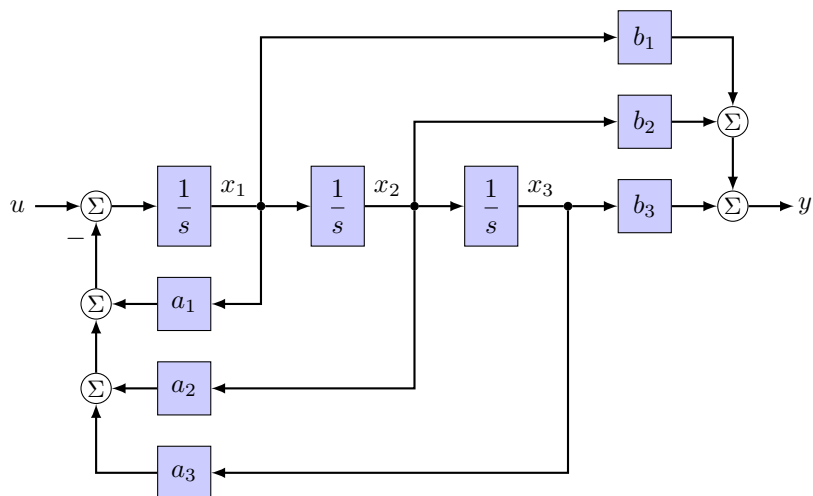
- 2.8. Determine transfer function from the reference input r to the output y for the following block diagram:



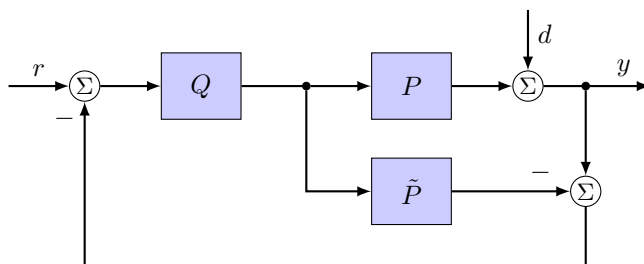
- 2.9. Determine the transfer function from u to y for the following block diagram:



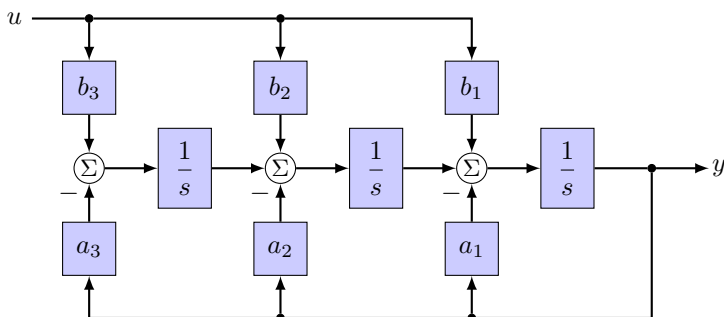
- 2.10. Determine the transfer function from u to y for the following block diagram:



- 2.11.** Determine the transfer function from r to y for the block diagram below. [Hint: Rearrange the diagram, noting that addition is commutative.]



- 2.12.** Determine the transfer function from d to y for the block diagram in Exercise 2.11. [Hint: The answer is not 1.]
- 2.13.** Determine the transfer function from u to y in the block diagram below.



Chapter 3

Linear dynamics

3.1 Poles and zeros

All of the examples of transfer functions we have seen so far have been *rational functions*, or ratios of polynomials. In fact, the transfer function for any system described by differential equations will always be a rational function. One can then factor the numerator and the denominator as follows:

$$H(s) = K \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)}. \quad (3.1)$$

We now have an important definition:

Poles of $H(s)$ are values of s where the denominator of $H(s)$ is zero. *Zeros* of $H(s)$ are values of s where the numerator of $H(s)$ is zero.

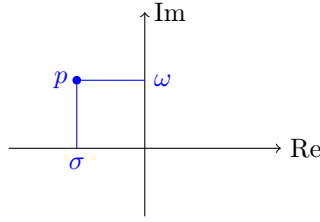
Thus, the zeros of $H(s)$ given by (3.1) are z_1, \dots, z_m , and the poles are p_1, \dots, p_n . The poles tell us about the dynamic response of the system: whether it is stable or unstable, whether its transient response is oscillatory or not, etc. One can in fact show that, as long as the poles p_j are distinct, the impulse response is a sum of exponentials $e^{p_j t}$ where p_j are poles:

$$h(t) = c_1 e^{p_1 t} + c_2 e^{p_2 t} + \cdots + c_n e^{p_n t}. \quad (3.2)$$

These poles may be real or complex. If

$$p = \sigma + i\omega,$$

then we may of course plot p in the complex plane as follows:



The corresponding impulse response is then

$$e^{pt} = e^{(\sigma+i\omega)t} = e^{\sigma t} e^{i\omega t}. \quad (3.3)$$

Since $e^{i\omega t} = \cos \omega t + i \sin \omega t$, this represents a sinusoidal signal, with frequency ω , with an exponential envelope given by $e^{\sigma t}$:

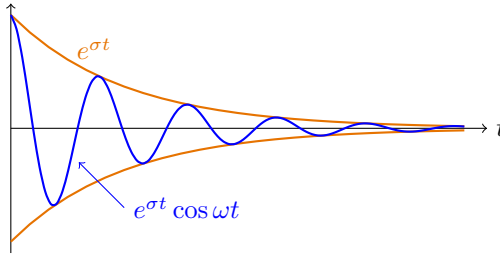


Figure 3.1 shows impulse responses for poles in various locations in the complex plane. Each response is of the form (3.3), for the corresponding value of the pole p . Note that poles in the left half-plane ($\sigma < 0$) have *stable* responses (they decay to zero), while poles in the right half plane ($\sigma > 0$) have *unstable* responses (they blow up). Poles on the imaginary axis are “neutrally stable”: they neither decay to zero, nor blow up¹. Thus, the stability of a system is determined by the pole locations:

A linear system is stable if and only if all of its poles lie in the left half of the complex plane. If even one pole is in the right-half plane, the system is unstable.

The imaginary part ω determines the frequency of oscillation. For poles on the real axis ($\omega = 0$), there is no oscillation, and the larger the imaginary part, the faster the oscillation. Note that the sign of ω does not matter: the impulse response for $p = \sigma - i\omega$ is identical to the plot for $p = \sigma + i\omega$, since in both cases, the real part of (3.3) is $e^{\sigma t} \cos \omega t$.

¹Actually this is true only of *simple poles* (see Appendix B.2). Higher-order poles on the imaginary axis will blow up, though not exponentially.

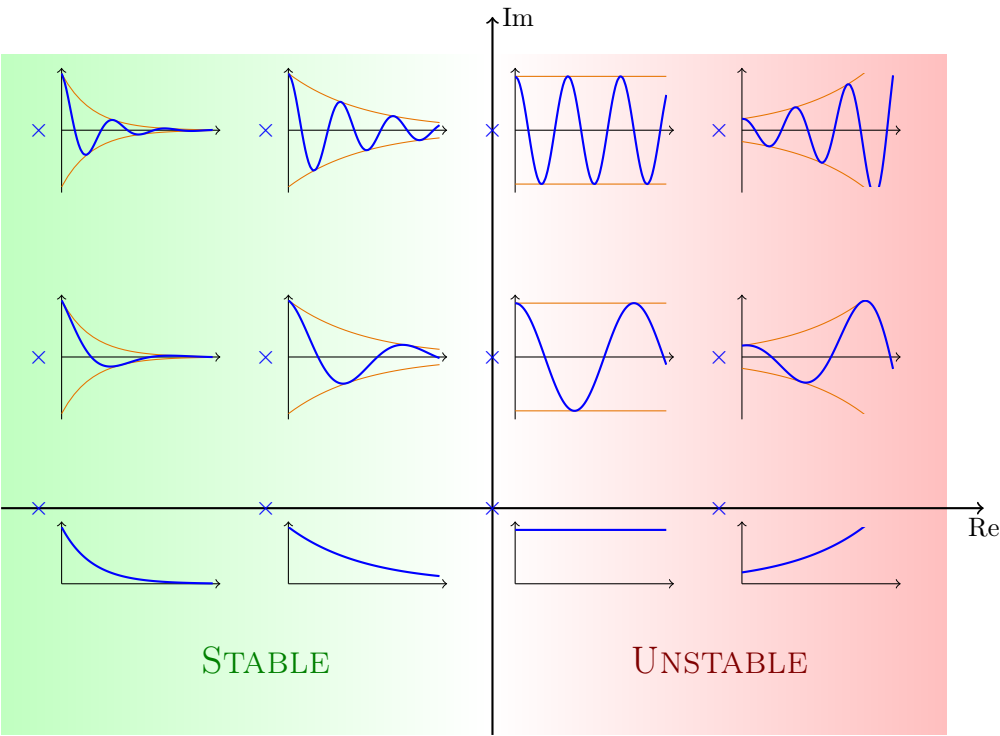


Figure 3.1: Impulse responses for different pole locations in the complex plane. The pole location p is shown as \times , and the corresponding impulse response e^{pt} is shown to the right of it.

Partial fractions

One might wonder how to calculate the constants c_j in (3.2), or even why an expression like this is true in the first place. The reason is that, as long as the p_j are all distinct, the rational function (3.1) may be written as a *partial fraction expansion*

$$H(s) = \frac{c_1}{s - p_1} + \frac{c_2}{s - p_2} + \cdots + \frac{c_n}{s - p_n} + d(s), \quad (3.4)$$

where $d(s)$ is a polynomial, and c_1, \dots, c_n are constants. If the degree of the denominator is larger than the degree of the numerator ($n > m$), then we say the transfer function is *strictly proper*, and in this case $d(s)$ is zero. Then, recalling that

$$\mathcal{L}\{ce^{pt}\} = \frac{c}{s - p},$$

we see that the inverse Laplace transform of $H(s)$ is given by (3.2).

We still have not seen why the partial fraction expansion (3.4) is possible, or how to calculate these constants c_j . We illustrate with an example.

Example 3.1. We calculate the partial fraction expansion for

$$H(s) = \frac{-s + 3}{s^2 + 3s + 2} = \frac{-s + 3}{(s + 1)(s + 2)}.$$

To do this, assume we can write

$$H(s) = \frac{c_1}{s + 1} + \frac{c_2}{s + 2}.$$

To calculate c_1 , multiply both sides of the above equation by $s + 1$, to obtain

$$c_1 + \frac{c_2(s + 1)}{s + 2} = (s + 1)H(s) = \frac{-s + 3}{s + 2}.$$

This must hold for all s , so in particular, we can set $s = -1$ to get

$$c_1 = \lim_{s \rightarrow -1} (s + 1)H(s) = \frac{1 + 3}{-1 + 2} = 4.$$

Similarly, multiplying both sides by $s + 2$ and evaluating at $s = -2$, we have

$$c_2 = \lim_{s \rightarrow -2} (s + 2)H(s) = \frac{2 + 3}{-2 + 1} = -5.$$

Thus,

$$H(s) = \frac{4}{s + 1} - \frac{5}{s + 2}. \quad \diamond$$

In complex analysis, the constant c_j is called the “residue at the pole p_j ,” as explained in Appendix B.2. These residues may be easily calculated in MATLAB with the command

```
[c, p, d] = residue(num, den)
```

Here, **c** is a vector of residues c_j , and **p** is a vector of poles p_j . The polynomial $d(s)$ is given in the vector **d**, in the usual MATLAB format for polynomials.

3.2 Second-order systems

A first-order system has a transfer function with a single pole, such as

$$H(s) = \frac{1}{s + a}.$$

If the coefficients in the transfer function are real (as they must be if the ODEs describing the dynamics are real), then the pole at $s = -a$ is also real, and the impulse response is simply the exponential

$$h(t) = e^{-at}.$$

Simple enough. A second-order system has two poles, such as the following:

$$H(s) = \frac{1}{(s + a)(s + b)}.$$

If a and b are both real, this is not much different from a first-order system: the impulse response is simply a sum of exponentials

$$h(t) = c_1 e^{-at} + c_2 e^{-bt}.$$

But what happens if the poles are complex? This can now happen even if the coefficients in the transfer function are real: since $(s + a)(s + b) = s^2 + (a + b)s + ab$, the condition that the coefficients be real simply implies that a and b be complex-conjugate pairs (that is, $b = \bar{a}$), so the poles can in fact be complex. This leads to *oscillations* in the impulse response, which one cannot have in a (real) first-order system.

Consider as an example the spring-mass system described by

$$m\ddot{y} + b\dot{y} + ky = au.$$

This equation has 4 parameters (m, b, k, a). Let us try to reduce the number of parameters, and put this equation in a standard form. Dividing by m , we have

$$\ddot{y} + \frac{b}{m}\dot{y} + \frac{k}{m}y = \frac{a}{m}u,$$

which now has 3 parameters. If we define new parameters $\omega_0^2 = k/m$, and $2\zeta\omega_0 = b/m$ and rescale the input, writing $\hat{u} = \frac{a}{m}u$, this becomes

$$\ddot{y} + 2\zeta\omega_0\dot{y} + \omega_0^2 y = \omega_0^2 \hat{u}. \quad (3.5)$$

We now have only two parameters, ω_0 and ζ , and we call this the standard second-order system.

The *standard 2nd-order system* has a transfer function

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (3.6)$$

where $\omega_0 > 0$ is called the (undamped) *natural frequency*, and ζ is called the *damping ratio*.

We may actually further reduce the number of parameters by scaling time as well: if we introduce a new, dimensionless time $\tau = \omega_0 t$, then letting $(\dot{})$ denote $d/d\tau$, (3.5) becomes $\ddot{y} + 2\zeta\dot{y} + y = \hat{u}$, and we have reduced the equation to a single parameter ζ . Note that ζ is also dimensionless, so this process is just an example of nondimensionalization. In control system design, however, we are often interested in changing the timescale of the system (this is an example of “modifying the dynamics,” one of the main principles of feedback), so it is often preferable not to rescale time, and to retain the parameter ω_0 .

In any case, the poles of the standard second-order system (3.6) are the roots of the polynomial $s^2 + 2\zeta\omega_0 s + \omega_0^2$, and applying the quadratic formula, we find the poles to be

$$p = -\zeta\omega_0 \pm \omega_0\sqrt{\zeta^2 - 1}. \quad (3.7)$$

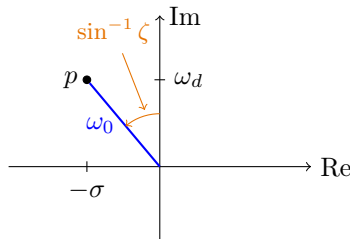
If $|\zeta| < 1$, then the poles are complex, and we write

$$p = -\sigma \pm i\omega_d, \quad \text{where} \quad \begin{aligned} \sigma &= \zeta\omega_0 \\ \omega_d &= \omega_0\sqrt{1 - \zeta^2}. \end{aligned}$$

The parameter ω_d is called the *damped natural frequency* (or simply the *damped frequency*). Since $\sigma^2 + \omega_d^2 = \omega_0^2$, the radius of the pole (the distance from the origin in the complex plane) is simply ω_0 . Also, if θ is the angle the pole makes with the imaginary axis, then

$$\sin \theta = \frac{\sigma}{\omega_0} = \zeta.$$

These relations are shown geometrically in the following diagram:



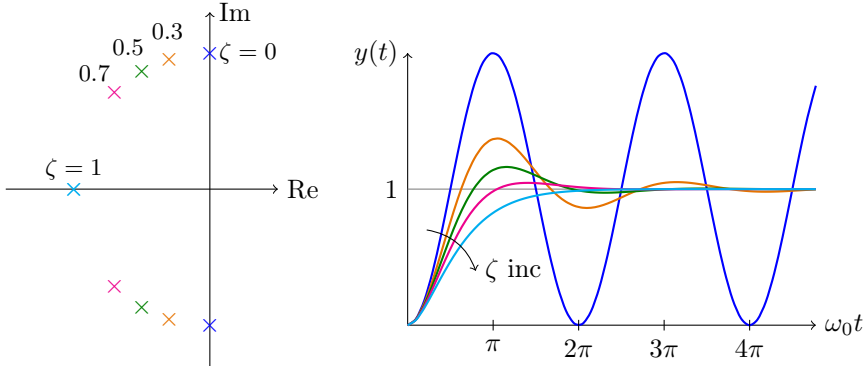


Figure 3.2: Pole locations (left) and step response (right) for a standard second-order system, for damping ratio $\zeta = 0, 0.3, 0.5, 0.7$, and 1.0 .

Let us now consider the response of this standard second-order system to a unit step input $\mathbb{1}(t)$:

$$u(t) = \mathbb{1}(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0. \end{cases}$$

This is called the *step response* of the system. Figure 3.2 shows the step response for several different values of ζ . Note that, for $\zeta = 0$, there is no damping: the step response oscillates forever. As ζ increases, the oscillations become more damped, and the “overshoot” decreases. (We will define overshoot more precisely in the next section, but the intuitive idea should be clear from the diagram.) Normally, we want systems to have a fast response with not much overshoot. If we are designing the dynamics (and we often are, as control engineers), a damping ratio $\zeta = 1/\sqrt{2} \approx 0.707$ is often a good choice, as it is a nice compromise between a fast response and not much overshoot.

Here we have focused on the case where $|\zeta| < 1$, for which the poles are complex. In this case, we say the system is *underdamped*. If $\zeta = 1$, we see from (3.7) that both poles are at $-\omega_0$, and we say the system is *critically damped*. If $\zeta > 1$, then the poles are real, and we say the system is *overdamped*. For an overdamped oscillator, one real pole is less than $-\omega_0$ (the “fast” pole) and the other pole is between $-\omega_0$ and 0 (the “slow” pole).

3.3 Time-domain specifications

Recall from Section 1.1 that one of the main reasons for using feedback is to make a system have some desired dynamics. Often these dynamics are specified as some desired characteristics of the step response. Here, we will define some properties of a typical step response.

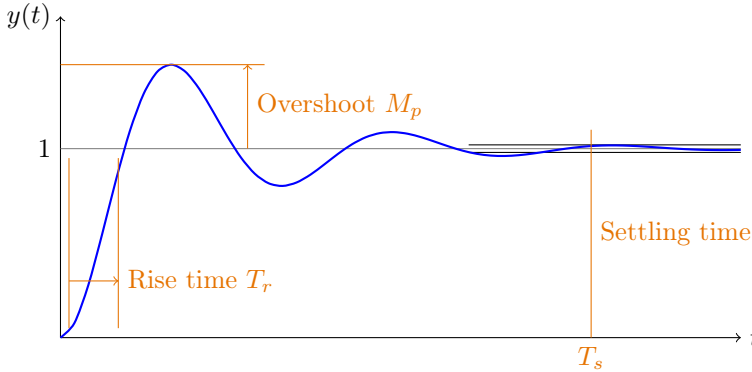


Figure 3.3: Time domain specifications, showing rise time, overshoot, and settling time.

Consider again the standard second-order system (3.6). The step response for this system can be shown to be

$$y(t) = 1 - e^{-\sigma t} \left(\cos \omega_d t + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin \omega_d t \right), \quad (3.8)$$

and is plotted (for $\zeta = 0.25$) in Figure 3.3.

This figure also shows some parameters that are useful when speaking about a step response. We have already mentioned the *overshoot*, denoted M_p in the figure. The overshoot is defined from a system's step response as the fraction of the final value by which the response initially rises above that final value. It is usually expressed as a percentage. If the step response never rises above its final value, the overshoot is zero.

We can actually find an explicit formula for overshoot of a standard second-order system, by differentiating the step response and finding the first time at which the derivative is zero. This maximum occurs at time $t = \pi/\omega_d$, and substituting into the formula for the step response, we find

$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}.$$

Thus, the overshoot depends only on the damping ratio ζ . This function is plotted in Figure 3.4.

Note that, for $\zeta = 0$, there is 100% overshoot, as we already saw in Figure 3.2, and as ζ increases, the overshoot decreases. For $\zeta = 1$, there is zero overshoot, as the poles become real (no oscillation). One feature to note is that, after about $\zeta = 0.7$, the overshoot is practically zero. Since increasing ζ results in a “slower” system, this is one reason why $\zeta = 0.7$ is a good design guideline for a step response: one gets a fast response with very little overshoot.

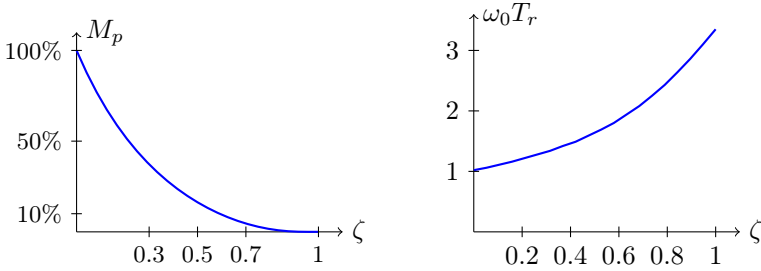


Figure 3.4: Overshoot M_p and normalized rise time $\omega_0 T_r$, as a function of damping ratio ζ .

Another parameter shown in the diagram is the *settling time* T_s , which is the time it takes a system to remain within $\pm 2\%$ of its final value. This choice of 2% is arbitrary, and is sometimes taken to be 1% or 5%, but we will use 2% unless stated otherwise. We can also obtain a formula for the settling time, as

$$T_s = \frac{\log 50}{\sigma} \approx \frac{3.9}{\sigma},$$

so the settling time is determined by σ , the real part of the pole.

The third parameter indicated in Figure 3.3 is the *rise time* T_r , which is defined as the time it takes the step response to go from 10% of its final value to 90% of its final value. (These choices are also arbitrary, but are conventional.) The rise time depends mostly on ω_0 , but also depends a little on ζ , as one can see from Figure 3.2. Figure 3.4 shows the normalized rise time $\omega_0 T_r$ as a function of ζ , computed from simulations of a second-order system. A reasonable approximation of this, obtained by a simple quadratic fit to the data, is given by

$$T_r \approx \frac{1}{\omega_0} (2.2\zeta^2 + 1.1).$$

Franklin et al. [11] use the approximation $T_r \approx 1.8/\omega_0$, which is close for ζ around 0.5–0.7, but relatively poor for other values of ζ . However, these relationships are mainly used for rough design guidelines, and precise results are rarely needed. Some values for T_r are given in Table 3.1.

These relations between time-domain specifications (M_p , T_s , and T_r) and pole locations (e.g., given by ω_0 , ζ , and/or σ) are shown in Figure 3.5. For instance, a time domain specification that the overshoot should be at most 5% implies that the poles should lie in a wedge-shaped region shown in the figure.

One can obtain similar expressions for time-domain specifications for a

Property	Symbol	$\zeta = 0.5$	$\zeta = 0.7$	$\zeta = 1$
Rise time	T_r	$1.6/\omega_0$	$2.2/\omega_0$	$3.3/\omega_0$
Overshoot	M_p	16%	4%	0%
Settling time	T_s	$7.8/\omega_0$	$5.6/\omega_0$	$3.9/\omega_0$

Table 3.1: Relation between time domain specifications and parameters for a standard second-order system.

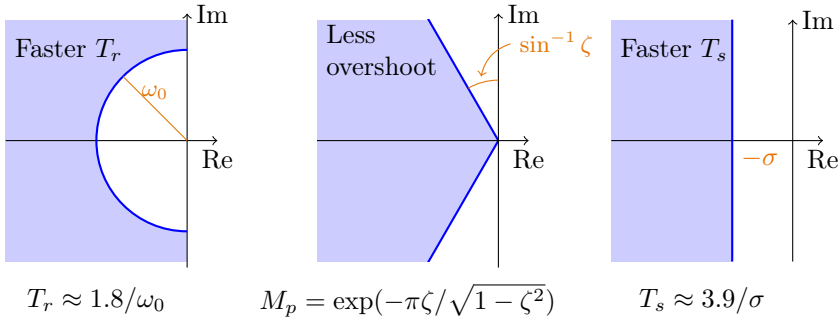


Figure 3.5: Relation between time-domain specifications and pole locations. The rise time is determined (approximately) by the undamped natural frequency ω_0 , the overshoot by the damping ratio ζ , and the settling time by the real part σ .

first-order system. For instance, consider the first-order system

$$H(s) = \frac{1}{s/p + 1},$$

where $p > 0$. This system has a pole at $s = -p$ and a zero-frequency gain of 1, and the step response is given by

$$y(t) = 1 - e^{-pt}, \quad \text{for } t \geq 0.$$

Thus, there is zero overshoot (since $(y(t) < 1$ for all t), so $M_p = 0$ for a first-order system. The rise time and settling time are given by

$$T_s = \frac{\log 50}{p} \approx \frac{3.9}{p}, \quad T_r = \frac{\log 9}{p} \approx \frac{2.2}{p}.$$

3.4 Effect of zeros

In the previous two sections, we focused on the pole locations for the standard second-order system (3.6). This transfer function has no zeros, only poles.

What happens if there are zeros in the transfer function? How do these affect the response?

It turns out, even when there are zeros, poles still have the dominant effect on the dynamics: the response is still the sum of exponentials $e^{p_j t}$, so the poles p_j still determine the stability and other characteristics. However, the zeros do affect the transient response. In general, zeros make the transient response worse, e.g., with more overshoot. Right-half-plane zeros are particularly insidious, and make the system start in the wrong direction. This can make control difficult for various reasons (as we will learn more about in Section 7.4).

To illustrate the effect of zeros, let us compare the two transfer functions below:

$$G_1(s) = \frac{1}{s^2 + 0.5s + 1}, \quad G_2(s) = \frac{s + 1}{s^2 + 0.5s + 1}.$$

Both have the same poles, but G_1 has no zeros, while G_2 has a zero at $s = -1$. Suppose the step response of G_1 is $y_1(t)$. Noting that

$$G_2(s) = G_1(s) + sG_1(s),$$

this implies the step response of G_2 is

$$y_2(t) = y_1(t) + \frac{d}{dt}y_1(t),$$

since multiplication by s in the frequency domain corresponds to differentiation in the time domain. The step responses for y_1 and y_2 are shown in Figure 3.6. Note that y_2 has a larger overshoot than y_1 . This is because, when one adds the derivative \dot{y}_1 to y_1 , the peak will be larger (since the slope is always positive before the peak).

Now let us consider the transfer function

$$G_3(s) = \frac{-s + 1}{s^2 + 0.5s + 1}.$$

This system also has the same poles as G_1 and G_2 , but now we have a right-half-plane zero at $s = 1$. Noting that $G_3 = G_1 - sG_1$, the step response is then

$$y_3(t) = y_1(t) - \frac{d}{dt}y_1(t).$$

This step response is also shown in Figure 3.6. Here, we see a new feature: the step response starts in the wrong direction! The output initially becomes negative, before turning around and eventually reaching the final value of $+1$. This “inverse response” can be quite non-intuitive for a human operator, and this behavior is typical of systems with right-half-plane (RHP) zeros.

Systems with RHP zeros do arise in the real world, however. An example is driving a car in reverse: suppose a car is facing North, and the driver turns the wheel all the way clockwise, as if to parallel park in a space to the right

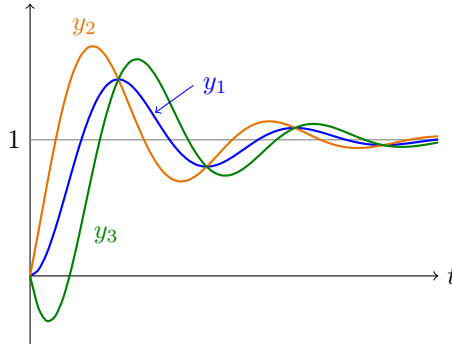


Figure 3.6: Step response for second-order system with no zeros (y_1), a left-half-plane zero (y_2) and a right-half-plane zero (y_3). Note the “inverse response” of the system with the right-half plane zero: it starts in the wrong direction.

(East) of the car. Then as she backs up, the driver initially moves to the West before the car swings around and she eventually moves to the East. The same effect happens with a rear-wheel steering motorcycle or bicycle, and this makes such vehicles notoriously difficult to stabilize. However, an important distinction between zeros and poles is that the zeros of a system depend on the choice of sensors and actuators, while the poles do not. For instance, in the example of driving the car in reverse, if the driver were to sit at the center of the rear axle of the car, then she would not move in the wrong direction first; the corresponding transfer function (of the appropriately linearized system) no longer has a RHP zero. We will return to these examples, and the effects of RHP zeros more generally, in Section 7.4.

Exercises

3.1. Show that the roots of

$$f(s) = s^2 + as + b$$

are in the left half-plane (negative real parts) if and only if $a > 0$ and $b > 0$.

3.2. For the block diagram shown in Figure 3.7, find the following (closed-loop) transfer functions:

- (a) from the reference signal r to y ;
- (b) from the reference signal r to e ;
- (c) from the sensor noise n to u .

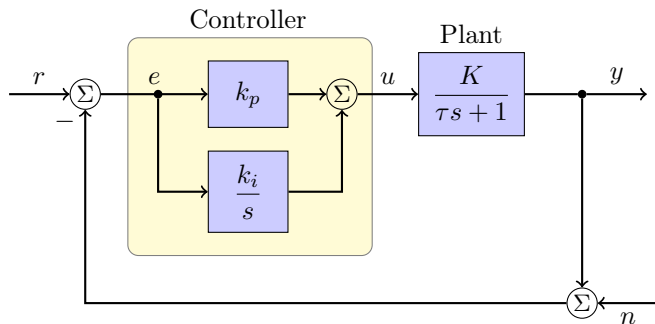


Figure 3.7: Block diagram for a first-order plant with proportional-integral (PI) control. The input r is a reference signal we wish the output y to track, and n represents sensor noise.

- 3.3. For the controller shown in Figure 3.7, find conditions on k_p and k_i that are necessary and sufficient for the closed-loop system to be stable, for $K, \tau > 0$. [Hint: Use problem 3.1.]
- 3.4. For the control system in Figure 3.7, determine the steady-state error to a step reference (i.e., for r equal to the unit step function).
- 3.5. For the control system in Figure 3.7, prove that no matter what the values of K and τ are, the controller provides enough flexibility to place the poles of the closed-loop system anywhere in the complex plane.
- 3.6. For the system in Figure 3.7, suppose you desire the closed-loop system to have the following properties:
 - i. Settling time $T_s \leq 0.5$ sec
 - ii. Overshoot less than 20%.
 - (a) Determine and sketch the region in the complex plane where the closed-loop poles must lie in order to satisfy these specifications. (Neglect the effects of any closed-loop zeros.)
 - (b) For the standard second-order system (3.6), what values of ω_0 and ζ correspond to this region of the complex plane?
 - (c) Let $K = 2$, $\tau = 2$. Find values for k_p and k_i so that the poles of the closed-loop system lie within the desired region.
 - (d) Using Matlab (e.g., the **step** command from the Control System Toolbox), simulate the closed-loop response to a step reference r , with $n = 0$. Does your response in fact satisfy the given specifications? If not, explain why not, and adjust the values of k_p and k_i so that they do.

- 3.7.** Consider the effects of sensor noise n for the system shown in Figure 3.7. Suppose there is high-frequency noise with a magnitude of 15 mV, and we want the noise at our actuator u to be no more than 0.6 V in magnitude at high frequencies (frequencies much larger than the natural dynamics of the system). What restrictions does this place on the gains k_p and k_i ?
- 3.8.** Assuming $K = 1$ and $\tau = 0.5$, design a PI controller so that the steady-state error to a step input is less than 1%, and the noise specification from Exercise 3.7 is also satisfied. Can this be achieved with a proportional controller alone ($k_i = 0$)? If so, give an example; if not, why not?

Chapter 4

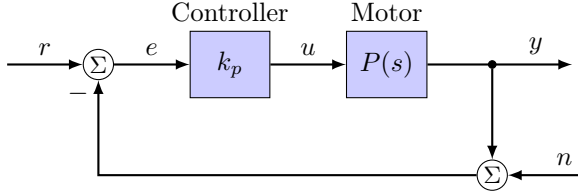
Classical feedback design

In this chapter, we present some methods for designing feedback controllers in the frequency domain. We will learn techniques such as root-locus plots, Bode plots, and Nyquist plots, and the Nyquist stability criterion. These methods were developed in the 1930's and 40's, before the invention of digital computers, and are mostly graphical in nature, enabling one to understand the essential characteristics of feedback loops without doing extensive calculations. The methods form the meat and potatoes of “classical control theory,” as described in texts such as [3, 16].

We have seen several examples where simple proportional control can achieve some impressive results. However, more often than not, proportional control is not sufficient for real control problems, even for a simple first-order plant. In the next section, we will revisit the example of controlling the speed of a DC motor, and show some of the limitations of simple proportional feedback. This will provide some motivation for studying more sophisticated controllers, such as PID (proportional-integral-derivative) controllers.

4.1 Feedback changes dynamics

Let us revisit the problem of controlling the speed of a DC motor, now that we have a better understanding of how to more accurately model the dynamics. (Recall that in Section 1.1 we modeled the dynamics as a pure time delay, but this did not give reasonable results.) We will also include some small sensor noise. Let us begin using simple proportional control, as shown in the diagram below:



The transfer function for our DC motor is given by

$$P(s) = \frac{k}{\tau s + 1},$$

which has a pole at $s = -1/\tau$. We call this an *open-loop pole*, because it is a pole of the system before the feedback loop has been closed. The block diagram represents a feedback law given by

$$u = k_p(r - y - n),$$

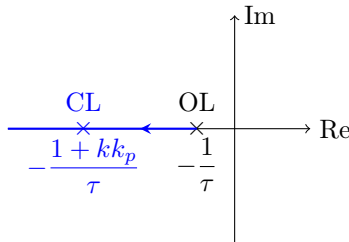
which is a feedback with a proportional gain given by k_p . The resulting transfer function from r to e is then

$$\begin{aligned} G_{er}(s) &= \frac{1}{1 + Pk_p} = \frac{1}{1 + kk_p/(\tau s + 1)} \\ &= \frac{\tau s + 1}{\tau s + 1 + kk_p}. \end{aligned}$$

This is a *closed-loop* transfer function, so there is a *closed-loop pole* where the denominator is zero, or at

$$s = -\frac{1 + kk_p}{\tau}.$$

When $k_p = 0$, the closed-loop pole is $-1/\tau$, the same as the open-loop pole; and as k_p increases, the pole becomes more negative, as shown in the following diagram:



For $k_p > 0$, the response gets *faster* (since the pole moves further in the left half plane). This is a good thing! So far, it seems that proportional feedback is doing just fine.

Next, let us look at the steady-state error to a step reference. Assuming the closed-loop pole is in the left half-plane, we know from Section 2.4 that this steady-state error is given by

$$G_{er}(0) = \frac{1}{1 + k k_p}.$$

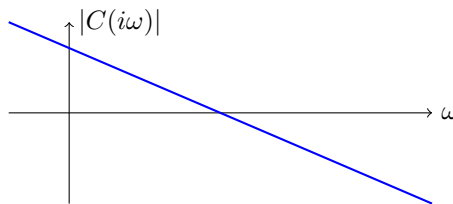
This value is never zero, but for large k_p , it approaches zero.

It appears that as we make k_p larger, everything gets better: the response gets faster, and the steady-state error gets smaller. Why not make k_p as large as possible? The answer, in this case, is because of sensor noise. Consider the transfer function from n to u :

$$G_{un} = \frac{-k_p}{1 + P k_p}.$$

As $\omega \rightarrow \infty$, $|P(i\omega)| \rightarrow 0$, so $G_{un}(i\omega) \rightarrow -k_p$. If k_p is very large, we will have severe amplification of sensor noise. We mentioned this previously, in Section 2.6, but now we see a concrete example.

We would like to do better. It seems that the ideal situation would be to have a controller with large gain at low frequencies (for good reference tracking), and small gain at high frequencies (to avoid amplifying sensor noise). Thus, instead of a constant proportional gain k_p , we would like a controller $C(s)$ whose gain *depends on the frequency*, with a frequency response something like this:



It turns out that this is exactly what the frequency response of an *integrator* looks like, $C(s) = k_i/s$. This is one of the motivations behind using integral feedback.

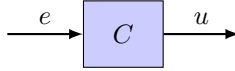
There are two main points to take away from this example:

- Feedback changes the pole locations.
- Sensor noise can cause major problems if high gains are used.

However, at this point, it is not clear what a controller such as $C(s)$ above would do to the dynamics of the overall feedback loop. We saw that with proportional control, the closed-loop system is still stable, but what about with integral control? Understanding how the dynamics change when feedback is introduced is one of the main goals of this chapter.

PID control

We now consider a very important class of controllers, called *Proportional-Integral-Derivative (PID)* controllers. The vast majority of controllers used in practice are PID controllers. We regard a controller as an input-output system where the input is typically the error e , and the output is the signal u that is sent to the plant:



A PID controller has the following form:

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

That is, it includes terms that are proportional to the error, the integral of the error, and the derivative of the error. The motivation for including the proportional term should be obvious by now, and the motivation for the integral term also makes sense, in light of the previous discussion. The motivation for including the derivative term is not yet clear, but don't worry: it will become clear shortly.

Taking the Laplace transform of the above expression for $u(t)$ gives the transfer function for a PID controller, as

$$C(s) = k_p + \frac{k_i}{s} + k_d s.$$

In a PID controller, there are therefore three parameters to choose: k_p , k_i , and k_d , the proportional, integral, and derivative gains. Why do we need three different knobs, and what do the different knobs do, intuitively? The answer depends very much on the plant $P(s)$ to be controlled. However, as we will see shortly, for a second-order system such as a spring-mass-damper, there is a convenient interpretation:

- Increasing k_p acts like a stiffer spring
- Increasing k_d acts like additional damping
- Increasing k_i improves steady-state tracking, but hurts transients.

It is sometimes convenient to write a PID controller as

$$C(s) = k_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right),$$

where τ_i and τ_d are the time constants of the integral and derivative terms. (Note that, since s has units of 1/time, the terms in parentheses are dimensionless.)

Feedback changes pole locations

We have seen several examples of how feedback changes the dynamics of a system, but now let us look at this effect more generally. Consider a plant with a transfer function

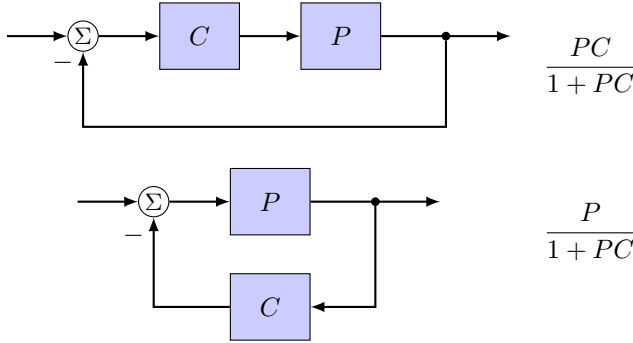
$$P(s) = \frac{n(s)}{d(s)}$$

where $n(s)$ and $d(s)$ are polynomials. The plant has poles where

$$d(s) = 0.$$

This is called the *characteristic equation* for the open-loop system, and its solutions are the open-loop poles.

Now, consider the two feedback loops shown in the diagrams below, with their corresponding closed-loop transfer functions:



In either case, the denominator of the closed-loop transfer function is $1 + PC$, so the closed-loop poles are where

$$1 + P(s)C(s) = 0.$$

If

$$C(s) = \frac{n_c(s)}{d_c(s)},$$

where $n_c(s)$ and $d_c(s)$ are polynomials, then

$$\begin{aligned} 1 + PC &= 0 \\ \implies 1 + \frac{n}{d} \frac{n_c}{d_c} &= 0 \\ \implies d(s)d_c(s) + n(s)n_c(s) &= 0. \end{aligned}$$

This is the closed-loop characteristic equation. These roots are nothing like the open-loop poles, which are roots of $d(s)$. (However note that with no feedback, for $C(s) = 0$, we have $n_c = 0$, $d_c = 1$, so we recover the open-loop

characteristic equation, $d(s) = 0$.) This ability of feedback to completely change the pole locations is both a blessing and a curse. We can do useful things like stabilize an unstable system, but we can also do bad things like destabilize a perfectly well-behaved system.

PD control of a spring-mass

Consider a proportional-derivative (PD) controller for the spring-mass system from Example 1.2. The plant transfer function is

$$P(s) = \frac{1}{ms^2 + bs + k}, \quad (4.1)$$

and the transfer function of a PD controller is

$$C(s) = k_p + k_d s,$$

where k_p is the *proportional gain* and k_d is the *derivative gain*. The closed-loop poles satisfy

$$\begin{aligned} 1 + PC &= 0 \\ \implies 1 + \frac{k_p + k_d s}{ms^2 + bs + k} &= 0. \end{aligned}$$

Multiplying by the denominator then gives the (closed-loop) characteristic equation

$$ms^2 + (b + k_d)s + (k + k_p) = 0.$$

On the other hand, the *open-loop* poles (of $P(s)$) satisfy the characteristic equation

$$ms^2 + bs + k = 0. \quad (4.2)$$

Comparing these two, we see the following:

For the second-order system (4.1), a derivative gain $k_d > 0$ acts like increasing b , the damping, while a proportional gain $k_p > 0$ acts like increasing k , the spring constant.

Proportional feedback

First, consider the case where $k_d = 0$ (proportional feedback), and let us look at how feedback affects the location of the poles in the complex plane. The closed-loop characteristic equation is now

$$ms^2 + bs + (k + k_p) = 0,$$

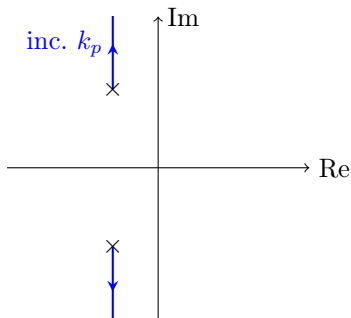
which has solutions

$$s = \frac{-b \pm \sqrt{b^2 - 4m(k + k_p)}}{2m}.$$

Let us assume that $b^2 < 4km$, so that the system is underdamped, and the poles are complex (see Section 3.2). Note that as k_p increases, the real part of the poles ($-b/2m$) remains unchanged, but the imaginary part increases. We could also see this by comparing with the characteristic equation for the standard second-order system

$$s^2 + 2\zeta\omega_0 s + \omega_0^2 = 0. \quad (4.3)$$

Here, $\sigma = \zeta\omega_0 = b/(2m)$ is independent of the gain k_p , so the real part of the poles ($-\sigma$) remains unchanged, while the natural frequency $\omega_0 = \sqrt{(k + k_p)/m}$ does depend on k_p , so the imaginary part increases with k_p . We can plot these pole locations in the complex plane as follows:



where \times indicates the location of the open-loop poles (solutions of (4.2)). Recalling our discussion of second-order systems from Section 3.2, we see that as k_p increases, the system will have the same exponential envelope (given by $-\sigma$, the real part of the pole), but a higher frequency of oscillation (given by ω_d , the imaginary part), and a higher overshoot (determined by $\zeta = \sin \theta$, where θ is the angle the pole makes with the imaginary axis).

Derivative feedback

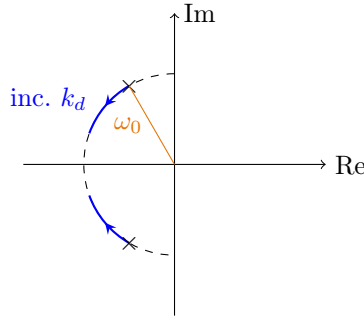
Next, consider the case where $k_p = 0$ but $k_d \neq 0$ (derivative feedback). Now, the closed-loop characteristic equation is

$$ms^2 + (b + k_d)s + k = 0.$$

Comparing with the characteristic equation for a standard second-order system (4.3), we see that

$$2\zeta\omega_0 = \frac{b + k_d}{m}, \quad \omega_0^2 = \frac{k}{m}.$$

Thus, increasing k_d changes the damping ratio ζ , while keeping ω_0 constant. The corresponding location of the poles in the complex plane is shown below:



Integral control of a spring-mass

We now have a fairly good understanding of how proportional-derivative (PD) feedback affects a second-order system, such as a spring-mass. Now, let us look at the effect of integral feedback

$$C(s) = \frac{k_i}{s}.$$

The closed-loop characteristic equation $1 + PC = 0$ now becomes

$$\begin{aligned} 1 + \frac{1}{ms^2 + bs + k} \frac{k_i}{s} &= 0 \\ \implies ms^3 + bs^2 + ks + k_i &= 0. \end{aligned}$$

Now the characteristic equation is *cubic*. We now have three poles instead of two, and we cannot use the same intuition we could for PD feedback, for which the equation remained quadratic. We cannot use the quadratic formula, nor can we use our results for the standard second-order system (since the equation is no longer second order). We could in principle use the formula for roots of a cubic polynomial to find the location of the 3 poles, but this would get messy, and furthermore would not scale well to higher-order systems. We need a better method if we wish to study the effect of integral feedback. This is part of the motivation behind the root-locus method, discussed in the next section.

4.2 Root-locus diagrams

The root-locus method is a technique invented by Walter Evans in 1948 [10] for plotting the poles of a system as a parameter is varied. The beauty of the method is that one may sketch these locations quickly, with little or no

calculation, even for systems with a fairly large number of poles. Before describing the details of the method, we first illustrate with some examples.

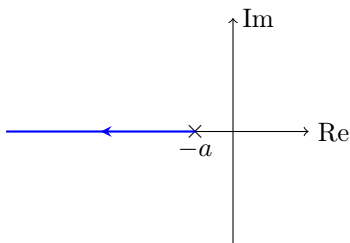
Consider a first-order plant with transfer function

$$P(s) = \frac{1}{s + a},$$

which has a pole at $s = -a$. Now, consider a proportional feedback $C(s) = K$. As in the previous section, we know the closed-loop characteristic equation is $1 + PC = 0$, which becomes

$$1 + \frac{K}{s + a} = 0. \quad (4.4a)$$

Solving, we see the closed-loop pole is at $s = -a - K$. We can sketch how the pole moves as the proportional gain K is increased from 0 to $+\infty$ as follows:



This sketch is called a *root locus diagram*: it shows how the location (locus) of the pole changes as a parameter K increases from 0 to $+\infty$.

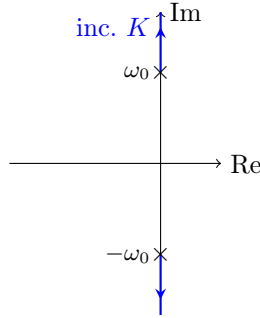
Now, consider a second-order system

$$P(s) = \frac{1}{s^2 + \omega_0^2}.$$

This system has no damping, with open-loop poles at $s = \pm i\omega_0$. Again using a proportional feedback, $C(s) = K$, the closed-loop characteristic equation is

$$\begin{aligned} 1 + \frac{K}{s^2 + \omega_0^2} &= 0 \\ \implies s^2 + \omega_0^2 + K &= 0, \end{aligned} \quad (4.4b)$$

which has solutions $s = \pm i\sqrt{\omega_0^2 + K}$. Sketching these locations as K increases from 0 to $+\infty$, we obtain the following root locus diagram:



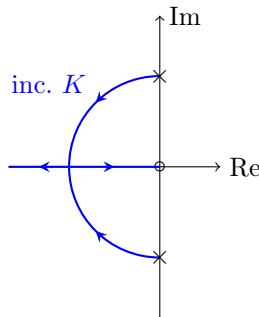
Now there are two “branches,” one starting at each open-loop pole.

Next, let us consider the same second-order plant $P(s)$, but with a derivative feedback, $C(s) = Ks$. Now the closed-loop characteristic equation is

$$1 + \frac{Ks}{s^2 + \omega_0^2} = 0 \quad (4.4c)$$

$$\implies s^2 + Ks + \omega_0^2 = 0,$$

which corresponds to a standard second-order system with $2\zeta\omega_0 = K$. If we plot these roots as K varies from 0 to $+\infty$ we obtain the following root locus diagram:



The two roots start at the open-loop poles for $K = 0$, and meet at the real axis when $\zeta = 1$. They then split apart, and one real pole goes to the origin as $K \rightarrow \infty$, while the other pole marches off to $-\infty$.

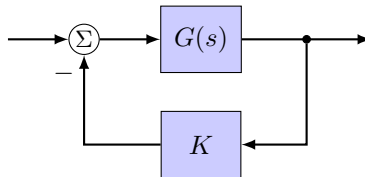
Note that each of the characteristic equations (4.4) has the form $1 + KG(s) = 0$ for some rational function $G(s)$. In all of these examples, $G(s)$ was the loop gain $P(s)C(s)$, but it does not need to be. This motivates the following definition:

The root-locus plot of $G(s)$ is a plot of all solutions to

$$1 + KG(s) = 0 \quad (4.5)$$

as K varies from $0 \rightarrow +\infty$.

The form of equation (4.5) is sometimes referred to as the *root-locus form*, where K is some parameter we wish to vary. (Note that K does not need to be the overall gain, though it often is.) The root locus plot therefore shows the location of the closed-loop poles of the following feedback system, as K varies:



However, bear in mind that this diagram is only one interpretation, and the method also describes closed-loop pole locations for either of the feedback loops shown on page 71, or indeed any system whose poles satisfy (4.5).

The method developed by Evans [10] is a graphical technique to plot all such points using a few simple rules. How could we possibly find an easy method for determining solutions to polynomial equations of arbitrary order? This sounds like a very difficult problem. The key is to notice the following: since (4.5) is equivalent to

$$G(s) = -\frac{1}{K}, \quad \text{for some } K > 0,$$

this implies that, if s lies on the root locus, then $G(s)$ must be a *negative real* number, or equivalently, that

$$\angle G(s) = 180^\circ. \quad (4.6)$$

Most of the rules for graphically determining the root locus rely only on this property.

Rules for sketching root-locus diagrams

We now give some basic rules for sketching root-locus diagrams. Our goal here is not to be able to sketch highly accurate root-locus plots by hand (MATLAB is very good at this: see the command `rlocus`), but rather to gain some intuition for how feedback changes pole locations. It turns out that, once we know some basic rules, some important and nontrivial facts about feedback systems will become obvious.

Suppose $G(s)$ in (4.5) has the form

$$G(s) = k \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)}. \quad (4.7)$$

We assume that $G(s)$ is a *proper* transfer function, which means simply that $n \geq m$, or that the degree of the numerator does not exceed the degree of the denominator. (Similarly, a *strictly proper* transfer function is one in which $n > m$.)

Since $G(s)$ has n poles, we see that (4.5) will also have n solutions, for any value of K . So the root locus plot consists of n *branches* of solutions in the complex plane. These branches are precisely those shown in the examples above.

Rule 1: Start and end. *The n branches of the root locus start at the open-loop poles, and end either at the open-loop zeros or at infinity.*

It is straightforward to see why this rule holds. If we write

$$G(s) = \frac{n(s)}{d(s)},$$

then equation (4.5) becomes

$$d(s) + Kn(s) = 0.$$

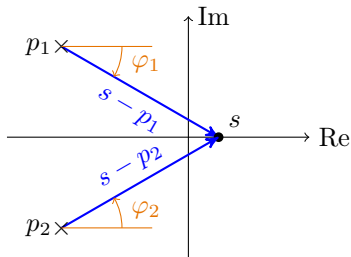
For $K = 0$, this then becomes $d(s) = 0$, and solutions of this are the open-loop poles (the poles of $G(s)$). Next, consider what happens as $K \rightarrow \infty$. If $d(s)$ remains finite (i.e., if s is finite), then we must have $n(s) = 0$. Hence, as $K \rightarrow \infty$, s must either go to a zero of $G(s)$, or s must go to infinity.

Rule 2: Real axis. *A point on the real axis lies on the root locus if the number of zeros + poles to the right of it is odd.*

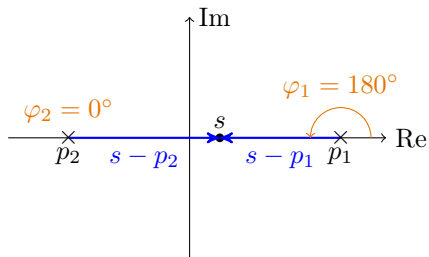
To see why this rule holds, note that, from (4.7), we have

$$\angle G(s) = \sum_{k=1}^m \angle(s - z_k) - \sum_{k=1}^n \angle(s - p_k). \quad (4.8)$$

Let us look at the contributions of terms like $(s - p_k)$, where s is on the real axis. First, note that a complex-conjugate pair of poles (or zeros) contributes nothing to this overall sum, as the phases cancel out, as shown in the following diagram:



Since $\varphi_1 = -\varphi_2$, there is no net contribution to the sum in (4.8). The same holds true for complex-conjugate zeros. Thus, the only terms in (4.8) that will contribute to the phase are the real poles and zeros. Now, consider the phase contribution of real poles:



From the diagram, we see that, if p_1 is a real pole that lies to the right of s , then $\varphi_1 = \angle(s - p_1) = 180^\circ$, while if p_2 lies to the left of s , then $\varphi_2 = \angle(s - p_2) = 0^\circ$. The same is true for zeros z_k , so the poles/zeros to the right of s are the only ones that contribute to the sum (4.8). Furthermore, since each pole/zero contributes a phase of 180° , we see that we need an odd number of them in order for the phase condition (4.6) to be satisfied.

Rule 3: Asymptotes. As $K \rightarrow \infty$, the branches tend to straight-line asymptotes radiating from a common point, or centroid, on the real axis. The location of the centroid is given by

$$\alpha = \frac{\sum_{k=1}^n p_k - \sum_{k=1}^m z_k}{r}, \quad (4.9a)$$

and the angles of the asymptotes are

$$\theta_k = \frac{\pi + 2\pi k}{r}, \quad k = 0, 1, \dots, r-1, \quad (4.9b)$$

where as before, r is the relative degree of $G(s)$.

This property is also straightforward to show. The idea is that, if $|s|$ is very large (compared to any of the poles/zeros of G), then $G(s)$ can be approximated by

$$G(s) \approx \frac{1}{(s - \alpha)^r}.$$

(To see this, imagine “zooming out” of the plot very far: m of the poles will approximately “cancel” with the m zeros, leaving $r = n - m$ remaining poles that appear to be at some common point α .) Letting $z = s - \alpha$, equation (4.5) then becomes $z^r = -K$, which has solutions

$$\begin{aligned} z &= (-K)^{1/r} = K^{1/r}(-1)^{1/r} = K^{1/r}(e^{i(\pi+2\pi k)})^{1/r} \\ &= K^{1/r}e^{i(\pi+2\pi k)/r}, \quad k = 0, \dots, r-1. \end{aligned}$$

This then describes the asymptotes specified in the formulas above. In fact, we even see that the r poles go off to infinity at the rate $K^{1/r}$ as $K \rightarrow \infty$, and all r poles maintain approximately the same distance from the centroid α . (That is, one pole does not shoot off to infinity faster than the others.) Of course, all of the above is valid only when $|s|$ is large, and therefore when K is large, so these are *asymptotes*, not exact solutions for the branches.

An intuitive way to think about the centroid α is as follows. First, note that if there are no zeros, then (4.9a) is just the formula for the center of mass of a system with point masses located at the open-loop poles p_i . (It is simply the mean of these complex numbers.) Now, if there are zeros, then (4.9a) can be thought of as the center of mass of a system for which there are masses at the open-loop pole locations p_i and balloons at the zero locations z_i , with “negative mass” equal to that of the positive mass at the pole locations.

Figure 4.1 shows the arrangement of the asymptotes for $r = 1, \dots, 4$. In practice, the formulas (4.9) are rarely required, as one can often determine the centroid by inspection, and the most common cases shown in Figure 4.1 are easy to remember.

Let us now summarize our rules.

Rules for sketching a root locus plot of $G(s)$:

1. *Start and end.* Branches start at open-loop poles, and end at open-loop zeros or at infinity.
2. *Real axis.* A point on the real axis lies on the root locus if the number of zeros + poles to the right of it is odd.
3. *Asymptotes.* As $K \rightarrow \infty$, the branches tend to straight-line asymptotes radiating from a common point, or centroid, on the real axis. The asymptotes are arranged as in Figure 4.1.

There are more rules for calculating other properties of root locus diagrams that we do not discuss here. For instance, [11] gives six rules altogether, including rules for computing departure angles, crossings of the imaginary axis, and points with multiple roots, in addition to our three rules above. However, these additional rules require some calculation, while the

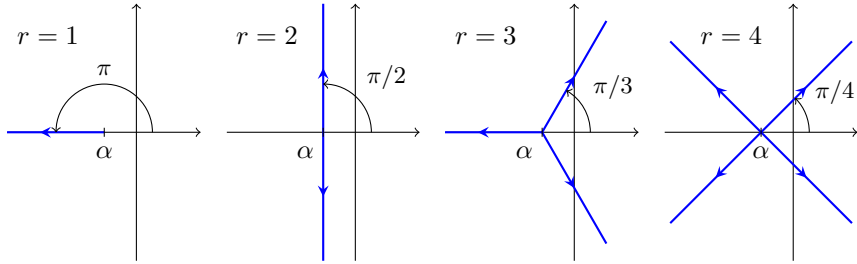


Figure 4.1: Asymptotes for systems with relative degree $r = 1, 2, 3, 4$, as given by (4.9b). The centroid α is given by (4.9a).

three rules above are simpler and do not require any calculation (except perhaps for the centroid of the asymptotes). In practice, if such details are important, it is generally easier to use software packages such as MATLAB, rather than spend time on meticulous hand sketches.

Our three rules give only the basic information about a root locus diagram, but they reveal some nontrivial facts about the behavior of feedback systems. For instance, we can deduce the following about the behavior of closed-loop dynamics:

- Any system whose open-loop transfer function has relative degree $r \geq 3$ will eventually go unstable for large enough proportional gain.
- Any system with an open-loop zero in the right half-plane will eventually go unstable for large enough gain.
- We would expect a system with a right-half-plane zero close to a right-half-plane pole to be very difficult to stabilize.

Here, when we say the system will “go unstable,” what we mean is that there will be a closed-loop pole in the right half-plane.

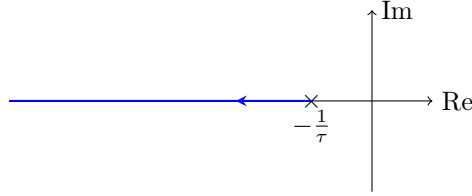
Examples of root locus diagrams

We now give some examples of root locus diagrams, showing how to apply the rules we have just learned.

Example 4.1. Consider our familiar example of a DC motor with proportional feedback:

$$P(s) = \frac{A}{\tau s + 1}, \quad C(s) = K.$$

We have already seen the root locus diagram for this system, but let us apply our rules, and verify that we get the same picture. The first step is to find the open-loop poles and zeros: here, we have a single pole at $s = -1/\tau$. We then plot this pole in the complex plane:



Applying the real-axis rule, we see that points on the real axis to the left of this pole are on the root locus, so we color those points blue. This actually completes the diagram: we see the closed-loop pole at $-1/\tau$ moves to the left as K increases. \diamond

Example 4.2 (Motor with PI feedback). Next, consider the same plant, but with proportional-integral (PI) feedback:

$$P(s) = \frac{A}{\tau s + 1} \quad C(s) = K \left(1 + \frac{1}{\tau_i s} \right).$$

We will assume τ_i is fixed, and we will vary the overall gain K . The loop gain PC is then

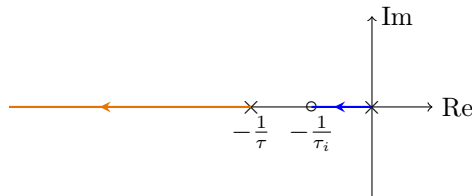
$$PC = \frac{KA(\tau_i s + 1)}{\tau_i s(\tau s + 1)},$$

so for the root locus method we take

$$G(s) = \frac{A(\tau_i s + 1)}{\tau_i s(\tau s + 1)}$$

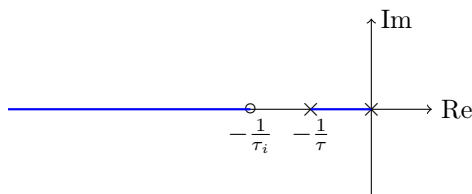
so the closed-loop characteristic equation $1 + PC = 0$ has the form $1 + KG(s) = 0$. The first step is to find the poles and zeros of $G(s)$: there are now two poles, one at $s = 0$ and one at $s = -1/\tau$, and there is now one zero, at $s = -1/\tau_i$.

Case 1: First, assume $\tau_i > \tau$. We plot the open-loop poles and zeros in the complex plane, marking poles by \times and the zero by \circ :

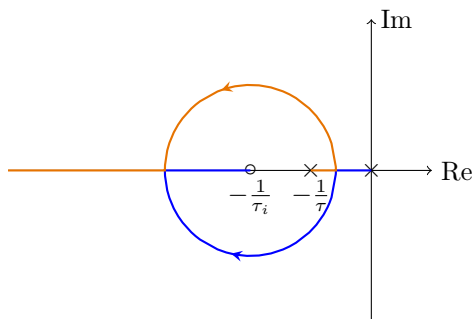


Next, applying the real-axis rule, we color the corresponding portions of the real axis that must be on the root locus. Applying Rule 1, we know that the branches start at the open-loop poles and end either at the open-loop zeros or at infinity, so we can draw the arrows as shown on the plot, showing how the poles move as K increases. Note that there is a single asymptote as $K \rightarrow \infty$, since the relative degree is $r = 2 - 1 = 1$.

Case 2: Now, assume $\tau_i < \tau$ (which is actually how one should design such a controller in practice). The open-loop poles and open-loop zero are the same as above, but now have a different arrangement:



The real-axis portions are found as before, and are also colored in the diagram above. However, now it is not so obvious how the poles move as K increases. Let us think about Rule 1: the branches must start at the open-loop poles, so in the blue segment between the two poles on the real axis, the poles must now start moving *towards* each other. We know that one of these poles must somehow end up at the open-loop zero, and the other must go off to $-\infty$, but how could they get there? It is impossible to know the precise details from our rules, but we do have a clue, since we know the real-axis portion of the plot. It turns out that once the poles meet, they split into a complex-conjugate pair, and then meet again on the real axis to the left of the zero:

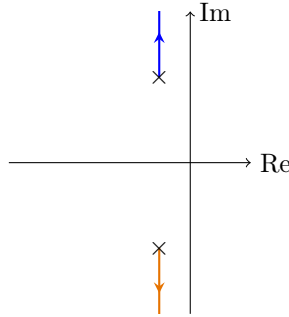


There are more advanced rules for determining when two poles meet (and therefore determining the “break-out” point, where they leave the real axis, and the “break-in” point when they return to the real axis), but we will not worry about calculating such details by hand. It turns out that, whenever two poles meet like this, they will leave/approach the real axis at angles of $\pm 90^\circ$. (This is not one of our official rules, but it is handy to know.) Even though our rules do not tell us all of the details, they do give us enough information to deduce this general shape, and we can easily get the details by plotting this in MATLAB. \diamond

Example 4.3. Consider a second-order system

$$P(s) = \frac{1}{s^2 + 2\zeta\omega_0 s + \omega_0^2},$$

with a proportional feedback $C(s) = k_p$. Let us look at how the closed-loop poles vary with $k_p \geq 0$. The closed-loop poles satisfy $1 + k_p P(s) = 0$, so in the root-locus method, we therefore take $G(s) = P(s)$. There are two open-loop poles, a complex pair with radius ω_0 , that make an angle $\theta = \sin^{-1} \zeta$ with the imaginary axis. These are plotted below:

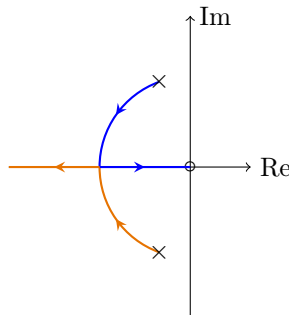


There are no open-loop poles or zeros on the real axis, so no part of the root locus lies on the real axis either. The relative degree is $r = 2$ (2 poles – 0 zeros), so there are two asymptotes, as shown in Figure 4.1, and the centroid is simply the midpoint between the two poles. Here, the closed-loop poles lie directly on the asymptotes, and we see the poles simply move vertically as the gain k_p increases; the real part remains constant. Note that our rules do not tell us exactly how the closed-loop poles move towards the asymptotes, but it is certainly a reasonable guess to assume that they simply move vertically; this is, of course, consistent with our previous results for proportional feedback of a second-order system, on page 72. \diamond

Example 4.4. Consider the same system as in the previous example, but now with derivative feedback $C(s) = k_d s$. We are now interested in how the closed-loop poles vary with $k_d \geq 0$. The closed-loop poles are now roots of $1 + k_d G(s)$, with

$$G(s) = \frac{s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}.$$

The open-loop poles are the same as before, a complex-conjugate pair, but now there is an open-loop zero at $s = 0$. These are plotted below:

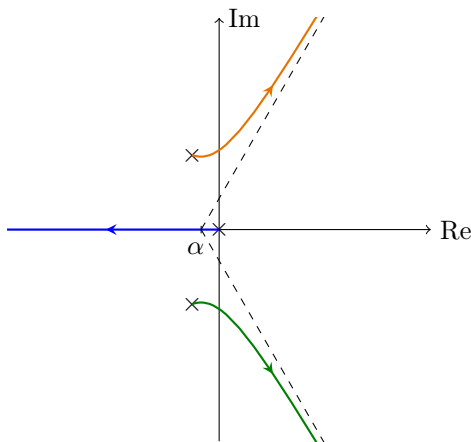


The real axis portion of the root locus is the entire negative real axis (since there is 1 open-loop zero to the right of these points, and 1 is odd). We know that as $k_d \rightarrow \infty$, one branch must go to the open-loop zero, and the other branch must go to infinity along the negative real axis. We do not know the details of how the poles do this, but we know they must meet at the real axis at some point, and then split. In this case, we happen to know that the complex poles move along a circle (the root-locus rules do not help here, but we can use what we know about second-order systems, as in the diagram on page 76), so the whole root locus diagram is as shown. \diamond

Example 4.5. Next, consider the same second-order system as in the previous examples, but now with integral feedback $C(s) = k_i/s$. We now wish to see how the closed-loop poles vary with $k_i \geq 0$. The closed-loop poles satisfy $1 + PC = 0$, or $1 + k_i G(s) = 0$, where

$$G(s) = \frac{1}{s(s^2 + 2\zeta\omega_0 s + \omega_0^2)},$$

so there are now 3 open-loop poles: the two from the previous example, plus an additional pole at $s = 0$. There are no open-loop zeros, so the relative degree is now 3, and there are 3 asymptotes, as shown in Figure 4.1. The centroid α is at the “center of mass” of the three poles, and the asymptotes are drawn as dashed curves in the diagram below:



The open-loop pole at $s = 0$ simply moves to the left as k_i increases. The other poles approach the asymptotes as shown in the diagram. If desired, there is a more advanced rule for computing the “departure angles” (the angles at which the branches leave the open-loop poles for small k_i), but we do not discuss that rule here: if you are interested, see [11, §5.2]. \diamond

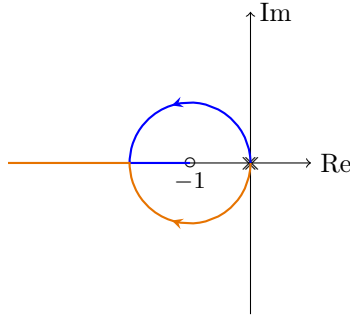
Example 4.6. Consider a double integrator

$$P(s) = 1/s^2.$$

Such a system arises, for instance, in the control of a satellite's orientation, since the angle θ of a satellite is given by $J\ddot{\theta} = u$, if u is a force provided by a thruster (or momentum wheel), and J is the moment of inertia. Suppose we want to control this system (for instance, tracking a reference angle) using a feedback loop with a PD controller $C(s) = K(1 + s)$. Let us see how the closed-loop poles vary with $K \geq 0$. The closed loop poles are roots of $1 + PC = 0$, or $1 + KG = 0$ with

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

There are thus two open-loop poles, both at $s = 0$, and one open-loop zero at $s = -1$. The real-axis portion is thus the portion of the real-axis to the left of the zero, as shown in the diagram below:



Note that there are *two* poles at the origin, and these must leave the real axis for $K > 0$. These branches then move to the left and again meet up on the real axis to the left of the zero, as shown in the figure. \diamond

Example 4.7. Consider again the double integrator from the previous example, but now instead of a PD controller $C(s) = K(1 + s)$, let us use a *lead compensator*, which looks like

$$C(s) = \frac{K(1 + s)}{s/10 + 1}.$$

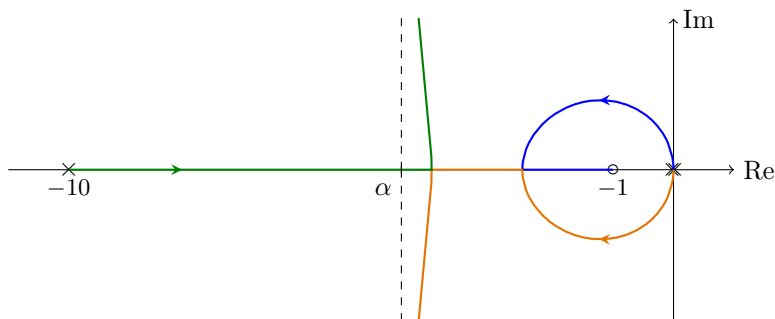
(This approximates the PD compensator $K(1 + s)$ for low frequencies, but has a finite gain for high frequencies.) Let us see what the closed-loop poles look like for this system, as K increases from 0 to $+\infty$. We now take

$$G(s) = \frac{s + 1}{s^2(s/10 + 1)},$$

so there are now three open-loop poles and one open-loop zero. The relative degree is now 2 (instead of 1 as in the previous example), so there are now two asymptotes, with a centroid given by

$$\alpha = \frac{0 + 0 + (-10) - (-1)}{2} = -4.5.$$

The resulting root-locus diagram is shown below:



Note that the new pole we have introduced moves to the right as K increases, until it meets up with one of the other poles, and they split to approach the vertical asymptotes. The behavior of the poles that start at $s = 0$ is actually quite similar to that in the previous example (except at very large values of K). \diamond

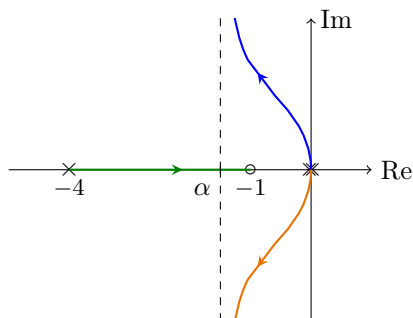
Example 4.8. Once again, consider the double integrator $P(s) = 1/s^2$, together with a lead compensator, but now move the pole in the lead compensator closer to the zero:

$$C(s) = \frac{K(1+s)}{s/4+1},$$

which has a pole at $s = -4$ instead of $s = -10$ as in the previous example. We now let

$$G(s) = \frac{s+1}{s^2(s/4+1)},$$

and plot the corresponding root locus diagram below:



Note that the plot has changed dramatically from the previous case: the poles that started at $s = 0$ no longer come together on the real axis, but rather go to the vertical asymptotes. We have no way of distinguishing between these two cases using our rules, so for these cases it is best to plot the root locus diagram using a computer program such as MATLAB. \diamond

Example 4.9. Suppose we are given a plant with some lightly damped poles:

$$P(s) = \frac{1}{(s + 0.1)^2 + 25},$$

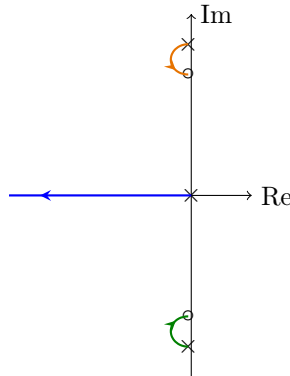
which has poles at $p = -0.1 \pm 5i$. This is a very lightly damped system, with $\zeta = 0.02$, as might arise for instance in the control of a flexible structure such as a satellite or a read-write head of a disk drive. We wish to use integral feedback, in order to have zero steady-state error to a step reference. If we use pure integral feedback, we know from Example 4.5 that this will immediately destabilize the system. Instead, we will use an integrator together with a *notch filter*, given by

$$C(s) = K \frac{(s + 0.1)^2 + 16}{s},$$

which has zeros at $z = -0.1 \pm 4i$. Let us see how the closed-loop poles vary with $K \geq 0$, to see whether our controller $C(s)$ can stabilize the closed-loop system. We let

$$G(s) = \frac{(s + 0.1)^2 + 16}{s((s + 0.1)^2 + 25)}$$

and plot the corresponding root locus diagram:

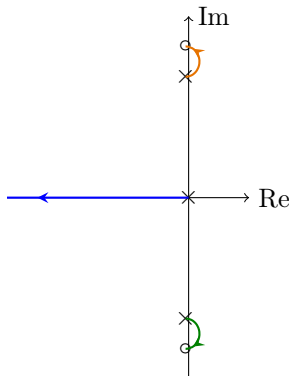


Note that the lightly damped poles happen to move towards the zeros in such a way that they become more heavily damped. So this controller should work reasonably well in practice. However, one must be careful in selecting the zeros for the notch filter, as the following example shows. \diamond

Example 4.10. Consider the same system as in the previous example, but swapping the location of the lightly damped poles and zeros:

$$G(s) = \frac{(s + 0.1)^2 + 25}{s((s + 0.1)^2 + 16)}$$

The root locus diagram is now as follows:



Note that now, the lightly damped poles move into the right-half plane, so the closed-loop system will be unstable for nearly all gains K . We cannot actually distinguish between these two cases using our rules (we need the additional rules for departure angles), but we will actually learn better ways of understanding the stability of these systems once we learn about the Nyquist stability criterion in Section 4.5. \diamond

The next example illustrates how one might design a PID controller in practice (though we will see better methods later), and shows that the transfer function $G(s)$ we use for the root locus method does not always need to be the loop gain.

Example 4.11. Suppose we start with the plant

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

a second-order system with no damping. We now know that one way to add damping to the system is to use a feedback loop with a PD controller

$$C(s) = k_p + k_d s.$$

Let us choose k_p and k_d such that the *desired* closed-loop characteristic polynomial is $s^2 + 2\zeta\omega_0 s + \omega_0^2$, with $\omega_0 = 2$, $\zeta = 1/\sqrt{2} \approx 0.7$. The *actual* closed-loop characteristic equation is

$$1 + P(s)C(s) = 1 + \frac{k_p + k_d s}{s^2 + 1} = 0,$$

so the actual characteristic polynomial is

$$s^2 + k_d s + k_p + 1.$$

Thus, in order to obtain the desired characteristic polynomial, we set $k_p + 1 = \omega_0^2$, which gives $k_p = 3$, and $k_d = 2\zeta\omega_0 = 2.8$.

Now, let us see how adding an integrator would change the dynamics of the closed-loop system. The new controller is

$$C(s) = k_p + k_d s + \frac{k_i}{s}.$$

Let us keep our original design for the proportional and derivative gains, with $k_p = 3$ and $k_d = 2.8$, and gradually increase the integral gain from $k_i = 0$. This corresponds to something you might do in practice: tune up a PID controller using the P and D terms, until the dynamics look reasonable, and then gradually turn up the integral gain, for instance to improve steady-state error, without making the transient dynamics too bad.

The closed-loop poles are solutions of

$$1 + PC = 1 + \frac{k_p + k_d s + k_i/s}{s^2 + 1} = 0.$$

We wish to plot solutions of this equation as k_i varies from 0 to ∞ . But there is a problem here: this equation is not of the form (4.5) (where here the parameter of interest is k_i). So it appears that we may not be able to use the root locus method for this problem. However, we can transform our characteristic equation to the form we need for root locus, as follows. First, multiply through by the denominator:

$$s^2 + k_d s + k_p + 1 + k_i/s = 0.$$

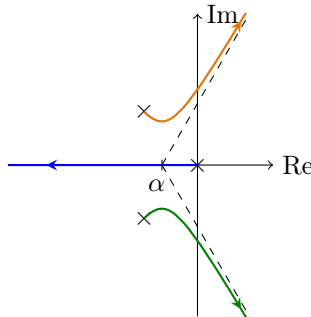
Now, divide through by $s^2 + k_d s + k_p + 1$, to obtain

$$1 + \frac{k_i}{s(s^2 + k_d s + k_p + 1)} = 0.$$

This expression is now in root-locus form $1 + k_i G(s) = 0$, with

$$G(s) = \frac{1}{s(s^2 + k_d s + k_p + 1)}.$$

The poles of G are simply the desired closed-loop poles from designing our PD controller, together with a pole at the origin. This makes sense, for when $k_i = 0$ we should recover our original PD controller. Overall, the root-locus plot is as shown below:

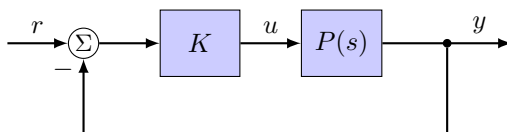


Note that the integrator does have a detrimental effect on the pole locations: it moves the poles further to the right, making the transients slower, and for large enough integral gain, the closed-loop poles move into the right-half plane, indicating an unstable closed-loop system. \diamond

4.3 Bode plots

In the previous section, we learned how a root-locus diagram can help us understand how the dynamics of a system change when feedback is introduced: in particular, it shows us the location of the closed-loop poles. This in turn tells us about the stability, the transient behavior (Section 3.1), and even time-domain characteristics such as overshoot and settling time (Section 3.3). This is indeed a useful and powerful tool, but it is only part of the story. Systems can have identical dynamics (i.e., identical closed-loop pole locations), and still behave very differently, as the following example shows.

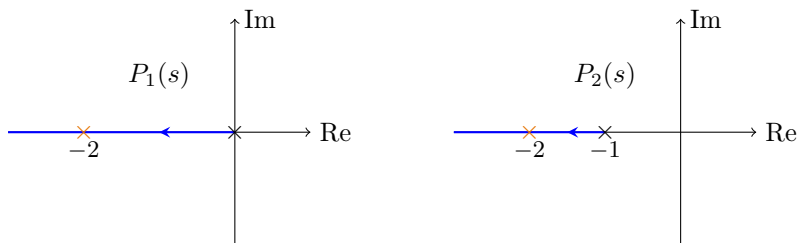
Example 4.12. Consider using proportional feedback to track a given reference signal $r(t)$, as indicated in the following diagram:



Let us compare two plants, P_1 and P_2 , both with proportional gain $K = 1$:

$$P_1(s) = \frac{2}{s}, \quad P_2(s) = \frac{1}{s+1}.$$

For the first system, the closed-loop poles satisfy $1 + P_1K = 0$, or $1 + 2/s = 0$, which has the single solution $s = -2$. For the second system, the closed-loop poles satisfy $1 + P_2K = 0$, which gives $1 + 1/(s+1) = 0$, which also has the single solution $s = -2$. So the closed-loop poles for the two systems are identical. The root locus plots for the two systems also look very similar:



Here, the closed-loop poles are also shown for $K = 1$: they are both at $s = -2$, so we would expect these systems to have the same “dynamics.” But

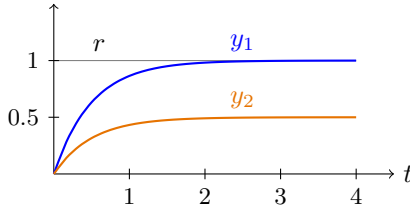
are the two closed-loop systems identical? Certainly not. The closed-loop transfer function (from r to y) is given by

$$T(s) = \frac{PK}{1 + PK},$$

so for the two different plants P_1 and P_2 , we have

$$T_1(s) = \frac{2/s}{1 + 2/s} = \frac{2}{s + 2}, \quad T_2(s) = \frac{1/(s + 1)}{1 + 1/(s + 1)} = \frac{1}{s + 2}.$$

The denominators are the same (as we already knew they would be, since the closed-loop poles are the same), but the numerators differ. The closed-loop responses to a step in the reference r are shown below:



The first system has zero steady-state error ($y(t)$ has a steady-state value of 1, matching the reference $r(t) = 1$), while the second system has 50% tracking error! Even though the closed-loop poles are the same (so the *dynamics* are the same), the two feedback loops have very different behavior with respect to reference tracking. \diamond

In order to understand differences such as this, we need a better method. Probably the single most useful tool for analyzing feedback systems is the *Bode plot*. Let us compare the Bode plots for the two systems in Example 4.12. Figure 4.2 shows the Bode plot for the *loop gain* $L(s) = KP(s)$ for the two systems. (Here, since $K = 1$, the loop gain is just $L(s) = P(s)$.)

First, let us focus on the magnitude plot (the top plot). As we mentioned in Section 2.6, we want the loop gain to be *large* ($|L(i\omega)| \gg 1$) at frequencies where we require good tracking performance (normally at low frequencies). This is because the transfer function from r to the tracking error $r - y$ is $S = 1/(1 + L)$ (the sensitivity function), so making $|L(i\omega)| \gg 1$ makes the tracking error small. In addition, we wanted $|L(i\omega)|$ to be *small* at high frequencies, in order to avoid too much amplification of sensor noise. The Bode plot for the first system thus looks ideal: at low frequencies ($\omega \ll 2$), the loop gain $L_1 = KP_1$ is large (in fact, it is infinite at zero frequency), so one should expect good tracking performance, while for high frequencies ($\omega \gg 2$), the loop gain is small, so we should not amplify sensor noise too badly. However, for the second system, the loop gain $L_2 = KP_2$ is never large: even at low frequencies, the magnitude is only 1 (0 dB), while we

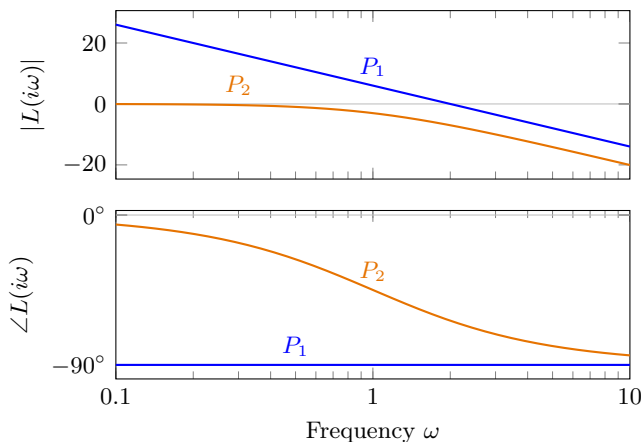


Figure 4.2: Bode plots of the loop gains $L(s)$ for the two systems in Example 4.12.

wanted $|L(i\omega)| \gg 1$. Thus, from the Bode plot of the loop gain, we expect the first system to perform much better than the second system: indeed, it does perform better, as we have already seen from the step response.

At this point, it is not immediately clear that the phase plot (the lower plot) gives any useful information, but we will see in Section 4.5 that the phase is critical for determining the stability of the closed-loop system.

As the previous example illustrates, Bode plots are a very useful tool for designing and analyzing feedback systems. In this section, we will learn some rules for sketching Bode plots by hand. As with the root locus method, our main goal is not to be able to draw extremely accurate Bode plots by hand (the `bode` command in MATLAB is very good at this), but rather to understand how the Bode plot changes as we modify some aspect of the feedback loop: for instance, increasing the overall gain, or moving the location of a zero or pole in our controller. Once we learn the basic rules, we will then be able to *design* controllers using Bode plots: we will know how to modify our controller in order to give us some desired characteristics we want our Bode plots to have.

Sketching Bode plots

Recall from Section 2.1 that the Bode plot of a transfer function $G(s)$ is really two plots: the magnitude plot and the phase plot. The magnitude plot shows $|G(i\omega)|$ as a function of ω , with ω plotted on a log scale, and $|G(i\omega)|$ either plotted on a log scale or given in decibels (dB). The phase plot shows $\angle G(i\omega)$ versus ω , with ω again plotted on a log scale. One of the convenient features of working with this type of plot (log-log for magnitude, and log-linear for

phase) is the following:

When transfer functions are multiplied, their Bode plots add.

Let us be more precise about what exactly this means. Suppose

$$G(s) = A(s)B(s),$$

and suppose we already know what the Bode plots of $A(s)$ and $B(s)$ look like. We wish to construct the Bode plot of $G(s)$. First, note that

$$\log |G(i\omega)| = \log |A(i\omega)B(i\omega)| = \log |A(i\omega)| + \log |B(i\omega)|.$$

So the value of $|G(i\omega)|$ in decibels (given by $20 \log_{10} |G(i\omega)|$) is just the sum of values of $|A(i\omega)|$ and $|B(i\omega)|$, given in decibels. Similarly, the phase of $G(i\omega)$ is given by

$$\angle G(i\omega) = \angle(A(i\omega)B(i\omega)) = \angle A(i\omega) + \angle B(i\omega),$$

so to construct the Bode phase plot of G , we can just add the phase plots of A and B .

Similarly, taking the reciprocal of a transfer function corresponds to flipping the Bode plot vertically: that is, one simply changes the sign of the magnitude plot (in dB), and phase plot. To see why this is so, suppose

$$G(s) = \frac{1}{H(s)}.$$

Then

$$\log |G(i\omega)| = \log (1/|H(i\omega)|) = -\log |H(i\omega)|,$$

so the Bode magnitude plot of G is the same as the Bode magnitude plot of H , but with the sign reversed. Similarly,

$$\angle G(i\omega) = \angle(1/H(i\omega)) = -\angle H(i\omega),$$

so the phase plot of G is also the same as that of H , with the sign reversed.

Note that multiplication of transfer functions is very common: if we connect a plant $P(s)$ in series with a controller $C(s)$, then we multiply their transfer functions, so we add their Bode plots.

Below, we will describe the Bode plots for some simple systems, consisting of a single pole or zero, or a pair of complex poles/zeros. These simple systems can then be combined into more complicated systems, simply by multiplying them together, and adding the Bode plots.

To begin, we will consider only systems with no right-half-plane (RHP) poles or zeros. Bode [3] referred to these as *minimum-phase* systems, because

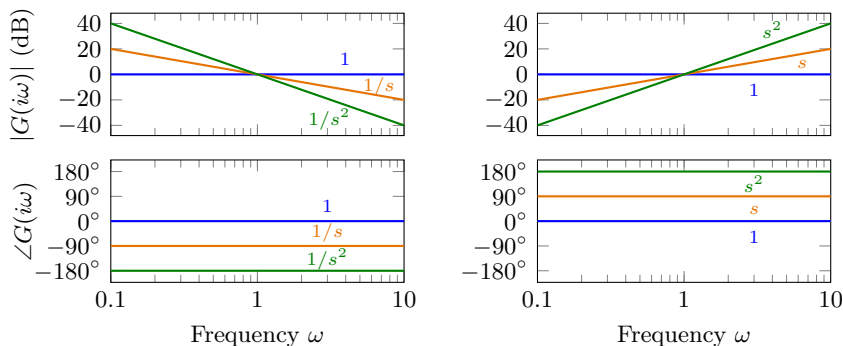


Figure 4.3: Bode plots for transfer functions of the form $G(s) = s^k$.

they have a smaller phase lag than the corresponding systems with right-half-plane poles or zeros (that is, with the RHP poles/zeros mirrored about the imaginary axis). Later we will see how to handle RHP poles and zeros (see page 105).

First, consider transfer functions of the form

$$G(s) = s^k, \quad k = 0, \pm 1, \pm 2, \dots$$

For instance, the constant $G(s) = 1$ is a special case of this, for $k = 0$, and the integrator $G(s) = 1/s$ is another special case, for $k = -1$. For transfer functions of this form, we find (for real $\omega > 0$)

$$|G(i\omega)| = |(i\omega)^k| = |i\omega|^k = \omega^k,$$

so

$$\log |G(i\omega)| = k \log \omega,$$

and the magnitude plot (on a log-log scale) is a straight line with slope k . The phase is also easy to calculate:

$$\angle G(i\omega) = \angle (i\omega)^k = \angle (\omega e^{i\pi/2})^k = \angle e^{ik\pi/2} = \frac{k\pi}{2}.$$

So the phase is simply the constant $k\pi/2$, or $90^\circ \cdot k$. The corresponding Bode plots are shown in Figure 4.3.

Real poles and zeros

Next, consider a transfer function with a real pole:

$$G(s) = \frac{a}{s + a}$$

with $a > 0$. For low frequencies ($\omega \ll a$), we have

$$G(i\omega) = \frac{1}{i\omega/a + 1} \approx 1,$$

so for $\omega \ll a$, the magnitude is approximately 1, and the phase is approximately 0. Conversely, for high frequencies ($\omega \gg a$), we have

$$G(i\omega) = \frac{a}{i\omega + a} \approx \frac{a}{i\omega},$$

so for $\omega \gg a$, the transfer function is closely approximated by $G(s) = a/s$. From Figure 4.3, we know that the magnitude plot has a slope of -1 (or -20 dB/decade), and the phase is approximately -90° . The corresponding low-frequency and high-frequency asymptotes are shown in Figure 4.4: note that the asymptotes intersect at $\omega = a$, which is often called the *corner frequency* or *breakpoint*. The overall Bode plot is shown in Figure 4.4: note that the magnitude curve smoothly transitions between the two asymptotes. Furthermore, at the corner frequency, we have

$$|G(ia)|^2 = \left| \frac{a}{ia + a} \right|^2 = \frac{1}{1 + 1} = \frac{1}{2},$$

so $|G|$ is $1/\sqrt{2}$, or about -3 dB. The phase is zero for low frequencies ($\omega \ll a$) and -90° for high frequencies ($\omega \gg a$). At $\omega = a$, the phase is -45° , and the transition from 0° to -90° happens over a frequency range of about two decades (from $\omega = a/10$ to $10a$). A useful approximation is that the phase change is approximately linear, at -45° per decade in frequency, from $\omega = a/10$ to $10a$. This approximation is also shown in Figure 4.4.

Next, consider a transfer function with a real zero:

$$G(s) = \frac{s + a}{a}.$$

Since this transfer function is the reciprocal of the one just studied, we know the Bode plot is the same as that shown in Figure 4.4, but with the sign reversed (on both the gain in dB and the phase). It is shown in Figure 4.5. Note that now, at the corner frequency $\omega = a$, the slope of the magnitude plot *increases*, and the phase also increases from 0° to 90° .

Complex poles and zeros

Next, consider a transfer function with a complex pair of poles:

$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (4.10)$$

where $\omega_0 > 0$ and $0 \leq \zeta \leq 1$. We know from Section 3.2 that these poles have a radius of ω_0 , and their angle with the imaginary axis is given by $\sin^{-1}\zeta$. What does the Bode plot look like?

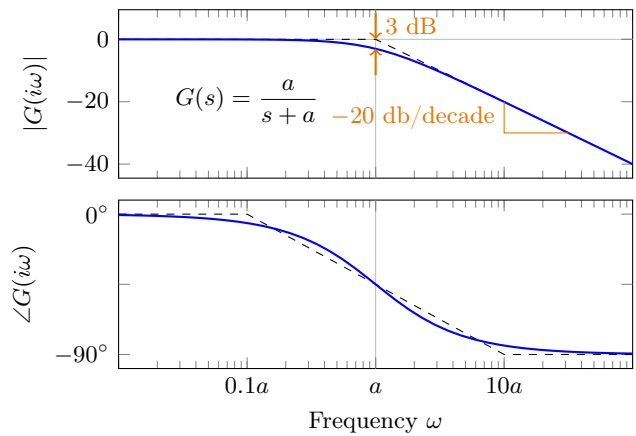


Figure 4.4: Bode plot of a transfer function with a real pole at $s = -a$, with asymptotes plotted as dashed lines.

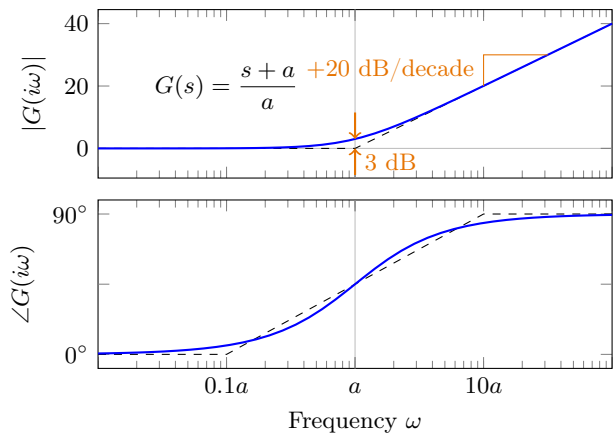


Figure 4.5: Bode plot of a transfer function with a real zero at $s = -a$, with asymptotes plotted as dashed lines.

For low frequencies ($\omega \ll \omega_0$), we have

$$G(i\omega) = \frac{1}{(i\omega/\omega_0)^2 + 2\zeta\omega/\omega_0 + 1} \approx 1,$$

so the magnitude is approximately 1, and the phase is approximately 0. Conversely, for high frequencies ($\omega \gg \omega_0$), we have

$$G(i\omega) \approx \frac{1}{(i\omega/\omega_0)^2} = \frac{\omega_0^2}{(i\omega)^2},$$

so for $\omega \gg \omega_0$, the transfer function is closely approximated by $G_{\text{hf}}(s) = \omega_0^2/s^2$. From Figure 4.3, we know that this magnitude plot has a slope of -2 (or -40 dB/decade), and the phase is approximately -180° . Furthermore, the magnitude plot for G_{hf} intersects the 0 dB line at frequency ω_0 , since $|G_{\text{hf}}(i\omega_0)| = |\omega_0^2/(i\omega_0)^2| = 1$. The corresponding low-frequency and high-frequency asymptotes are shown in Figure 4.6.

In several respects, the plot in Figure 4.6 is similar to the Bode plot for a real pole, as we can see by comparing with Figure 4.4: both plots have the same low-frequency behavior, with constant magnitude and zero phase. Both magnitude curves have a decrease in slope at some corner frequency ($\omega = a$ for the real pole, and $\omega = \omega_0$ for the complex pair). After this decrease, the plot for the real pole has a slope of -1 , while the plot for the complex pair has a slope of -2 (this makes perfect sense, since the complex pair is really *two* poles, so should have an effect close to that of two real poles). The phase changes from 0° before the corner frequency to -180° after the corner frequency, again consistent with what we would expect for two real poles.

However, the Bode plot for the complex poles also shows a new feature: a resonant peak close to $\omega = \omega_0$. The height of this peak corresponds to the value of the damping ratio ζ . For $\zeta = 0$, the peak is infinitely high: $|G(i\omega)|$ has a singularity at $\omega = \omega_0$. Furthermore, for $\zeta = 0$, the phase plot is discontinuous at $\omega = \omega_0$: the phase jumps abruptly from 0° for $\omega < \omega_0$ to -180° for $\omega > \omega_0$. As ζ increases, the resonant peak becomes smaller, and the phase change becomes more gradual, until for $\zeta = 1$ there is no peak at all, and the phase change spans about a decade in each direction (from $\omega_0/10$ to $10\omega_0$), as it does for the real pole.

Now, consider a transfer function with a pair of complex zeros:

$$G(s) = \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{\omega_0^2}.$$

As with the case of a real zero, we notice that this transfer function is the reciprocal of the one just studied, so its Bode plot is the same as that shown in Figure 4.6, but with the sign reversed. It is shown in Figure 4.7. Now at the corner frequency $\omega = \omega_0$, the slope of the magnitude plot *increases* by 2, and the phase increases from 0° to 180° . For small ζ , there is also a resonant valley near $\omega = \omega_0$.

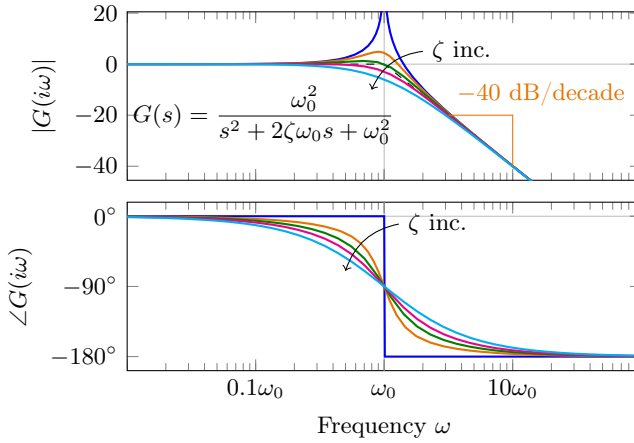


Figure 4.6: Bode plot of a transfer function with a pair of complex poles, for $\zeta = 0, 0.3, 0.5, 0.7$, and 1.0 . Asymptotes are plotted as dashed lines.

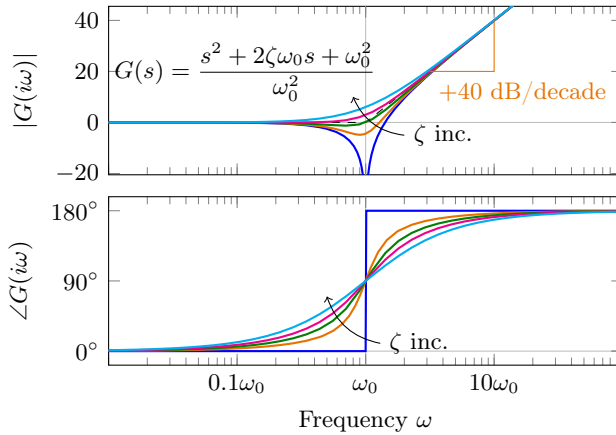


Figure 4.7: Bode plot of a transfer function with a pair of complex zeros, for $\zeta = 0, 0.3, 0.5, 0.7$, and 1.0 . Asymptotes are plotted as dashed lines.

Summary

We now summarize the Bode plots for these basic minimum-phase systems:

For a system with a real pole/zero at $s = -a$:

- The slope of the Bode magnitude plot decreases/increases by 1 at $\omega = a$.
- The phase decreases/increases by 90° at $\omega = a$, over a broad range of frequencies spanning about a decade in either direction (from $\omega = a/10$ to $10a$).

For a system with a complex pair of poles/zeros, which are roots of $s^2 + 2\zeta\omega_0 s + \omega_0^2$ (with $\omega_0 > 0$, $\zeta \geq 0$):

- The slope of the Bode magnitude plot decreases/increases by 2 at $\omega = \omega_0$. If ζ is small, there will be a resonant peak/valley near $\omega = \omega_0$, and the height of this peak/valley goes to infinity for $\zeta = 0$.
- The phase decreases/increases by 180° at $\omega = \omega_0$. For $\zeta = 0$, the phase change is discontinuous, and it becomes more gradual as ζ increases.

Now that we see how the Bode plots behave for real poles/zeros, and pairs of complex poles/zeros, we can build up Bode plots for more complicated functions simply by adding them together. We illustrate with a few examples below.

Example 4.13 (DC motor). Let us sketch a Bode plot for the DC motor, which has a transfer function given by

$$P(s) = \frac{K}{\tau s + 1}$$

where $K, \tau > 0$. When sketching Bode plots, it often helps to start with the zero-frequency gain: as long as there are no poles at $s = 0$, we can find the zero-frequency gain simply by plugging in $s = 0$, which here gives $P(0) = K$. So for low frequencies, we know the Bode magnitude plot has the constant value K , with phase zero.

Next, we note the corner frequencies (breakpoints): we have a real pole at $-1/\tau$, so the corner frequency is $\omega = 1/\tau$. Since this is a real pole, the slope of the magnitude plot decreases by 1 at this frequency, and the phase decreases by 90° . The complete Bode plot is shown in Figure 4.8. \diamond

Example 4.14 (Lead compensator). Next, consider the following transfer function

$$C(s) = \frac{s + 1}{s + 100}.$$

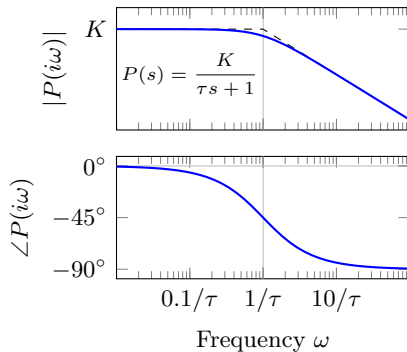


Figure 4.8: Bode plot for Example 4.13. The asymptote is plotted as a dashed line: for low frequencies, the transfer function has a magnitude of K with zero phase; at the break frequency $1/\tau$, the slope decreases by -1 (or -20 dB/decade), and the phase decreases by 90° (spread over about a decade in frequency).

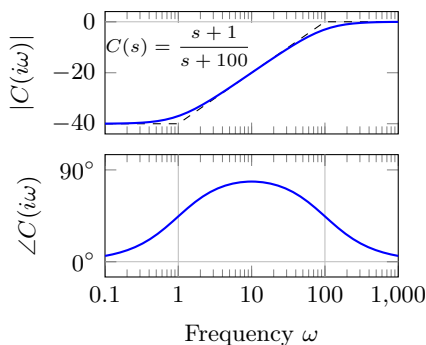


Figure 4.9: Bode plot for Example 4.14, with asymptotes plotted as dashed lines. The zero-frequency gain is $1/100$, or -40 dB. There is a zero at $s = -1$, so at the corner frequency $\omega = 1$, the slope of the magnitude plot increases by 1 (or 20 dB/decade), and the phase increases by 90° (spread over about a decade in frequency). There is a pole at $s = -100$, so at $\omega = 100$, the slope of the magnitude plot decreases by 1 , returning the slope to 0 , and the phase decreases back to 0° as well.

If a controller has a transfer function of this form, it is called a *lead compensator*, for reasons that we will see shortly. To sketch this Bode plot, we begin by considering the zero-frequency gain, which here is $C(0) = 1/100$. So at low frequencies (below those of any of the poles or zeros), the magnitude is -40 dB, with zero phase. There are corner frequencies at $\omega = 1$, corresponding to the zero at $s = -1$, and $\omega = 100$, corresponding to the pole at $s = -100$. The overall Bode plot is shown in Figure 4.9.

From the Bode plot, we see why such a controller is called a lead compensator. Note that the phase is always *positive*: this positive phase (known as *phase lead*) can be desirable for closed-loop stability, as we will see in Section 4.6, so such a controller (or compensator) can be useful in stabilizing underdamped systems.

◇

Example 4.15 (Lag compensator). Consider a controller of the form

$$C(s) = 2 \frac{s + 10}{s + 1},$$

which is called a *lag compensator*. The Bode plot of $C(s)$ is sketched in

Figure 4.10: Bode plot for Example 4.15. The zero-frequency gain is 20, which is 26 dB. There are two corner frequencies: $\omega = 1$ (a real pole at $s = -1$), and $\omega = 10$ (a real zero at $s = -10$). At $\omega = 1$, the slope decreases by 1 and the phase decreases by 90° . However, the phase never quite makes it to -90° , because the next breakpoint at $\omega = 10$ starts to bring the phase back to 0° before it reaches -90° .

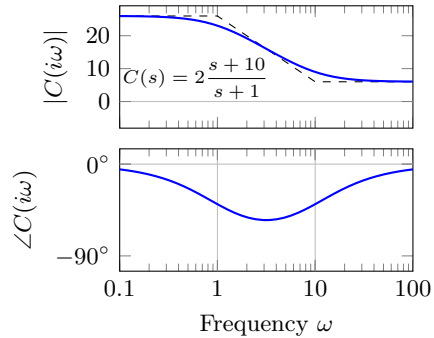


Figure 4.10. Note that the zero-frequency gain is 20 (or 26 dB), and the high-frequency gain is 2 (or 6 dB). Now, unlike Example 4.14, the corner frequency for the pole ($\omega = 1$) is lower than the corner frequency of the zero ($\omega = 10$), so the slope of the magnitude plot first decreases (“breaks down”) before it increases again (“breaks up”) to zero slope. Note that this controller has the shape we typically like a controller to have: large gain at low frequencies (for good performance), and smaller gain at high frequencies (to avoid amplifying high-frequency noise). However, note also that the phase of $C(s)$ is negative, so the controller adds some phase lag to the system. (This is why it is called a lag compensator.) We will see later that such negative phase is bad for stability of a closed-loop system, but a controller like this can work well for systems like a first-order plant that do not have a problem with closed-loop stability.

◇

Example 4.16 (PI controller). Now, let us sketch the Bode plot for a PI controller, with transfer function

$$C(s) = k_p + \frac{k_i}{s} = k_p \frac{s + k_i/k_p}{s}$$

We see there is a pole at $s = 0$ and a zero at $s = -k_i/k_p$. The zero frequency gain is now infinite (because of the pole at $s = 0$), but we may still tell the overall gain either by looking at the value for high frequencies (at which $C(i\omega) \rightarrow k_p$) or by looking at the low-frequency asymptote $C(i\omega) \rightarrow k_i/(i\omega)$, which intersects 0 dB at $\omega = k_i$.

Recall that an alternative way to represent a PI controller is

$$C(s) = k_p \left(1 + \frac{1}{\tau_i s} \right) = k_p \frac{\tau_i s + 1}{\tau_i s},$$

and in this form the corner frequency is $1/\tau_i$, while the high-frequency gain is still k_p .

◇

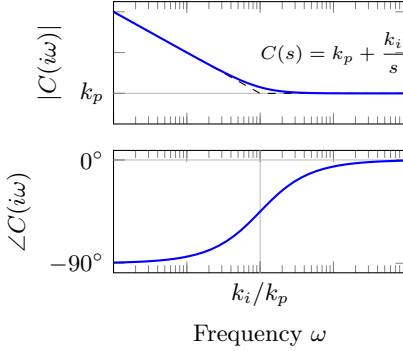


Figure 4.11: Bode plot for Example 4.16. There is a pole at $s = 0$, so for low frequencies, the Bode plot resembles k_i/s , and has a slope of -1 and a phase of -90° . There is a break-point (corresponding to a real zero) at $\omega = k_i/k_p$ at which the slope increases by 1 and the phase increases by 90° . At high frequencies, $C(i\omega) \rightarrow k_p$.

Bode gain-phase relation

We now look at an important relationship between the magnitude and phase plots. It turns out that, in many cases, the phase of the frequency response is uniquely determined by the magnitude, a result due to Bode [3, p. 313]. That is, given only the Bode magnitude plot, you can figure out the corresponding phase plot.

At first, it may seem as though such a result is not possible. For instance, the three transfer functions below all have identical magnitude plots, but different phase plots:

$$G_1(s) = \frac{1}{1+s} \quad G_2(s) = \frac{1}{1-s} \quad G_3(s) = \frac{2-s}{(1+s)(2+s)}.$$

Thus, we will certainly need some restrictions on the transfer functions to be considered. In particular, we will consider only transfer functions with no poles or zeros in the right half-plane (i.e., minimum-phase systems). This rules out transfer functions G_2 and G_3 , which have a right-half-plane pole and zero, respectively. The sign of G must also be chosen appropriately, since for instance

$$G_4(s) = \frac{-1}{1+s}$$

also has the same magnitude plot as G_1 , but the phase differs by 180° at all frequencies. An additional condition is therefore also needed, namely that $G(\sigma) > 0$ for some real $\sigma \geq 0$. This then rules out G_4 , since $G_4(0) < 0$. (Note that, since G has no zeros in the right half-plane and is continuous there, $G(\sigma)$ must have the same sign for all $\sigma > 0$, so it suffices to check the sign at a single point.)

The Bode gain-phase relation (derived in Appendix B.4) says that, as long as $G(s)$ satisfies the conditions above, then defining $\lambda = \log(\omega/\omega_0)$ for some frequency ω_0 , we have

$$\angle G(i\omega_0) = \frac{\pi}{2} \int_{-\infty}^{\infty} \frac{d \log |G(i\omega_0)|}{d\lambda} \rho(\lambda) d\lambda, \quad (4.11)$$

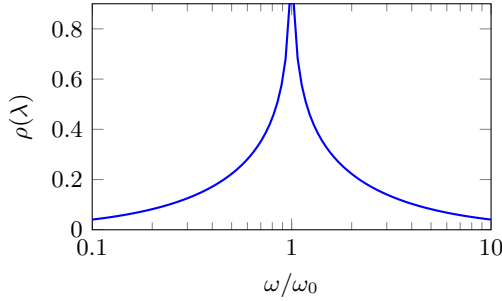


Figure 4.12: Weighting function ρ used in Bode's gain-phase relation.

where the function ρ is given by

$$\rho(\lambda) = \frac{2}{\pi^2} \log \coth \frac{|\lambda|}{2} = \frac{2}{\pi^2} \log \left| \frac{\omega + \omega_0}{\omega - \omega_0} \right|.$$

The quantity $d \log |G|/d\lambda$ is simply the *slope* of the Bode magnitude plot (since $|G|$ and ω are both plotted on a log scale), so this result says that the phase of the frequency response is determined uniquely by the slope of the magnitude plot.

The function ρ satisfies

$$\int_{-\infty}^{\infty} \rho(\lambda) d\lambda = 1,$$

so the integral in (4.11) may be viewed as a *weighted average* of the slope $d \log |G|/d\lambda$ over all frequencies (note that λ ranging from $-\infty$ to ∞ corresponds to ω ranging from 0 to ∞). The weighting function ρ is plotted in Figure 4.12. Note that ρ strongly weights frequencies close to ω_0 (in fact, it becomes infinite at $\omega = \omega_0$), but then falls off for frequencies far from ω_0 . Thus, if the slope of the magnitude plot is roughly constant in a region around frequency ω_0 , then (4.11) gives

$$\angle G(i\omega_0) \approx \frac{\pi}{2} \frac{dA}{du} \Big|_{u=0} = \frac{\pi}{2} \frac{d \log |G(i\omega)|}{d \log \omega} \Big|_{\omega=\omega_0}.$$

So if the slope of the Bode magnitude plot is n , the phase is approximately $n\pi/2$, or $90^\circ \cdot n$. This gives us a simplified version of the Bode gain-phase relation:

If the slope of the Bode magnitude plot is n (and is roughly constant in a region about a particular frequency), then the phase at that frequency is $n\pi/2$, or $90^\circ \cdot n$.

Note that this is consistent with all of the Bode plots we have seen so far: in Figure 4.3, we saw that the Bode magnitude plots of s^k has slope k , and the phase is constant at $90^\circ \cdot k$. For a real pole, shown in Figure 4.4, the slope is zero before the corner frequency, and the phase is also zero, while above the corner frequency, the slope is -1 (or -20 dB/decade), and the phase is -90° . You should verify this relation for the other Bode plots in this section. The Bode gain-phase relation is a useful one to remember, as it enables one to easily recall (or deduce) the phase plots for all of these transfer functions.

Non-minimum-phase systems

All of the examples we have looked at above had something in common: all of the poles and the zeros were in the left half of the complex plane. As we mentioned previously, such systems are called *minimum phase* systems. If a transfer function has a pole or zero in the right-half plane, it is called *non-minimum phase*. Of course, we will need new rules for sketching Bode plots for non-minimum phase systems. However, do not despair: the rules are almost the same as the rules for the corresponding minimum-phase system (that is, with any right-half-plane poles or zeros reflected about the imaginary axis to the left half plane). The general rule is as follows:

If a system has a right-half plane pole, zero, or complex pair of poles/zeros, the Bode magnitude plots are the same as for the corresponding system with a left-half-plane pole/zero (i.e., mirrored about the imaginary axis), but the phase is reversed.

Thus, for systems with right-half-plane poles

$$G(s) = \frac{a}{-s + a}, \quad G(s) = \frac{\omega_0^2}{s^2 - 2\zeta\omega_0 s + \omega_0^2},$$

(with $a, \zeta, \omega > 0$) or right-half plane zeros

$$G(s) = \frac{-s + a}{a} \quad G(s) = \frac{s^2 - 2\zeta\omega_0 s + \omega_0^2}{\omega_0^2},$$

the Bode magnitude plots are identical to those shown in Figures 4.4–4.7, but the phase plots are reversed in sign. Two examples are shown in Figure 4.13.

4.4 Loop transfer function

The main object we study in frequency-domain analysis is the loop transfer function (or loop gain), which we typically denote $L(s)$. We already have some idea why the loop gain is so important: for instance, in Section 2.6,

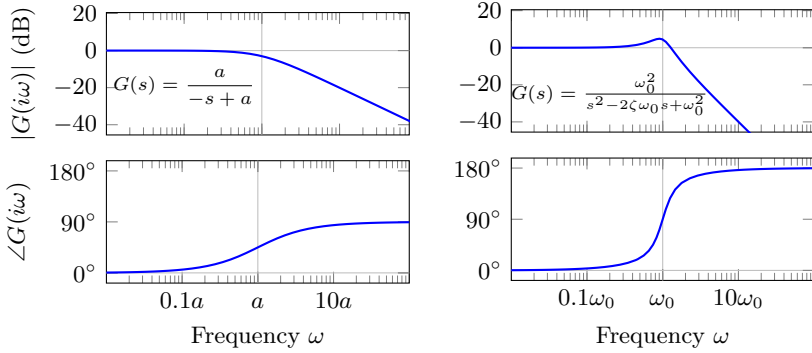
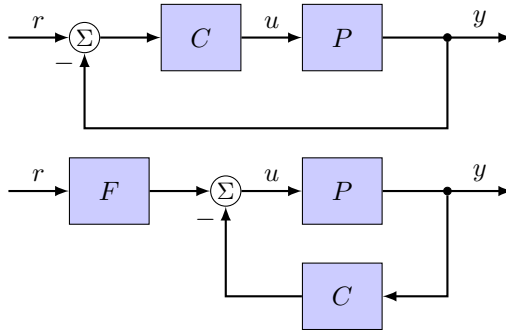


Figure 4.13: Bode plots for transfer functions with right-half-plane (RHP) poles. *Left:* a real RHP pole at $s = a$; *Right:* a complex pair of RHP poles at radius ω_0 (here, $\zeta = 0.3$). Note that the gain plot is the same as that for the corresponding system with LHP poles (see Figures 4.4 and 4.6) but the phase plots are reversed.

we saw that the loop gain $L = PC$ arose in the denominator of all of the closed-loop transfer functions known as the “gang of four”. It turns out that many important characteristics of feedback systems may be determined from the loop transfer function.

To appreciate why it is useful to work with the loop transfer function, let us begin by considering the typical feedback loops shown in the following block diagrams:



In both of the above examples, the loop gain is $L = PC$. Note that we can modify L *directly* by our choice of controller C : for instance, if we would like L to have a pole at a particular location, we can include such a pole in C . The loop gain is an *open-loop* transfer function: it is the transfer function around the loop, but before the loop has actually been closed. The poles and zeros of L will always be the same as the poles and zeros of the plant P (which we are given), together with the poles and zeros of C (which we get to choose).

By contrast, the closed-loop transfer functions such as

$$S = \frac{1}{1 + PC}, \quad T = \frac{PC}{1 + PC}$$

vary in a complicated way as we vary C : for instance, the closed-loop poles may move in complex ways as we change C (this is why we introduced the root-locus method). However, we can *infer* a great deal about the closed-loop system by studying L . This is, in fact, what we did with the root-locus method: based on properties of the *open-loop* transfer function L (namely, its poles and zeros), we deduced the possible locations of *closed-loop* poles.

Performance from the loop transfer function

Recall from Section 2.6 that a useful way to quantify the performance of a feedback system is through the sensitivity function

$$S = \frac{1}{1 + L},$$

where L is the loop gain. In particular, we would like $|S(i\omega)| \ll 1$ for good performance. For a tracking problem, this would then make the tracking error small, since S is the transfer function from the reference input to the tracking error. At the same time, for a disturbance rejection problem, making $|S(i\omega)|$ small makes the system less sensitive to disturbances, since $|S|$ is the amount by which disturbances are attenuated, relative to the open-loop system (see Section 2.6). So in either case, we want $|S(i\omega)| \ll 1$.

On the other hand, recall that we do not want the loop gain to be large at high frequencies, or else our controller will amplify sensor noise too much. At high frequencies (i.e., frequencies higher than those of the reference signals we wish to track, or disturbances we wish to reject), we want the *complementary* sensitivity function

$$T = \frac{L}{1 + L}$$

to be small. Of course, we cannot have both $|S(i\omega)|$ and $|T(i\omega)|$ small at the same frequency, since $S + T = 1$. One of the main ideas of control design is to choose a controller $C(s)$ such that $|S|$ is small for frequencies at which we desire good performance, and $|T|$ is small for frequencies at which we do not. So it is of great importance to be able to understand how $|S|$ and $|T|$ behave as we change the loop gain $L = PC$.

Let us try to approximate the closed-loop magnitude plots $|S(i\omega)|$ and $|T(i\omega)|$ using the Bode magnitude plot of L . First, consider the case where $|L(i\omega)| \gg 1$. Then $|1 + L(i\omega)| \approx |L(i\omega)|$, so

$$\begin{aligned} |S(i\omega)| &= \left| \frac{1}{1 + L(i\omega)} \right| = \frac{1}{|1 + L(i\omega)|} \approx \frac{1}{|L(i\omega)|} \\ |T(i\omega)| &= \left| \frac{L(i\omega)}{1 + L(i\omega)} \right| = \frac{|L(i\omega)|}{|1 + L(i\omega)|} \approx 1. \end{aligned}$$

Conversely, if $|L(i\omega)| \ll 1$, then $|1 + L(i\omega)| \approx 1$, so

$$|S(i\omega)| = \frac{1}{|1 + L(i\omega)|} \approx 1$$

$$|T(i\omega)| = \frac{|L(i\omega)|}{|1 + L(i\omega)|} \approx |L(i\omega)|.$$

So, at least when $|L(i\omega)|$ is either large or small, we can determine useful approximations of the *closed-loop* transfer functions $|S|$ and $|T|$ directly from the Bode plot of L :

$$\begin{aligned} \text{If } |L| \gg 1, \quad \text{then } |S| &\approx \frac{1}{|L|}, \quad |T| \approx 1. \\ \text{If } |L| \ll 1, \quad \text{then } |S| &\approx 1, \quad |T| \approx |L|. \end{aligned}$$

Thus, for good performance at low frequencies, we should design our controller $C(s)$ so that the loop gain $L(s)$ will have *large gain* at frequencies of interest (typically low frequencies), and *small gain* at other frequencies (typically high frequencies). Let us illustrate with an example.

Example 4.17. Consider a first-order plant given by

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

with two different choices of controllers:

$$C_1(s) = 10, \quad C_2(s) = 1 + \frac{1.25}{s}.$$

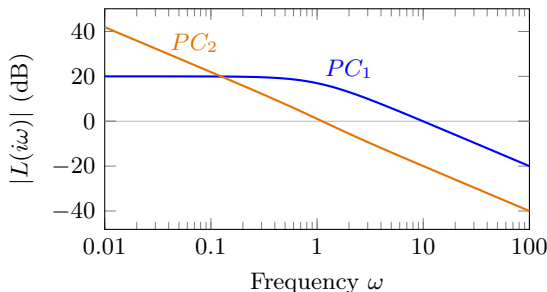
It is clear that C_1 is a proportional controller, and C_2 is a PI controller. Which is the better controller for this plant?

First, we might look at the closed-loop poles: are the closed-loop systems even stable for either of the controllers? Solving $1 + PC_1 = 0$ gives $s = -11$, so the closed-loop system with C_1 is stable. Solving $1 + PC_2 = 0$ gives

$$1 + \frac{1}{s+1} \left(1 + \frac{1.25}{s} \right) = 0 \implies s^2 + 2s + 1.25 = 0,$$

whose roots are a complex pair, $s = -1 \pm 0.5i$. This is also a reasonable location for the closed-loop poles: most importantly, they are in the left-half plane (so the closed-loop system is stable); the real part is the same as that of the plant's pole at $s = -1$, and the damping ratio is not too small, with $\zeta = 1/\sqrt{1 + 0.5^2} \approx 0.9$, so there will be very little overshoot in the closed-loop step response. Based on the closed-loop poles, both controllers look fine, although we may be inclined to choose C_1 , since its dynamics are faster (pole further in the left-half plane, and no overshoot at all).

However, now let us look at the Bode plots of the loop gain $L = PC$ for the two systems:



For low frequencies, the loop gain PC_1 goes to 20 dB, or a factor of 10. Since $10 \gg 1$, our approximation $|S| \approx 1/|L|$ should be valid, and we find $|S(i\omega)| \approx 0.1$ for low frequencies. Thus, there will be about a 10% tracking error for low frequencies. For PC_2 , however, the loop gain goes to infinity as $\omega \rightarrow 0$, so $|S| \rightarrow 0$, and there is zero tracking error for a step input ($\omega = 0$). This should come as no surprise, since we know integral feedback gives zero steady-state error to a step input. In fact, at all frequencies below 0.1 rad/sec, the tracking will be better for C_2 than for C_1 (since the loop gain is larger, and therefore $|S|$ is smaller).

Next, let us look at high frequencies. Here, as long as the magnitude of the loop gain is small, we can approximate $|T| \approx |L|$, so we see that $|T|$ is a factor of 10 (20 dB) smaller for C_2 than for C_1 . Thus, C_1 will amplify sensor noise $10\times$ more than C_2 .

Looking at the Bode plots of the loop gain, then, we see that controller C_2 now looks superior to C_1 in almost every way: it has better low-frequency tracking, and better high-frequency noise characteristics. Of course, which controller is actually “better” really depends on the specifications, since if the specifications were to track frequencies at $\omega = 1$ rad/sec, then C_1 actually does better at this frequency. But for most reasonable specifications, C_2 is superior to C_1 , for the reasons described above. \diamond

The above example illustrates how a Bode plot of the loop gain can help us determine the performance of our feedback system. We see that, for good performance, we want large loop gain (so that $|S|$ is small), and at high frequencies we generally want small loop gain (so that $|T|$ is small), to avoid too much amplification of sensor noise.

With this in mind, an integrator $L(s) = 1/s$ looks like it would make a good choice for a loop gain: as shown in Figure 4.3, its Bode magnitude plot is large for low frequencies, and small for high frequencies, just as we desire. We can change the “crossover frequency” (where $|L(i\omega)| = 1$) simply by changing the overall gain: if $L(s) = a/s$, then we would have good performance (large $|L(i\omega)|$) for frequencies $\omega \ll a$, and good noise rejection (small $|L(i\omega)|$) for frequencies $\omega \gg a$.

But if $L(s) = 1/s$ is good, then shouldn’t $L(s) = 1/s^2$ be better? Its Bode plot is even larger at low frequencies, and even smaller at high frequencies,

so it should be even better for both low-frequency performance and high-frequency noise rejection. And for that matter, how about $L(s) = 1/s^3$?

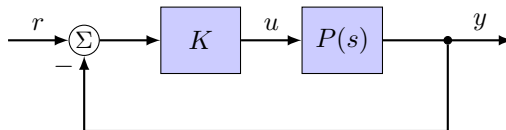
It turns out that $1/s^2$ and $1/s^3$ are *not* good choices for the loop gain, but for a different reason: *stability* of the closed-loop system. It turns out that these would be unstable (or borderline unstable) for all gains. (One can actually see this immediately from a root-locus plot for these systems.)

It would be nice to be able to tell something about closed-loop stability from the Bode plot. This is where the Nyquist stability criterion comes in. In the end, we will see that, for closed-loop stability, *the slope of the magnitude plot of L should be about -1 at crossover* (i.e., where $|L(i\omega)| = 1$). So $1/s^2$ and $1/s^3$ are “too steep,” since their slopes are -2 and -3 , respectively. At this point, it is not at all clear where such a guideline for slope at crossover could possibly come from. Furthermore, it is not the whole story: there are systems for which this condition does not imply closed-loop stability. To understand where this comes from, and to understand the stability criteria more completely, we need the Nyquist stability criterion.

4.5 Nyquist stability criterion

The Nyquist stability criterion provides a way to determine the *closed-loop* stability (e.g., stability of S or T) directly from the *open-loop* Bode plot (e.g., Bode plot of the loop gain L). The method is thus similar in spirit to the root locus method, in which we also obtained information about the closed-loop system (in particular, locations of the closed-loop poles) by looking at the loop gain (in particular, the poles and zeros of L). However, the Nyquist criterion is considerably more powerful than the root locus method, as it lets us determine closed-loop stability directly from a *Bode plot* of L , and therefore lets us look at both dynamics (e.g., stability) and performance (e.g., tracking error) at the same time.

In order to really understand the Nyquist stability criterion, we will need to introduce another type of plot, the *Nyquist plot*. However, we can get a sense for how it works using tools we are already familiar with, such as Bode plots. Let's first illustrate with an example. Consider a typical feedback system for tracking a reference input r , as in the following diagram:



with

$$P(s) = \frac{15}{(s+1)(s+2)(s+3)}. \quad (4.12)$$

The loop gain is thus $L(s) = KP(s)$, and the open-loop poles are at $s = -1, -2, -3$. These are all in the left half plane, so the *open-loop* system

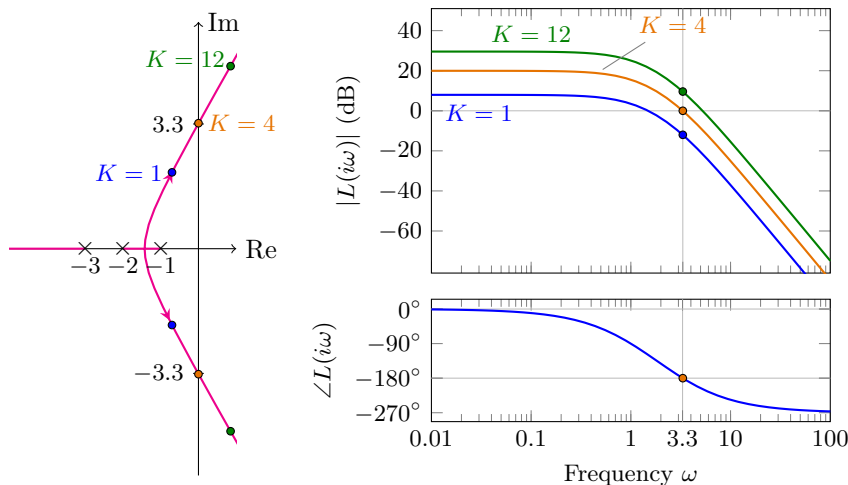


Figure 4.14: Root locus plot (left) and Bode plot (right) for $L(s) = KP(s)$, with P given by (4.12), for $K = 1, 4$, and 12 . The phase plot is the same for all three values of K . In particular, note the magnitude of L where the phase is -180° .

is stable. Let us investigate the stability of the *closed-loop* system as the controller gain K varies. This is something we can easily tell from the root locus plot for $P(s)$, which is shown in Figure 4.14. This figure also shows the closed-loop poles for the specific choices $K = 1, 4, 12$. We see that for $K = 1$, the closed-loop system is stable; for $K = 4$, the closed-loop poles are on the imaginary axis; and for $K = 12$, the closed-loop system is unstable.

Now let us look at the Bode plot for $L = KP$. Figure 4.14 also shows the Bode plots for the same values of K . As the gain K increases, the Bode magnitude simply shifts up, and the phase plot remains unchanged. Note in particular what happens at $\omega = 3.3$, where the phase is -180° : for $K = 1$, the magnitude of L is less than 1 (0 dB); for $K = 4$, the magnitude is exactly 1, and for $K = 12$, the magnitude is larger than 1. As we see from the root locus plot, for $K = 4$ there is a pair of closed-loop poles right on the imaginary axis, in particular at $s = \pm i\omega_0$ for $\omega_0 = 3.3$. Therefore, the closed-loop characteristic equation $1 + KP(s) = 0$ must have a solution $s = i\omega_0$ for this value of K . But this means that $L(i\omega_0) = KP(i\omega_0) = -1$, which has magnitude 1 and phase -180° . So this is why, right at the value of K for which the system transitions from stable to unstable, the magnitude at the frequency for which $L(i\omega) = -180$ is exactly 1.

This gives us a basic method for determining closed-loop stability from a Bode plot of the loop transfer function L : look at the magnitude at the point where the phase crosses -180° . If this magnitude is less than 1 (0 dB), then the closed-loop system is stable; if it is greater than 1, then the closed-loop system is unstable. This method works for many systems, but not all.

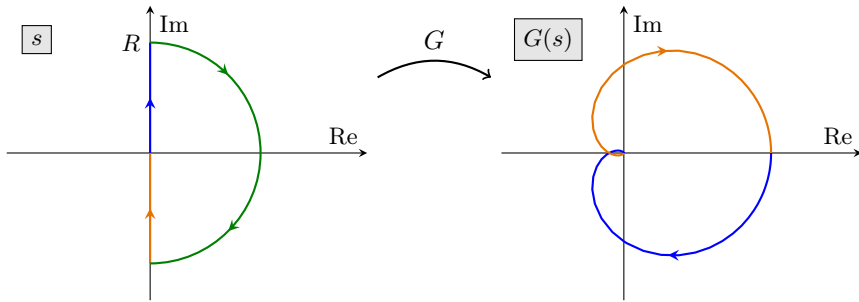


Figure 4.15: The D-shaped contour on the axes on the left is called the *Nyquist contour*, and one usually considers the limit $R \rightarrow \infty$. Points s on this contour are mapped to corresponding points $G(s)$ shown on the plot to the right; the resulting plot is called the *Nyquist plot* of G . Note that the large green semicircle on the left maps to the origin on the right.

(Again, to fully understand it, we will need the Nyquist criterion.) But it does work as long as the open-loop system is stable, and as long as there is only one frequency at which the phase is -180° .

An intuitive way to understand this condition is as follows: when the phase of the loop gain is -180° , the signal that is fed back is *in phase* with the original signal. (One might expect an in-phase signal to have 0° phase, but remember that we have negative feedback, with a minus sign going into the sum in the diagram above.) If this amplification around the loop is > 1 , then the amplitude will increase each time around the loop, leading to instability; but if the amplification is < 1 , then the amplitude will decrease, and the closed-loop system will be stable. This intuitive explanation leaves out some of the subtleties that are handled by the more general Nyquist criterion (e.g., if L is unstable, or if there are multiple crossings of -180°), but it may give an intuitive feel for why the Nyquist criterion makes sense.

Nyquist plot

The Nyquist criterion uses another way of plotting the frequency response, called the *Nyquist plot*. The Nyquist plot is defined as shown in Figure 4.15. One starts with a D-shaped contour in the complex plane, called the *Nyquist contour*, as shown in the figure: the Nyquist contour includes the imaginary axis from $-iR$ to iR , where R is some large number (ultimately, we will let $R \rightarrow \infty$), as well as a semicircle of radius R . To draw the Nyquist plot of a transfer function $G(s)$, one maps each point s on the Nyquist contour to a corresponding point $G(s)$. The result is another closed contour, called the *Nyquist plot* of G .

In Figure 4.15, the colored portions of the Nyquist contour on the left

map to the corresponding colored portions of the Nyquist plot on the right. Note that the large semicircle of radius R maps to the origin on the right: this is true for all transfer functions that are strictly proper (more poles than zeros), since $|G(s)| \rightarrow 0$ as $s \rightarrow \infty$. Note also that the orange portion of the curve (corresponding to the negative imaginary axis on the left) is simply the mirror image of the blue portion of the curve (corresponding to the positive imaginary axis). This is because $G(-i\omega) = \overline{G(i\omega)}$, since the coefficients in the transfer function G are real. In practice, one often plots only one of these halves (typically the portion for $\omega > 0$).

The Nyquist plot may sound complicated, but the end result is actually just another way of plotting the frequency response. In fact, all of the information needed to plot the Nyquist plot is already contained in the Bode plot. Recall that the Bode magnitude plot shows $|G(i\omega)|$ for $\omega > 0$, and the Bode phase plot shows $\angle G(i\omega)$. But the blue portion of the D-shaped Nyquist contour in Figure 4.15 consists of points $s = i\omega$, as ω goes from $0 \rightarrow \infty$. So the Bode plots give the magnitude (radius) and phase (angle) of the blue branch of the Nyquist plot. (This is also often the only portion that is plotted, as mentioned above.)

To illustrate this, consider again the transfer function $P(s)$ from (4.12). The Bode plot and corresponding Nyquist plot are shown in Figure 4.16. We will start at $s = 0$ on the Nyquist contour, the beginning of the blue curve. This corresponds to the low-frequency limit on the Bode plot, labeled by point A in the Bode and Nyquist plots. As we move vertically on the Nyquist contour, the frequency ω increases. At point B , the phase is -90° (from the Bode plot), and the magnitude is a little less than at point A . So the corresponding point on the Nyquist plot intersects the negative imaginary axis (phase $= -90^\circ$) at a radius a little smaller than point A . At point C , the magnitude of $G(i\omega)$ is 1, so the Nyquist plot intersects the unit circle here, with phase somewhere between -90 and -180 . At point D , the phase reaches -180° , so the Nyquist plot intersects the negative real axis. At higher frequencies, the magnitude $|G(i\omega)| \rightarrow 0$, so the Nyquist plot goes to the origin.

Now that we know what a Nyquist plot is, we are now ready for the actual Nyquist criterion. We begin with a simplified version:

Nyquist criterion (simplified): If all the poles of the loop transfer function $L(s)$ are in the left half plane, then the closed-loop system is stable if and only if the Nyquist plot of L does not encircle the -1 point.

Note that, for the Nyquist criterion, we always draw the Nyquist plot of the *loop gain* L ; this is an *open-loop* transfer function, but the Nyquist criterion lets us conclude something about the stability of the *closed-loop* system.

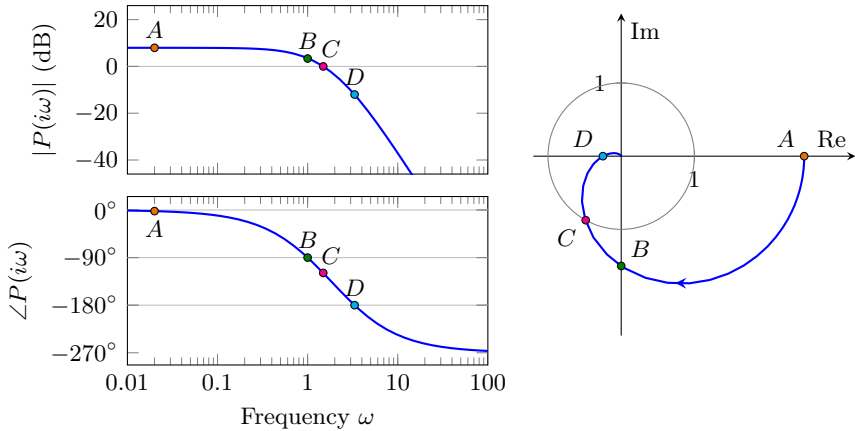


Figure 4.16: Bode and Nyquist plots of $P(s)$, from equation (4.12). (Only the positive-frequency half of the Nyquist plot is shown; this is the blue portion of Figure 4.15.) As the frequency ω increases, the points A, B, C and D on the Bode plot (left) map to the corresponding points on the Nyquist plot (right).

To see how the Nyquist criterion works, let us to return to the example we considered previously, for the plant $P(s)$ in (4.12), with four different gains $K = 1, 4, 12$. The corresponding Nyquist plots of $L(s) = KP(s)$ are shown in Figure 4.17.

First, note that $L(s)$ is stable: all its poles are in the left half plane, so we can apply the simplified Nyquist criterion above. For $K = 1$, the Nyquist plot of $L(s)$ does not encircle the -1 point, so the closed-loop system is stable. This agrees with what we observed in the root locus diagram shown in Figure 4.14.

Next, consider the case for $K = 4$. Note how the Nyquist plot changes: the gain has increased, but the phase remains the same, so the radius of each point on the Nyquist plot increases by a factor of 4. Thus, changing the overall gain corresponds to either “enlarging” or “shrinking” the Nyquist plot. Here, note that for $K = 4$, the Nyquist plot *intersects* the -1 point: this corresponds to one or more poles (here, two) exactly on the imaginary axis.

Finally, for $K = 12$, the Nyquist plot grows still further, and it now does encircle the -1 point (twice, actually). The Nyquist criterion then implies that the closed-loop system is unstable. This again agrees with the root locus plot in Figure 4.14.

We can use this (simplified) Nyquist criterion to deduce the closed-loop stability directly from the Bode plot of the loop gain $L(s)$. Suppose the loop gain L is stable, and suppose the magnitude plot of L crosses 0 dB only once,

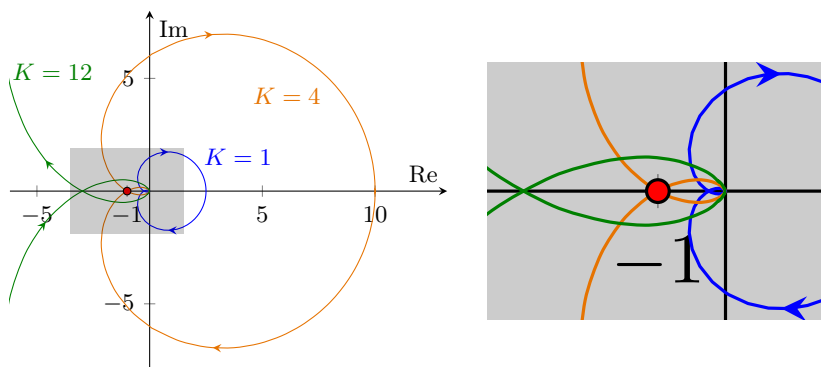


Figure 4.17: Nyquist plot for $L(s) = KP(s)$, with P given in (4.12), and $K = 1, 4, 12$. The region around the -1 point is enlarged at right. For $K = 1$, the curve does not encircle the -1 point; for $K = 4$, the curve intersects the -1 point exactly; and for $K = 12$, the curve encircles the -1 point (twice).

and the phase plot of L crosses -180° only once, with $L(0) > 0$.

Consider the phase at the frequency ω_c where $|L|$ crosses 0 dB:

- if $\angle L(i\omega_c) > -180^\circ$, the closed-loop is *stable*;
- if $\angle L(i\omega_c) < -180^\circ$, the closed-loop is *unstable*.

Alternatively, consider the gain at the frequency ω_{pc} where $\angle L = -180^\circ$:

- if $|L(i\omega_{pc})| < 1$, the closed-loop is *stable*;
- if $|L(i\omega_{pc})| > 1$, the closed-loop is *unstable*.

This then agrees with the intuitive picture we began with, except now we have a rigorous theory to back it up. (We have not yet proved the Nyquist criterion, of course; see below if you are interested in a proof.)

The simplified criterion above is limited to loop transfer functions that are stable. The full version applies to any loop gain $L(s)$, stable or unstable, and is stated as follows:

Nyquist stability criterion: Let P be the number of *open-loop* RHP poles (poles of the loop gain $L(s)$), and let N be the number of clockwise encirclements the Nyquist plot of L makes of the -1 point. Then the number of *closed-loop* RHP poles is

$$Z = P + N.$$

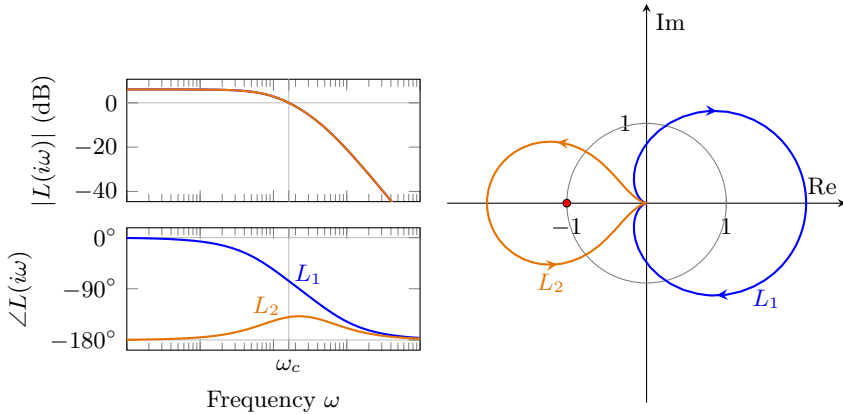


Figure 4.18: Bode and Nyquist plots for a stable loop gain L_1 and an unstable loop gain L_2 , from equation (4.13). The Bode magnitude plots for the two systems are identical; only the phase plots differ. For both systems, the closed-loop system is stable: since L_2 has one RHP pole, the Nyquist plot needs 1 counter-clockwise encirclement of -1 for the closed-loop system to be stable.

A counter-clockwise encirclement counts as -1 clockwise encirclements; thus, for closed-loop stability ($Z = 0$), one requires P *counter-clockwise* encirclements of the -1 point. The Nyquist criterion is an application of a useful result from complex analysis known as the argument principle, and its proof is given in Appendix B.3.

Our main interest is to know whether the closed-loop system is stable or unstable; the Nyquist criterion tells us in addition how many unstable closed-loop poles we have. The following example illustrates the Nyquist criterion for an unstable open-loop system.

Example 4.18. Consider the two different loop transfer functions

$$L_1(s) = \frac{10}{(s+1)(s+5)}, \quad L_2(s) = \frac{10}{(s-1)(s+5)}. \quad (4.13)$$

Note that L_1 is stable, with poles at $s = -1, -5$, while L_2 is unstable, with a RHP pole at $s = +1$. Let us use the Nyquist criterion to determine whether the corresponding closed-loop systems are stable.

The Bode and Nyquist plots for both of these loop transfer functions are shown in Figure 4.18. The Bode magnitude plots are identical, but the phase plots differ, so the Nyquist plots are quite different. Since L_1 is stable, we can use the simplified Nyquist criterion for L_1 . We see that the the Nyquist plot of L_1 does not encircle the -1 point: the corresponding closed-loop system is therefore stable.

For L_2 , however, we must use the full version of the Nyquist criterion. Now, the number of open-loop RHP poles is $P = 1$. The Nyquist plot of L_2 does encircle the -1 point, but in a counter-clockwise direction, so the number of *clockwise* encirclements is $N = -1$. Thus, the number of closed-loop RHP poles is $Z = P + N = 1 - 1 = 0$. The closed-loop system corresponding to L_2 is thus also stable. \diamond

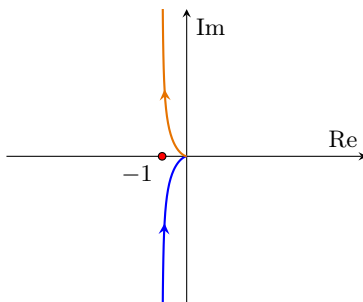
Poles on the imaginary axis

An important special case arises when the loop transfer function has poles that are exactly on the imaginary axis. This happens, for instance, when integral feedback is used: here, the controller includes a term proportional to $1/s$, so the loop transfer function has a pole at $s = 0$. The Nyquist contour shown in Figure 4.15 goes through $s = 0$, but if $G(s)$ has a pole at $s = 0$, then $G(0)$ is undefined, so how are we to draw the Nyquist plot?

For example, consider the loop transfer function

$$L(s) = \frac{1}{s(s+1)}, \quad (4.14)$$

which has a pole at $s = 0$. If we naïvely use MATLAB to generate a Nyquist plot of L , it will produce something like the following:



Let us look at what is happening here: starting at $\omega = -\infty$, the Nyquist plot of $L(i\omega)$ starts at the origin, and we proceed along the orange curve in the direction of the arrow as ω increases. As ω approaches zero from below, $|L(i\omega)|$ goes to infinity, with a phase of 90° , and is undefined for $\omega = 0$. Then, for small positive ω , we all of a sudden jump to the blue curve with very large magnitude and phase -90° , and again approach the origin as $\omega \rightarrow +\infty$.

So the Nyquist plot shoots off to infinity for small frequencies $\omega \rightarrow 0$. This might be okay, but for stability, we need to be able to tell whether the Nyquist plot encircles the -1 point. The plot above is not even a closed curve, so we cannot possibly answer this question! What are we to do?

In this case, the solution is to modify the Nyquist contour so that it “goes around” any poles on the imaginary axis. To draw the Nyquist plot for L , we use a modified Nyquist contour shown in Figure 4.19: in particular, we indent

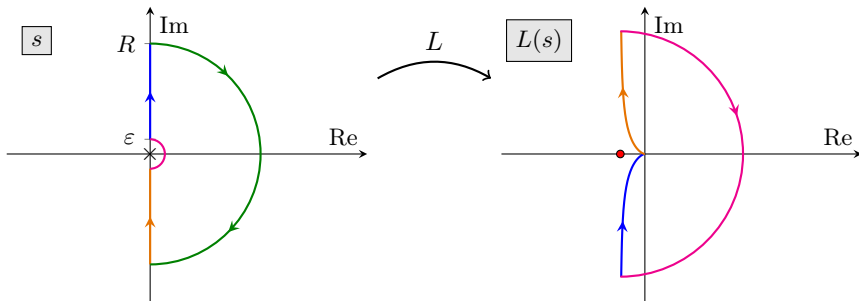


Figure 4.19: Modified Nyquist contour, indented to avoid a pole on the imaginary axis. The corresponding Nyquist plot for $L(s)$ given by (4.14) is shown at right.

the contour around the pole at $s = 0$, with a small semicircle of radius ϵ , and consider the limit $\epsilon \rightarrow 0$. The corresponding Nyquist plot is shown at right in Figure 4.19. This small semicircle of radius ϵ maps to a large circle of radius about $1/\epsilon$ (since $L(s) \approx 1/s$ for $|s| \ll 1$). Importantly, while s moves *counter-clockwise* around the small semicircle, $G(s)$ moves *clockwise* around the large circle: for $s = \epsilon e^{i\theta}$ with θ varying from $-\pi/2$ to $\pi/2$, we have

$$G(s) \approx \frac{1}{\epsilon e^{i\theta}} = \frac{1}{\epsilon} e^{-i\theta},$$

so its phase varies from $\pi/2$ to $-\pi/2$. Thus, the picture is as indicated in the figure: the Nyquist plot does not encircle the -1 point, and the closed-loop system is stable.

This case of a single pole at $s = 0$ is quite common, since it occurs whenever integral feedback is used, so it is worth remembering that in this case, the Nyquist plot is “closed” in a clockwise fashion, including the right half plane, not in a counterclockwise fashion including the left half plane. With this in mind, one can correctly interpret Nyquist plots (e.g., those produced by MATLAB) without the need to explicitly use an indented contour.

4.6 Stability margins

The Nyquist stability criterion answers a yes-or-no question: it tells us whether or not the closed-loop system is stable. Often, it is useful to know *how close* a system is to instability. Some systems may be quite sensitive: if we change our control gains only slightly, the closed-loop system may become unstable; other systems might be quite robust to such changes. Stability margins give us a sense for how close the closed-loop system is to instability. The idea is that, if the Nyquist plot comes close to the -1 point, then small perturbations of the loop transfer function can make the closed-loop system unstable.

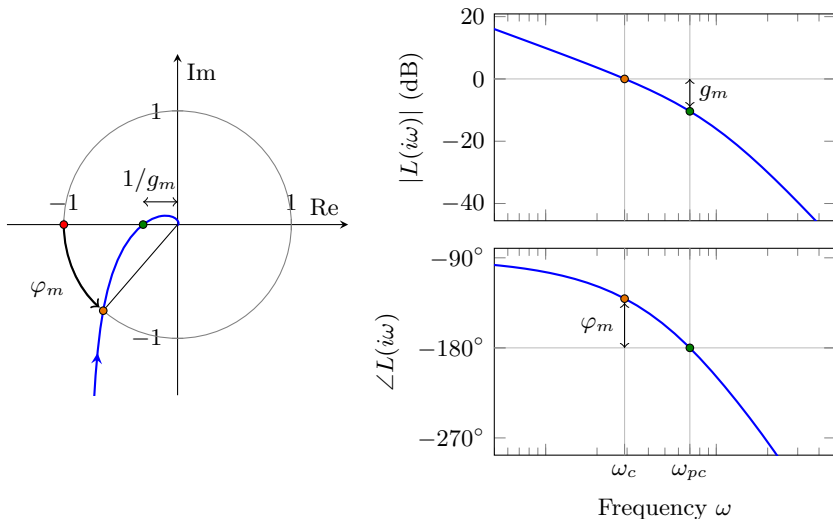


Figure 4.20: Gain margin g_m and phase margin φ_m illustrated on a Nyquist plot (left) and on a Bode plot (right).

There are different types of stability margins. The first is the *gain margin* g_m , which is the amount by which we can increase (or decrease) the gain before the closed-loop system goes unstable. It is defined as

$$g_m = \frac{1}{|L(i\omega_{pc})|},$$

where ω_{pc} is the *phase crossover frequency*, the frequency at which the phase $\angle L(i\omega)$ crosses -180° . Figure 4.20 shows a typical Nyquist plot and Bode plot, with the gain margin indicated. Note that, on the Nyquist plot, the gain margin is determined by the point at which the Nyquist plot intersects the negative real axis; if the intersection is close to the -1 point, the gain margin will be close to 1; if the plot never intersects the negative real axis, the gain margin is infinite. To determine the gain margin from the Bode plot, one simply looks at the magnitude at the point where the phase is -180° : if the magnitude is -6 dB, then the gain margin is 6 dB, or about a factor of 2, meaning that the gain can be increased by a factor of 2 before the closed-loop system goes unstable.

The *phase margin* is defined by

$$\varphi_m = 180^\circ + \angle L(i\omega_c),$$

where ω_c is the *crossover frequency*, at which the magnitude $|L(i\omega)|$ crosses 0 dB. The phase margin is also indicated in Figure 4.20. On the Nyquist plot, the phase margin is determined by the point at which the curve intersects

the unit circle (magnitude 1). The angle between this intersection point and the -1 point is the phase margin. On the Bode plot, the phase margin is determined by the phase at the crossover frequency. If the phase at crossover is -120° , then the phase margin is 60° .

The phase margin is particularly useful, as it can often be related to the effective damping ratio of the closed-loop system:

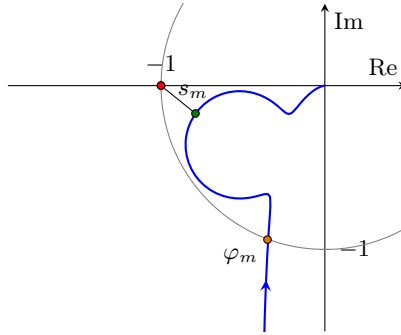
$$\zeta \approx \frac{\varphi_m}{100}, \quad (4.15)$$

where φ_m is given in degrees. The above is only a guideline, not an exact relation, but it is quite handy. For instance, one sees that if the phase margin is $\varphi_m \approx 30^\circ$, then the closed-loop system will be quite oscillatory, since $\zeta = 0.3$ is a relatively small damping ratio (see Figure 3.2). For this reason, one typically wants to design a feedback system so the phase margin is at least 50° .

Gain and phase margins are the most commonly used (classical) stability margins. However, they can be misleading: for instance, consider the loop transfer function

$$L(s) = \frac{0.39(s^2 + 0.1s + 0.55)}{s(s+1)(s^2 + 0.06s + 0.5)}$$

(considered in [1, Example 9.8]), whose Nyquist plot is shown below:



This system has an infinite gain margin, and a healthy phase margin of 69° . However, we see that the Nyquist plot comes quite close to the -1 point. Systems like this motivate an additional notion of stability margin, which is simply the minimum distance from the Nyquist plot to the -1 point, denoted s_m in the figure. We call this quantity the *stability margin* (it is sometimes called the *vector margin*), and it is defined by

$$s_m = \min_{\omega \in \mathbb{R}} |1 + L(i\omega)|. \quad (4.16)$$

The stability margin s_m is usually a better single indicator of how close a system is to instability than either the gain or phase margin, and it is easy to see s_m from the Nyquist plot. However, one cannot immediately identify s_m from a Bode plot of the loop transfer function. For this reason, it is usually best to use a Nyquist plot to check stability.

4.7 Loopshaping

The ability to determine the stability of the closed-loop system from the loop transfer function enables a powerful technique for designing feedback systems: *loopshaping*. We have seen that all of the main characteristics of a feedback system (e.g., performance, closed-loop stability) are determined by the loop transfer function L . The idea of loopshaping is as follows: given a plant P that we wish to control, we choose a *desired* loop transfer function L , and then find a controller C such that $L = PC$.

The main question is then how to choose L . Recall from Section 4.4 that a good measure of the performance is the sensitivity function $S = 1/(1 + L)$, and for good performance we want $|S|$ small. Recall further that, for $|L| \gg 1$, we have $|S| \approx 1/|L|$, so for frequencies at which we want good performance (typically low frequencies), we would like *large gain*: $|L(i\omega)| \gg 1$.

Conversely, for frequencies at which we do not want our controller to do much (typically high frequencies), we would like $|T|$ small, to avoid too much amplification of sensor noise. Since $|T| \approx |L|$ when $|L| \ll 1$, this means that for these frequencies we would like *small gain*: $|L(i\omega)| \ll 1$.

In addition, we need to worry about stability of the closed-loop system. From the Nyquist criterion, we know that the phase at the crossover frequency ω_c needs to be more than -180° , and to have a reasonable phase margin (say 50°), we need the phase to be more than about -130° . From the Bode gain-phase relation (Section 4.3, page 103), we know that, at least if L is minimum phase (which is the best-case scenario), then the slope at crossover needs to be about -1 . A slope of exactly -1 would have a phase of -90° , for a healthy phase margin of $\varphi_m = 90^\circ$. So in principle we could tolerate a slightly steeper slope, but a slope as large as -2 is clearly out of the question, as then the phase at crossover would be -180° , and the phase margin would be 0, on the verge of instability. Usually the loop gain becomes steeper after the crossover frequency (either because of additional lags in the natural dynamics, or because of poles we intentionally include in our controller), and recall that the phase is determined by averaging the slope over a range of frequencies, so in practice a good guideline is for a slope of -1 at crossover.

These ideas may be summarized as follows.

The loop transfer function should usually have a magnitude as shown in Figure 4.21:

- High gain at low frequencies (for good performance)
- Low gain at high frequencies (to avoid noise amplification)
- Slope about -1 around the crossover frequency (for stability)

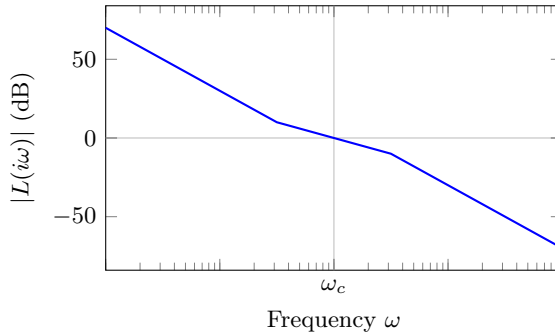


Figure 4.21: Typical desired magnitude of loop transfer function: good performance (high gain) at low frequencies; low gain at high frequencies; slope about -1 where magnitude crosses 1.

The specifics of how high the gain should be at low frequencies, what the crossover frequency should be, etc, will depend on the particular control objective (for instance, how much tracking error can be tolerated over different frequency ranges). Also, keep in mind that the above is for the common case in which one desires control to be effective at low frequencies; there are certainly applications in which one does not care about low frequencies, and wants the control to be effective at other frequencies (for instance, in a disturbance rejection problem, in which we want to reject disturbances in a particular range of frequencies). So more generally, we want $|S|$ to be small ($|L|$ large) at frequencies for which we want the control to be effective.

Sensitivity function from the Nyquist plot

We have seen in Section 4.6 that the Nyquist plot of the loop transfer function L gives a lot of information about the stability of a closed-loop system. It not only answers the question of *whether* the closed-loop system will be stable, but also *how close* it is to instability. But in order to understand the performance of the closed-loop system, one really wants to look at the sensitivity function $S = 1/(1 + L)$ (see §2.6 and §4.4). Can we learn anything about S (a closed-loop transfer function) from the Nyquist plot of L (an open-loop transfer function)?

The answer is yes. Consider a particular frequency ω , and the corresponding point $L(i\omega)$ on the Nyquist plot. As shown in Figure 4.22, the complex number $1 + L(i\omega)$ can then be viewed as a vector from the -1 point to the point $L(i\omega)$ on the Nyquist plot. Thus, the distance from the -1 point to the point $L(i\omega)$ is $|1 + L(i\omega)| = 1/|S(i\omega)|$. To summarize:

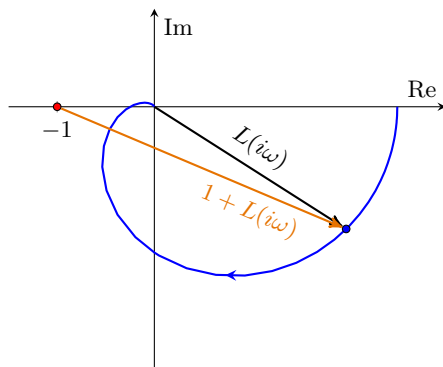


Figure 4.22: The distance from the -1 point to a point $L(i\omega)$ on the Nyquist plot is $|1 + L(i\omega)| = 1/|S(i\omega)|$.

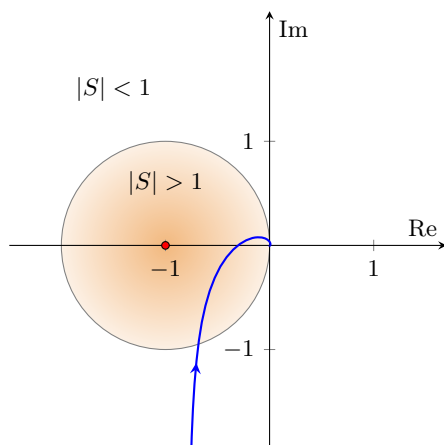


Figure 4.23: For frequencies at which the Nyquist plot $L(i\omega)$ is outside the unit circle centered about the -1 point, $|S(i\omega)| < 1$, so feedback is beneficial; at frequencies for which the Nyquist plot is inside this circle, $|S(i\omega)| > 1$, and feedback is detrimental.

At any frequency ω , the distance from the -1 point to the point $L(i\omega)$ on the Nyquist plot is $1/|S(i\omega)|$.

This simple observation lets us learn a lot from the Nyquist plot. First, for good performance, we want $|S|$ to be small, so we want the Nyquist plot of L to be far from the -1 point. Second, we know that large values of $|S|$ will mean poor stability margins: if $|S|$ is large, then $|1 + L|$ is small, so the Nyquist plot comes close to the -1 point. In fact, recalling the definition of the stability margin s_m from Section 4.6, we have

$$\max_{\omega} |S(i\omega)| = \frac{1}{s_m}. \quad (4.17)$$

Thus, a large maximum value of $|S|$ implies a small stability margin, so any small perturbations or errors in the plant model may result in an unstable closed-loop system.

Furthermore, we can distinguish between frequencies for which feedback is *beneficial* and frequencies for which feedback is *detrimental*. If $|S(i\omega)| < 1$, then feedback helps us: it makes the system less sensitive to disturbances, less sensitive to changes in the plant, and reduces the tracking error (as explained in Section 2.6). From the above discussion, we see that $|S(i\omega)| < 1$ as long as the distance from the -1 point to the Nyquist plot is greater than 1. The situation is depicted in Figure 4.23. At frequencies for which the Nyquist plot lies outside the unit circle centered at -1 , we have $|S| < 1$, so feedback is beneficial; at frequencies for which the Nyquist plot lies inside this circle, $|S| > 1$ and feedback is detrimental: it *exacerbates* the effect of external disturbances, the closed-loop system is more sensitive to changes in the plant, etc. The Nyquist plot of L therefore gives a great deal of useful information about the closed-loop system. Again, the reason it is helpful to look at L rather than look at S itself is that we have direct control over L (through our choice of controller), while changes in our controller modify S in complex ways that are difficult to predict.

Bandwidth

The bandwidth of a feedback system is the range of frequencies $\omega \in [\omega_1, \omega_2]$ over which control is “effective.” Usually, we take $\omega_1 = 0$ (so that control is effective at low frequencies) and we denote the bandwidth by a single frequency $\omega_B = \omega_2$.

How precisely do we characterize when feedback is “effective?” A convenient measure of the effectiveness of a feedback system, once again, is the sensitivity function $S = 1/(1 + L)$. When $|S|$ is small, the feedback system is effective, in a number of different ways: it reduces the effects of disturbances, achieves small tracking error, and makes the system less sensitive to changes in the plant (see §2.6). However, for high frequencies, we know $|L| \rightarrow 0$, so $|S| \rightarrow 1$ as $\omega \rightarrow \infty$. We thus adopt the following more precise definition (from [22]):

The *bandwidth* ω_B of a feedback system is the frequency where $|S(i\omega)|$ first crosses $1/\sqrt{2} \approx 0.707$ (≈ -3 dB) from below.

This choice of -3 dB is somewhat arbitrary, but it is conventional, and makes sense for a number of reasons, as we will soon show.

Another commonly used definition of the bandwidth is in terms of the complementary sensitivity function $T = L/(1 + L)$. Recall that, for the frequencies at which control is effective, we have $T \approx 1$. But for high frequencies, $|T| \approx |L| \rightarrow 0$. Thus, we can define the bandwidth based on T as the frequency at which $|T(i\omega)|$ first crosses $1/\sqrt{2} \approx -3$ dB from above. We will denote this frequency by ω_{BT} .

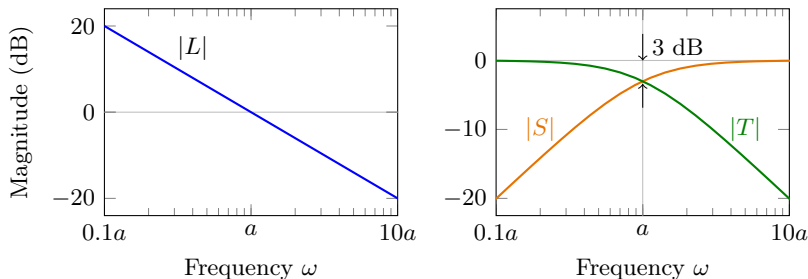


Figure 4.24: Loop gain (left) and closed-loop transfer functions (right) for $L(s) = a/s$. For this system, $\omega_B = \omega_{BT} = \omega_c = a$.

There is one more important frequency we will want to consider: recall that the *crossover frequency* ω_c is the frequency where $|L(i\omega)|$ first crosses 1 from above.

To understand where the seemingly arbitrary value of $1/\sqrt{2}$ comes from, consider as an example the loop gain for an integrator, $L(s) = a/s$. For this particular L , we have

$$S = \frac{1}{1+L} = \frac{s}{s+a}$$

$$T = \frac{L}{1+L} = \frac{a}{s+a}.$$

The crossover frequency is $\omega_c = a$ (since $L(ia) = a/(ia)$, which has magnitude 1). Furthermore, for $\omega = a$, we have $|T(i\omega)| = |S(i\omega)| = 1/\sqrt{2}$, or a magnitude of about -3 dB. The Bode plots of these three transfer functions are shown in Figure 4.24. Note that, for low frequencies, $|S|$ is small, so feedback is effective for frequencies up to about $\omega_B = a$, at which point $|S|$ first exceeds $1/\sqrt{2} \approx -3$ dB. Alternatively, in terms of $|T|$, we see $T \approx 1$ for low frequencies, indicating good tracking for frequencies up to about $\omega_{BT} = a$, at which point $|T|$ first drops below -3 dB. Thus, for the integrator, $\omega_B = \omega_{BT} = \omega_c$.

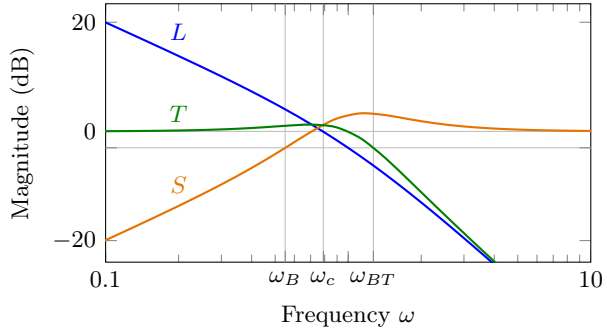
For most cases, however, these three frequencies are not the same. For a more typical case, consider

$$L(s) = \frac{1}{s(s+1)}. \quad (4.18)$$

A Bode magnitude plot of L is shown in Figure 4.25, along with the magnitude plots of $|S|$ and $|T|$. The bandwidth is $\omega_B = 0.55$ radians/sec, the crossover frequency is $\omega_c = 0.79$, and $\omega_{BT} = 1.27$. Note that $\omega_B < \omega_c < \omega_{BT}$ (see Exercise 4.14). Note also that $|S|$ and $|T|$ are both > 1 (0 dB) in the vicinity of the crossover frequency.

Which of these notions of bandwidth is the best way to characterize the range of frequencies over which feedback is effective? For a feedback system,

Figure 4.25: Magnitude of loop transfer function L , and closed-loop transfer functions S and T , for L given by (4.18).



ω_B is the best choice. The main reason for this is that if $|S|$ is small (e.g., for $\omega < \omega_B$), this guarantees that $T \approx 1$, since $S + T = 1$. However, if $|T| \approx 1$ (e.g. for $\omega < \omega_{BT}$), this does *not* guarantee that $|S|$ is small, because the phase of T is also important: for instance, if $T = -1$, then $S = 2$, so the closed-loop sensitivity is worse than open-loop (since $|S| > 1$). Thus, even though ω_{BT} is a common definition of bandwidth (for instance, it is what is calculated by the MATLAB command `bandwidth(T)`), it is often a poor indicator of performance, as the following example illustrates.

Example 4.19. Let us compare the following two loop transfer functions:

$$L_1(s) = \frac{10(1-s)}{s(s+21)}, \quad L_2(s) = \frac{10}{s}.$$

Computing the complementary sensitivity functions for both systems, we find

$$T_1(s) = \frac{10}{s+10} \frac{1-s}{1+s}, \quad T_2(s) = \frac{10}{s+10}.$$

Note that the Bode magnitude plot of $(1-s)/(1+s)$ is 1 everywhere (it is an *all-pass transfer function*), so the magnitude plots of T_1 and T_2 are identical. These are plotted in Figure 4.26, and we see the bandwidth based on T is $\omega_{BT} = 10$ for both systems. Note, however, that the *phase* of T_1 is a problem: even though $|T_1| \approx 1$ up to about $\omega = 10$, the phase of T_1 is far from 0° (it is in fact closer to -180°), so the closed-loop system will *not* reject disturbances or accurately track signals in this frequency range. This poor performance is evident from the Bode magnitude plots of S : we see that the bandwidth based on S_1 is $\omega_{B1} = 0.36$, much lower than ω_{BT} , while for S_2 we have $\omega_{B2} = 10$, indicating good performance over a much greater range of frequencies.

The loop gains L_1 and L_2 are shown in Figure 4.27. Note that, even though $\omega_{BT} = 10$ for both systems, the loop gain L_1 is still small for frequencies $\omega > 0.5$, so we would not expect good performance for frequencies in the range $\omega \in [0.5, 10]$, even though these are below ω_{BT} . Again, ω_B is a much better indicator of performance. The closed-loop step responses are

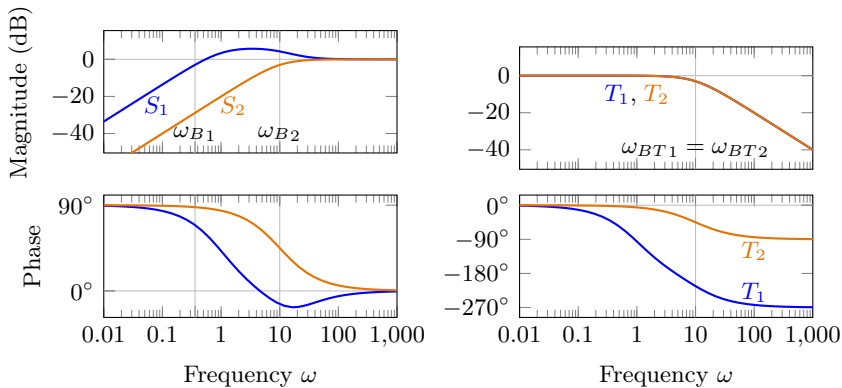


Figure 4.26: Bode plots for closed-loop transfer functions S and T , for Example 4.19. Note that ω_{BT} is the same for both systems, but ω_B is much higher for L_2 than for L_1 .

also shown in Figure 4.27: clearly, T_2 has a much faster response than T_1 . It should be clear from this example that ω_{BT} is not a reliable indicator of closed-loop performance for this system, while ω_B is. \diamond

Bode's integral formula

Let us take another look at the example above, for the loop transfer function $L(s) = a/s$. The Bode plot of the corresponding sensitivity function S was shown in Figure 4.24, and has a particularly nice feature: $|S(i\omega)| \leq 1$ for all ω . This means that, for this system, feedback is always beneficial: recall that $|S| < 1$ means feedback is beneficial (for instance, in terms of disturbance rejection or sensitivity to changes in the plant), while $|S| > 1$ means that feedback is detrimental, and the closed-loop system performs worse than the open-loop system.

By contrast, for the example shown in Figure 4.25, there are frequencies (around crossover) for which $|S(i\omega)| > 1$, so the closed-loop system performs worse than the open-loop system at these frequencies. We would of course prefer to design a controller so that this does not happen: we would like $|S(i\omega)| < 1$ for all frequencies, as in Figure 4.24. Can we always design such a controller?

Unfortunately, the answer is no: for virtually all systems (and all “real” physical systems), we will have $|S(i\omega)| > 1$ for some frequencies. This fact is a consequence of the following theorem:

Theorem 4.20 (Bode's integral formula). *Assume that the loop transfer function $L(s)$ goes to zero faster than $1/s$ as $s \rightarrow \infty$, and let p_k denote the*

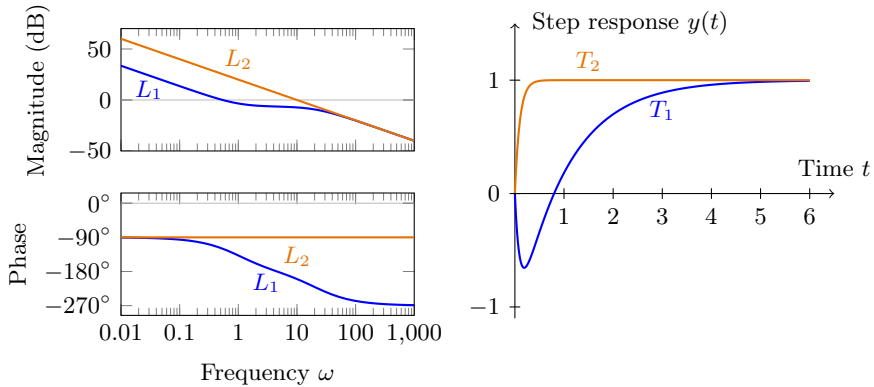


Figure 4.27: Bode plot of loop gain (left) and closed-loop step response (right) for the two systems in Example 4.19.

poles of L in the right half plane. Then the sensitivity function $S = 1/(1 + L)$ satisfies

$$\int_0^\infty \log |S(i\omega)| d\omega = \pi \sum_k p_k. \quad (4.19)$$

A more precise statement of the theorem is given, along with a proof, in Appendix B.4 (Theorem B.7). The integral in (4.19) has an interesting interpretation: it is the area under the graph of $|S(i\omega)|$, where $|S|$ is plotted on a log scale, and frequency ω is plotted on a linear scale. In the case that $L(s)$ is stable (no right-half-plane poles), the sum on the right of (4.19) is zero, so the theorem implies that the negative area ($|S(i\omega)| < 1$) must be exactly balanced by a positive area ($|S(i\omega)| > 1$). This property is a type of conservation law, sometimes referred to as the “area rule” or the “waterbed effect.” if one tries to push down the sensitivity $|S(i\omega)|$ for some range of frequencies, then $|S(i\omega)|$ must increase at other frequencies, in such a way that the total area is preserved. In short, there is no free lunch. Figure 4.28 illustrates this effect, for $L(s)$ given by (4.18): the negative area is balanced by an equal positive area. If L has right-half-plane poles, the situation is even worse: the net area must then be positive.

Of course, an obvious way around this restriction is to choose the loop transfer function $L(s)$ such that it does not go to zero faster than $1/s$ as $s \rightarrow \infty$. In other words, L should have relative degree 1 (or less). Unfortunately, all “real” systems have relative degree at least 2: every feedback system will have a sensor and an actuator, and each of these devices is a physical device, whose transfer function will “roll off” to zero at high enough frequency (i.e., each has relative degree at least 1). Since the overall loop gain includes both the sensor dynamics and the actuator dynamics, and the controller must be proper (at least as many poles as zeros), the overall loop transfer function

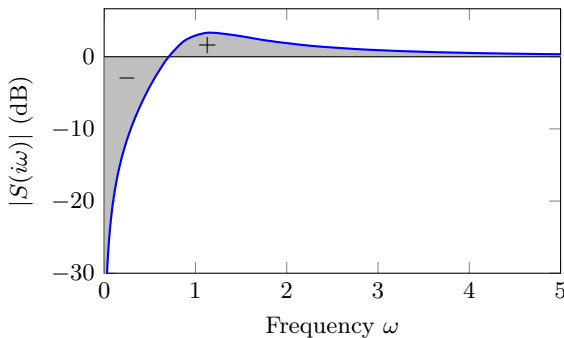


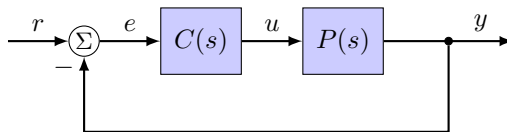
Figure 4.28: Bode's integral formula (the “area rule” or “waterbed effect”) for $L(s)$ given by (4.18). Note that $|S|$ is plotted on a log scale (dB), while frequency is plotted on a linear scale.

must have relative degree at least 2.

At this point, one might worry that feedback might provide little benefit after all: although we can reduce sensitivity over some frequency range, we will inevitably increase sensitivity for other frequencies, making matters worse, so what is the point? The reality is that, at least for minimum-phase systems, the constraints imposed by the area rule are not that restrictive. As indicated in Figure 4.28, one can get very large reduction in sensitivity (large negative values) over a frequency range for which one desires good performance, and then spread the “penalty” of increased sensitivity over a large frequency range, with only a small increase in sensitivity. For systems with right-half-plane poles (and right-half-plane zeros), however, the constraints imposed by the area rule can be quite severe.

4.8 Frequency-domain design

Now that we have a better understanding of how to use Bode and Nyquist plots to design controllers, let us illustrate with some examples. In each of the examples in this section, we consider the problem of tracking a reference r , using a feedback loop as shown below:

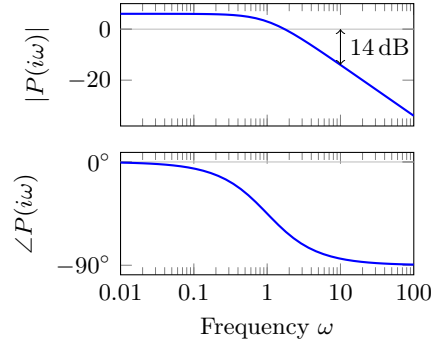


However, the techniques apply equally as well to disturbance rejection problems. In both cases, the main goal is to design a controller such that the loop gain is large over the frequency range over which we would like feedback to be effective.

Example 4.21. Consider the plant

$$P(s) = \frac{2}{s+1}, \quad (4.20)$$

Figure 4.29: Bode plot of the plant $P(s)$ given by (4.20). In order to place the crossover frequency at 10 rad/sec, a proportional controller would need a gain of 14 dB.



and suppose we wish to design a controller so that the bandwidth is around 10 rad/sec, and there is zero steady-state error for tracking a step reference r . A Bode plot of this plant is shown in Figure 4.29.

A good first step is to try proportional control,

$$C_P(s) = k_p.$$

We can determine the value of k_p that gives (at least approximately) the desired bandwidth of 10 rad/sec directly from this Bode plot. From Section 4.7, we know the bandwidth is approximately the same as the crossover frequency, and we can set the crossover frequency to 10 rad/sec by choosing $k_p = 14 \text{ dB} \approx 5$, as shown in Figure 4.29. However, the resulting controller will not meet the specifications. It will have the correct bandwidth, but will not have zero steady-state tracking error: the steady-state error will be given by

$$|S(0)| = \frac{1}{|1 + k_p P(0)|} = \frac{1}{11} \neq 0.$$

One can see this from the Bode magnitude plot of the plant (or of the loop gain, which is the same plot, shifted up by 14 dB), by looking at the magnitude for low frequencies: for zero steady-state error, this magnitude should go to infinity, so $|S| \approx 1/|L| \rightarrow 0$.

A common way to achieve this is to use integral control:

$$C_I = \frac{k_i}{s}.$$

We can determine the value of k_i that achieves a crossover frequency at 10 rad/sec using the same procedure we used for choosing k_p above: we look at the Bode plot of $P(s)/s$, find the magnitude at 10 rad/sec to be -34 dB , and then choose $k_i = 34 \text{ dB} \approx 50$. The corresponding loop gain is shown in Figure 4.30: the crossover is right where we want it at 10 rad/sec, and there is zero steady-state tracking error, since $|L(i\omega)| \rightarrow \infty$ as $\omega \rightarrow 0$, so it appears we are done. However, this is not a good controller, for a different

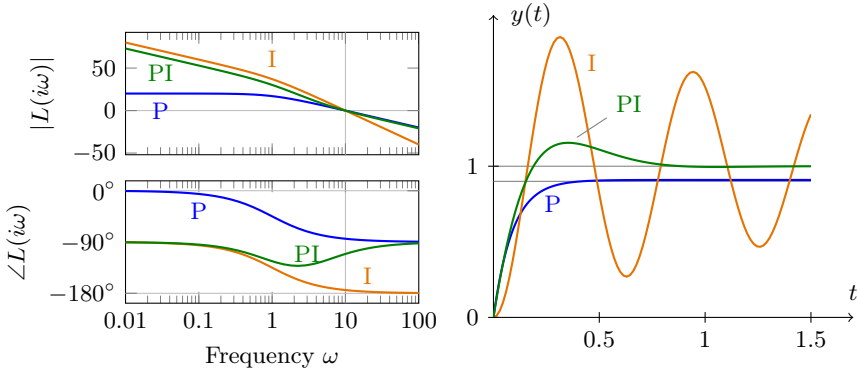


Figure 4.30: Bode plots of loop gain (left), and closed-loop step response (right) for the proportional (P), integral (I), and proportional-integral (PI) controllers from Example 4.21.

reason: the phase margin is too small. From the Bode plot of the loop gain in Figure 4.30, we see the phase at the crossover frequency is just a little larger than -180° : the phase margin is only 6° . So while the closed-loop system will be stable, it will have very poor damping, and will be “close” to unstable: the transient response will have many oscillations. For instance, the damping ratio for the closed-loop system can be approximated by (4.15) as $\zeta \approx 6/100$, which is very lightly damped. This poor transient behavior is reflected in the closed-loop step response shown in Figure 4.30. Eventually, the oscillations will settle down and we will have zero steady-state error, but the long, oscillatory transient is probably not acceptable.

A solution is to use proportional-integral (PI) control:

$$C_{PI}(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s}. \quad (4.21)$$

A Bode plot of this controller is shown in Figure 4.31. Note that the PI controller has a pole at $s = 0$, and also a zero at $s = -k_i/k_p$. The advantage of the zero is that, for frequencies higher than the corner frequency k_i/k_p , the phase increases, so we remove the “phase penalty” caused by the integrator (which reduced the phase by -90° everywhere). In order to benefit from this zero, we should choose this corner frequency k_i/k_p to be a little before the crossover frequency (here 10 rad/sec), so that at least some of the phase increase happens before the crossover frequency, for a corresponding improvement in phase margin. If we choose k_i/k_p too large, we will not get the phase benefit at crossover, which was the whole point of adding the zero in the first place; if we choose k_i/k_p too small, the gain at low frequencies will be smaller, resulting in worse performance; the phase margin will also be too large, resulting in an overdamped system, with slower transients. A

Figure 4.31: Bode plot of integral controller $C_I(s) = k_i/s$ and PI controller $C_{PI}(s) = k_p + k_i/s$. The behavior at low frequencies is identical, but the PI controller recovers the negative phase after the corner frequency k_i/k_p , at the expense of higher gain at high frequency.

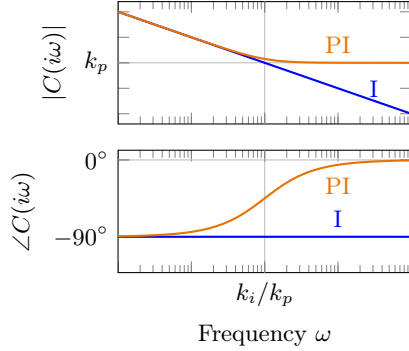
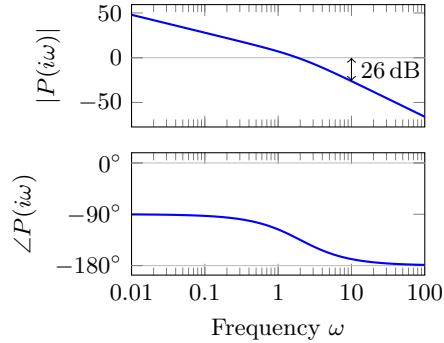


Figure 4.32: Bode plot of the plant $P(s)$ given by (4.22). In order to place the crossover frequency at 10 rad/sec, a proportional controller would need a gain of $k_p = 20$ (26 dB).



reasonable choice is $k_i/k_p = 5$: the corner frequency is then 5 rad/sec, enough before the crossover frequency that we should have a healthy phase margin, but not so low that we sacrifice much performance. The resulting PI controller then looks like $C(s) = k_p(1 + 5/s)$; the proportional gain k_p that achieves a bandwidth of 10 rad/sec can then be determined as before, giving $k_p = 13 \text{ dB} \approx 4.5$. The resulting loop gain and closed-loop step response are shown in Figure 4.30. From the Bode plot, we see that including the zero has given us a little less gain at low frequencies than the PI controller, but not much; and the phase margin is now a healthy 69° , so the undesired oscillations in the closed-loop step response have been removed. \diamond

Example 4.22. Consider a plant given by

$$P(s) = \frac{5}{s(s+2)}, \quad (4.22)$$

with the same requirements as in the previous example: namely, a bandwidth of around 10 rad/sec, and zero steady-state error for tracking a step reference. A Bode plot of this plant is given in Figure 4.32.

As before, we first consider a proportional feedback $C(s) = k_p$. From the Bode plot, we see that the value of k_p that achieves a crossover frequency of

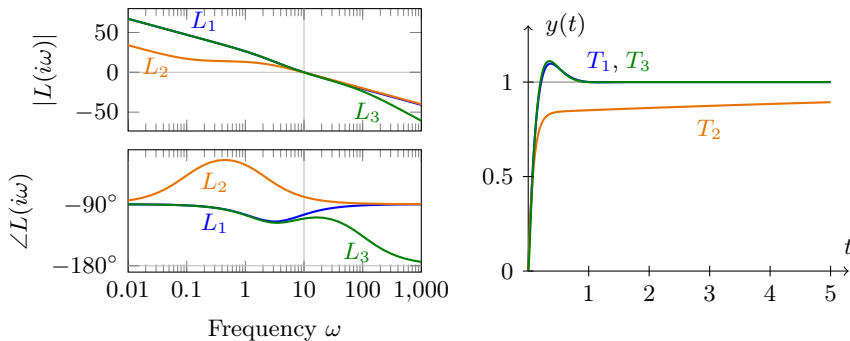


Figure 4.33: Bode plots of loop gain (left), and closed-loop step response (right), for controllers given by (4.23a)–(4.23c).

10 rad/sec is $26 \text{ dB} \approx 20$. For this plant, proportional feedback will always give zero steady-state error (for any value of k_p), since

$$|S(0)| = \lim_{s \rightarrow 0} \frac{1}{|1 + k_p P(s)|} = 0.$$

So it appears we have met the specifications with a proportional controller. However, the same problem occurs as with pure integral feedback in the previous example: the phase margin is too small. The phase at the crossover frequency is 169° , so the phase margin is too small ($\varphi_m = 11^\circ$). The closed-loop system will be too close to instability, with oscillations in the transient response.

In order to improve the phase margin, we can use a proportional-derivative controller

$$C(s) = k_p + k_d s.$$

This controller has a zero at $-k_p/k_d$, which will increase the phase, and as with the PI controller in the previous example, we should place the zero before the crossover frequency, so that the phase increase happens before crossover. Choosing $k_p/k_d \approx 5 \text{ rad/sec}$ gives a phase at 10 rad/sec of 105° , for a phase margin of $\varphi_m = 75^\circ$. Choosing k_p as in the above examples to set the crossover frequency at 10 rad/sec, we find $k_p = 9.1$, so the overall PD controller is

$$C_1(s) = 9.1 + 1.82s. \quad (4.23a)$$

The resulting loop gain $L_1 = PC_1$ is shown in Figure 4.33, along with the closed-loop step response, and both look good: the step response has a small overshoot, and it not oscillatory, and there is a nice large gain at low frequencies. The Nyquist plot of L_1 is also shown in Figure 4.34, and stays sufficiently far from the -1 point, indicating good robustness.

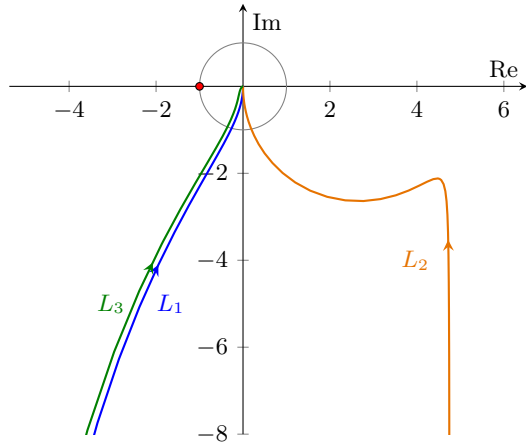


Figure 4.34: Nyquist plot of the loop gain $L = PC$ for controllers given by (4.23a)–(4.23c).

Note that the choice of where to place the zero k_p/k_d is important. One might think that, if 75° phase margin is good, maybe an even bigger phase margin is better, and we can get a bigger phase margin by moving the zero to a lower frequency. What happens if we choose $k_p/k_d = 0.1$ instead of 5? Choosing $k_p = 0.2$ to set the crossover frequency at 10 rad/sec, we obtain a controller

$$C_2(s) = 0.2 + 2s. \quad (4.23b)$$

Compared to C_1 , now the proportional gain is much smaller, and the derivative gain is a little larger. The corresponding Bode plot of the loop gain $L_2 = PC_2$ is also shown in Figure 4.33, along with the step response, and we see the performance of C_2 is dramatically worse than C_1 . From the Bode plot, the gain is much lower for low frequencies, indicating worse performance. In the step response, we see the initial transient is still fast, but then it appears the response has a non-zero steady-state error. Actually, the steady-state error is still zero (we know it must be since $|L| \rightarrow \infty$ as $s \rightarrow 0$), but it takes a long time to get there: the settling time is about 25 seconds, compared with 1 second for controller C_1 . So the system initially responds quickly, but then has a very slow transient. (Intuitively, this is because there is a (slow) closed-loop pole close to the open-loop zero we placed at $s = -0.1$; a root-locus plot helps explain why this happens.) The Nyquist plot for L_2 is also shown in Figure 4.34, and indicates the stability margins are very large—actually too large; we are sacrificing performance unnecessarily.

So C_1 looks like a good controller (certainly much better than C_2), but we can still do better. Figure 4.35 shows a Bode plot of the controllers C_1 and C_2 . There are a few things to notice from this plot. First, note that, at low frequencies, the magnitude of C_2 is much smaller than the magnitude of C_1 . We know we usually want our controllers to have high gain where we desire performance, so this also helps illustrate why C_2 performs poorly. But

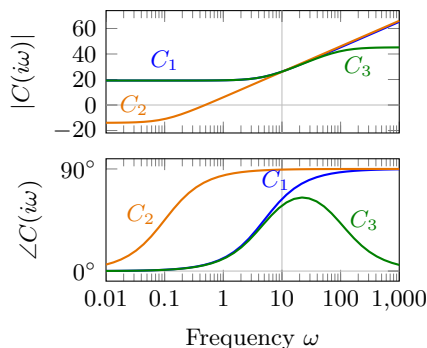


Figure 4.35: Bode plots of controllers C_1 , C_2 , and C_3 , from equations (4.23a)–(4.23c). Note that C_1 and C_3 are almost identical for frequencies below 10 rad/sec.

another observation is that both C_1 and C_2 have *infinite* gain as $\omega \rightarrow \infty$. In practice, this is impossible to achieve, and even if it were possible to achieve, it would not be desirable, because of amplification of sensor noise (see §2.6). Now, the main reason we introduced the zero in our controller is for our controller to provide some positive phase, which we needed to improve the phase margin at crossover. But after the crossover frequency of 10 rad/sec, we no longer care about the phase anymore.

A common way to address this problem and reduce the gain at high frequencies is to introduce a low-pass filter, or simply another pole faster than the crossover frequency:

$$\frac{1}{s/p + 1}, \quad p \gg \omega_c.$$

This term will add more negative phase (-45° at frequency p , up to -90° for frequencies more than about $10p$ —see §4.3), but as long as we make p large enough, this phase will not have much affect at the crossover frequency. If we make p too large, though, then it will not have its desired effect of reducing the gain at high frequencies. Here, let us set $p = 100$, and combine our low-pass filter with C_1 , to obtain

$$C_3(s) = \frac{C_1(s)}{s/p + 1}. \quad (4.23c)$$

This controller is shown in Figure 4.35, and we see that its gain and phase looks identical to C_1 for low frequencies, but for higher frequencies, the gain stays constant (while C_1 keeps increasing). The phase of C_3 at the desired crossover frequency (10 rad/sec) is a little worse than the phase of C_1 , but C_3 still gives us a significant phase benefit. The Nyquist plot of the corresponding loop gain $L_3 = PC_3$ is also shown in Figure 4.34, and looks almost identical to L_1 , except for a slightly smaller phase margin. The corresponding Bode plot and closed-loop step response are also shown in Figure 4.33: the Bode plot of L_3 is almost identical to that of L_1 , except at high frequen-

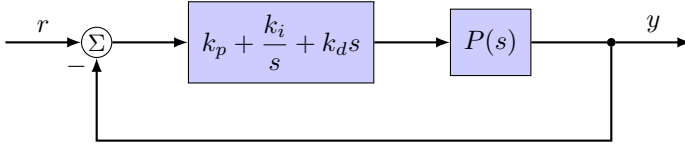


Figure 4.36: PID-controller for the spring-mass-damper system in Exercise 4.1.

cies, where the gain is smaller (a good thing); and the step responses look practically identical. For this system, C_3 is the best controller, by far. \diamond

Exercises

- 4.1.** Consider a spring-mass-damper system, where the mass is m , the spring constant is k , and the damper constant is c . The transfer function from an applied force to the mass position is given by

$$P(s) = \frac{1}{m(s^2 + 2\zeta\omega_0 s + \omega_0^2)},$$

where $\omega_0 = \sqrt{k/m}$ and $\zeta = c/(2\sqrt{mk})$. We will use a PID controller as shown in Figure 4.36.

- Find expressions for the PID gains k_p , k_i , and k_d such that the closed-loop transfer function has the characteristic polynomial (i.e., denominator) $s^3 + a_2 s^2 + a_1 s + a_0$, for some given constants a_0, a_1, a_2 .
- Suppose $m = 1$, $\zeta = 0.5$, and $\omega_0 = 2$. Compute a PID controller that places the closed-loop poles at $p = -3, -2 \pm 2i$. Plot the resulting response to a step reference r .

- 4.2.** For the control system from Exercise 4.1, consider the following scenarios:

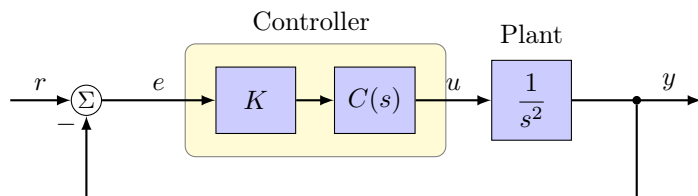
- k_i and k_d are as in Ex. 4.1, but k_p varies from 0 to a large value;
- k_p and k_d are as in Ex. 4.1, but k_i varies from 0 to a large value;
- k_p and k_i are as in Ex. 4.1, but k_d varies from 0 to a large value.

For each scenario,

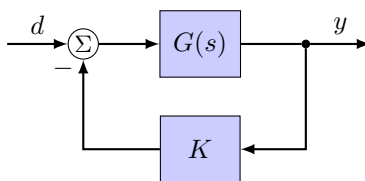
- Find a transfer function $G(s)$ such that the closed-loop poles are the roots of $1 + kG(s)$, where k is either k_p , k_i , or k_d .
- Use a computer to plot the root locus for the given scenarios.

- (c) Briefly describe what would happen to the mass's step response as each gain becomes large. What are the implications for effective control?

4.3. Consider a feedback system whose block diagram is given below, to control the double integrator $P(s) = 1/s^2$:



- (a) Find the transfer function from r to y , and the corresponding equation for the closed-loop poles. Place this in root-locus form $1 + KG(s) = 0$, and specify $G(s)$.
- (b) For each of the controllers below, draw a root locus plot showing how the poles vary as the gain K varies from 0 to ∞ . You should first sketch the plots by hand, noting any asymptotes, but you may use a computer to check your sketches.
- Proportional controller $C(s) = 1$
 - Proportional-derivative controller $C(s) = 1 + s$
 - Proportional-integral controller $C(s) = 1 + \frac{1}{s}$
- (c) Which controller works best, and why?
- 4.4.** This problem deals with root locus plots for the control system shown in the following block diagram:



For each transfer function below,

- Sketch, first *by hand* and *without using a computer*, the root-locus plot with respect to K .
- Briefly explain the rules you applied to obtain your sketch, and show any calculations you used for asymptotes.
- Verify your sketch with a computer, and explain any differences from your sketch.

- Determine if the closed-loop system is stable for *all*, *some*, or *none* of $K \geq 0$.

$$(a) \quad G(s) = \frac{(s+3)(s+7)}{s(s+1)(s+5)(s+10)}$$

$$(b) \quad G(s) = \frac{s^2 + s + 10}{s(s^2 + s + 12)}$$

$$(c) \quad G(s) = \frac{s+5}{(s+1)(s+2)(s+3)(s+7)}$$

$$(d) \quad G(s) = \frac{s+2}{s^2(s+10)(s^2+6s+18)}$$

$$(e) \quad G(s) = \frac{s+2}{s^2(s+9)}$$

4.5. Show that, for $G(s)$ given by (4.10), the peak frequency on the Bode plot occurs at $\omega_p = \omega_0 \sqrt{1 - 2\zeta^2}$, and the peak magnitude can be approximated, for small ζ , by $1/(2\zeta)$.

4.6. Sketch the asymptotes of the Bode plot magnitude and phase for each of the following transfer functions. After completing the hand sketches, verify your results using a computer.

$$(a) \quad G(s) = \frac{(bs+1)^2}{s(as+1)^2}, \quad 0 < a \ll 1 \ll b$$

$$(b) \quad G(s) = k_d s + k_p + \frac{k_i}{s}, \quad 0 < k_i \approx k_d \ll k_p$$

$$(c) \quad G(s) = \frac{k(s+a)}{(s-b)(s+b)(s+c)}, \quad 0 < a \ll b \ll c, \quad k > 0.$$

4.7. We wish to construct a minimum-phase transfer function (no right-half-plane poles or zeros) with the following properties:

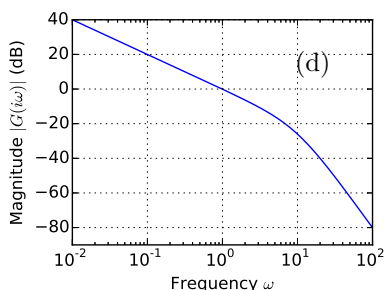
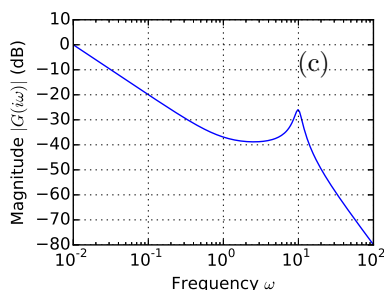
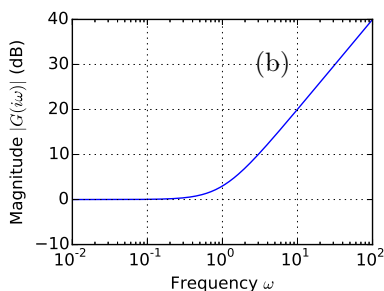
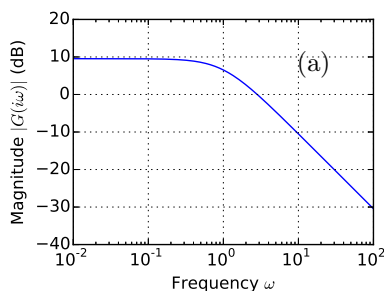
- the zero-frequency gain is $A > 1$,
- the gain at high frequencies is $B < 1$, and
- for mid frequencies, the slope is -20 dB per decade, achieving a magnitude of roughly 1 at frequency ω_b .

(In control design, transfer functions such as this are often used to specify performance requirements at different frequencies.)

- Illustrate the specifications on a Bode plot. How many poles and zeros are required?
- Find a transfer function $W(s)$ with these properties. Remember that the third property need only be approximate.

- (c) Check your result by plotting a Bode plot of your $W(s)$, in MATLAB, for $A = 2$, $B = 10^{-2}$, and $\omega_b = 10$.

4.8. Each of the plots below shows the magnitude of a minimum-phase transfer function (no RHP poles or zeros). For each transfer function, sketch the corresponding Bode phase plot, and give a brief explanation of how you arrived at your sketch.



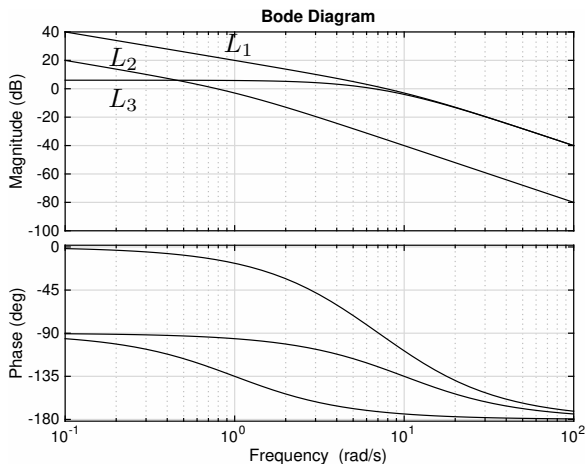
4.9. You have been hired by CoolAudioPro L.L.C. to design linear filters for live and studio applications. For each of the following real-world situations, construct a transfer function that satisfies the given requirements, and use a computer to draw the Bode plot.

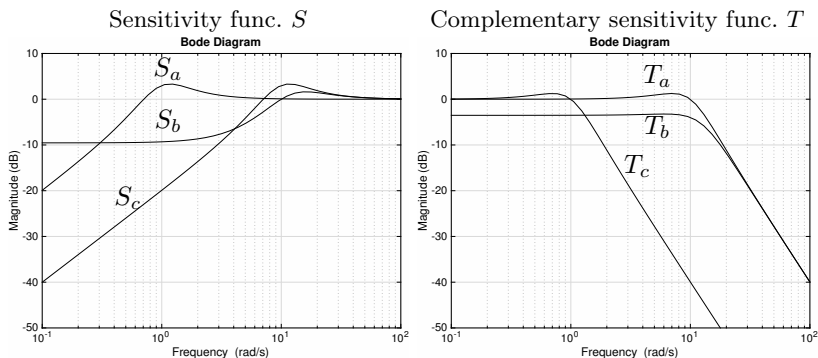
Some things to keep in mind:

- $1 \text{ Hz} = 2\pi \text{ rad/s}$; you must first convert frequencies to rad/s.
 - The audible frequency range is 20 Hz to 20000 Hz; frequencies outside this range are typically irrelevant.
 - Keep the filters minimum phase, unless it is necessary to use right-half-plane zeros.
- (a) Digital filters that simulate natural reverberations can sound harsh if the raw signal's high-frequency content is too strong. To correct this, design a *low-pass filter* that minimally changes the magnitude at low- to mid-frequencies, but rolls off high frequencies above 7000 Hz, with a slope of $-20 \text{ dB per decade}$.

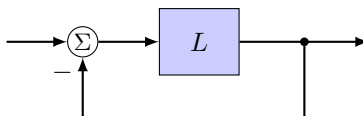
- (b) Female vocalists rarely sing below 200 Hz, but “plosives” (such as the “p” sound) can create large low-frequency spikes in the microphone. To correct this, design a *high-pass filter* that minimally changes the magnitude at mid- to high-frequencies, but rolls off low frequencies below 200 Hz, with a steep slope of 60 dB per decade.
- (c) Sound engineers for rock and pop music sometimes reduce the magnitude at mid-frequencies to provide a heavier, brighter sound. Design a mild *band-rejection filter* that minimally changes the magnitude near 20 Hz and 20000 Hz, but reduces the magnitude by up to about 6 dB in a broad range around 800 Hz.
- (d) In the studio, recording a single sound source with multiple microphones can improve tone and stereo spread. It is common, however, for destructive interference to create undesirable “phasing” effects. To correct this, design an *all-pass filter* that has no effect on the magnitude, but gradually drops the phase from about 0° at 20 Hz to -180° at 500 Hz, and finally to about -360° at 20000 Hz.
- (e) At live concerts of your favorite band, “Hedgeclipper,” the lead singer’s microphone frequently experiences unstable microphone feedback at a frequency of 400 Hz. To correct this, design a *notch filter* that minimally changes the magnitude at nearly all frequencies, except in a narrow frequency range centered at 400 Hz, where it provides roughly 30 dB of attenuation.

4.10. The following Bode plot shows three different loop transfer functions, L_1, L_2, L_3 . Match each loop transfer function with the corresponding sensitivity function $S = 1/(1 + L)$ and complementary sensitivity function $T = L/(1 + L)$. Justify your answer.





4.11. Consider the feedback loop below:



For each of the loop transfer functions L listed below, plot the Nyquist plot and use the Nyquist criterion to determine the stability of the closed-loop system. In particular, for each system, state the number of (closed-loop) right-half-plane poles.

(a) $L(s) = \frac{20}{(s+1)(s+10)}$

(b) $L(s) = \frac{-20}{(s+1)(s+10)}$

(c) $L(s) = \frac{20}{(s-1)(s+10)}$

(d) $L(s) = \frac{4}{(s-1)(s+10)}$

(e) $L(s) = \frac{4}{s(s+1)(s+10)}$

(f) $L(s) = \frac{1-s}{s(s+2)}$

(g) $L(s) = \frac{s-1}{s(s+2)}$

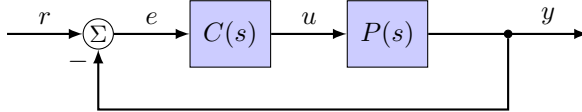
4.12. For a plant with the transfer function

$$P(s) = \frac{1}{(2s+1)(0.5s+1)(0.1s+1)},$$

consider the three PI controllers

$$C_1(s) = 0.2 + \frac{1}{s}, \quad C_2(s) = 2 + \frac{1}{s}, \quad C_3(s) = 2 + \frac{0.1}{s},$$

to be used for tracking a reference, as in the following block diagram:



- (a) For each of these controllers, find the gain and phase margins using Matlab, and illustrate them on the appropriate Bode and Nyquist plots. Show all three controllers on the same axes. (This makes it easier to compare them, and also saves paper. It may be difficult to see the gain margin clearly on the Nyquist plot, and that is okay.)
 - (b) Plot the response $y(t)$ of the closed-loop system, for a step input $r(t)$. Show all three controllers on the same plot, and compare them. Which controller is the “best”? How could you tell this from the Bode and Nyquist plots?
- 4.13.** When plotting a Nyquist plot in MATLAB (with the `nyquist` command), turning the grid on produces a grid with lines along which the complementary sensitivity function $|T(i\omega)|$ is a constant. Find formulas for these lines. In particular, show that if $L = x + iy$ (with $x, y \in \mathbb{R}$), then $|T| = 1$ for $x = -1/2$. If $|T| \neq 1$, show that curves of constant $|T|$ are circles $(x - a)^2 + y^2 = a(1 + a)$ for $a = |T|^2/(1 - |T|^2)$.
- 4.14.** Show that, if the phase margin is less than 90° , then $\omega_B < \omega_c < \omega_{BT}$.

Chapter 5

State-space systems

The frequency-domain methods that we have studied in Chapters 2–4 are the main tools of “classical” control theory. We now shift our focus back to time-domain (state-space) methods. We first encountered state-space systems in Section 1.2, but then immediately transformed them to the frequency domain (finding the corresponding transfer function). The tools we will develop in the next few sections will work directly with the state-space representations, not transfer functions. These tools comprise what is known as “modern control,” in which the focus is on linear algebra, rather than graphical methods. We will introduce notions of “controllability” and “observability” that are completely absent in the classical approach. We will develop systematic procedures for control design, for instance using state feedback and optimal control, that can be both effective and easy to use. Some distinctions between classical and modern control are summarized in Table 5.1.

It is tempting to think that, because the modern tools are “systematic” and “optimal” that they are in all ways better than the classical tools. It

Classical	Modern
Transfer functions	State space
$Y(s) = G(s) \cdot U(s)$	$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ $\mathbf{y} = \mathbf{Cx} + \mathbf{Du}$
Frequency domain	Time domain
Bode, Nyquist plots	Controllability, observability
PID controllers	State feedback, observers
Manual tuning of gains	Systematic methods, optimal control

Table 5.1: Differences between classical and modern control tools.

would be a mistake to draw this conclusion. For simple problems, the classical tools can actually give greater insight into what is happening with feedback; furthermore, we shall see that even after designing a controller using modern tools such as optimal control, the best way to understand how well the controller will really work in practice is to go back to the classical tools such as Bode and Nyquist plots.

But let us proceed, and begin to develop these modern tools.

5.1 Solution of state-space equations

Recall the form of a linear state-space system, encountered in Section 1.2:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (5.1a)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (5.1b)$$

Here, \mathbf{A} is an $n \times n$ matrix, and \mathbf{x} is a vector in \mathbb{R}^n . We will mainly be concerned with the case where the input u and output y are scalars (and hence \mathbf{B} is a column vector, \mathbf{C} is a row vector, and \mathbf{D} is a scalar). However, most of the techniques we will study easily generalize to the case of multiple inputs and outputs. The system of equations (5.1) is known as a *linear time-invariant (LTI) system*. If the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ depend on time, then we have a linear time-varying (LTV) system. There are also tools for working with these (for instance, see page 200, or an optimal control course such as MAE 546), but they are more complicated, and here we restrict ourselves to the time-invariant case.

Zero-input solution: the matrix exponential

For now, we will focus on the state equation (5.1a), and in particular, we will first consider the case where the input \mathbf{u} is zero:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (5.2)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$. We want to understand solutions of this equation. In MATLAB, one can solve this using the `initial` command, but we will need to know something about the analytical solution in order to develop more tools for feedback control.

First, consider the case where $n = 1$ (so $\mathbf{A} = a$ is a scalar), and (5.2) becomes

$$\dot{x} = ax, \quad x(0) = x_0.$$

This solution is simply

$$x(t) = e^{at}x_0,$$

as can be found by separating variables and integrating:

$$\begin{aligned}\int_{x_0}^{x(t)} \frac{dx}{x} &= \int_0^t a \, dt \\ \implies \log x(t) - \log x_0 &= at \\ \implies x(t) &= e^{at} x_0.\end{aligned}$$

For the matrix case ($n > 1$), there is an analogous expression:

The solution to (5.2) is

$$\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{x}_0. \quad (5.3)$$

But what is meant by $e^{\mathbf{A}t}$ when \mathbf{A} is a matrix? This expression in fact looks quite bizarre: what could it possibly mean to raise e to the power of a matrix? We need to define what this means in order for (5.3) even to make sense.

Recall that, if a is a scalar, we may expand e^{at} in a Taylor series about $t = 0$:

$$e^{at} = 1 + at + \frac{a^2 t^2}{2!} + \frac{a^3 t^3}{3!} + \cdots = \sum_{k=0}^{\infty} a^k \frac{t^k}{k!}.$$

This series expansion is something that generalizes to the matrix case. Let us define

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \mathbf{A}^2 \frac{t^2}{2!} + \mathbf{A}^3 \frac{t^3}{3!} + \cdots = \sum_{k=0}^{\infty} \mathbf{A}^k \frac{t^k}{k!}, \quad (5.4)$$

known as the *matrix exponential*. In the above expression,

$$\mathbf{A}^2 = \mathbf{A}\mathbf{A}, \quad \mathbf{A}^3 = \mathbf{A}\mathbf{A}\mathbf{A}, \quad \text{etc},$$

with the usual matrix multiplication. Thus, if \mathbf{A} is an $n \times n$ matrix, then $e^{\mathbf{A}t}$ is also an $n \times n$ matrix.

The claim is that, with the above definition of $e^{\mathbf{A}t}$, the expression (5.3) is the solution to (5.2). First, we check that the initial condition is satisfied:

$$\mathbf{x}(0) = e^{\mathbf{A} \cdot 0} \mathbf{x}_0 = (\mathbf{I} + \mathbf{A} \cdot 0 + \cdots) \mathbf{x}_0 = \mathbf{I} \mathbf{x}_0 = \mathbf{x}_0,$$

as desired. Next, we check that the differential equation is satisfied:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \frac{d}{dt} (\mathbf{I} + \mathbf{A}t + \mathbf{A}^2 \frac{t^2}{2!} + \mathbf{A}^3 \frac{t^3}{3!} + \cdots) \mathbf{x}_0 \\ &= (\mathbf{0} + \mathbf{A} + \mathbf{A}^2 t + \mathbf{A}^3 \frac{t^2}{2!} + \cdots) \mathbf{x}_0 = \mathbf{A} e^{\mathbf{A}t} \mathbf{x}_0 = \mathbf{A} \mathbf{x}(t),\end{aligned}$$

which was to be shown. Thus, indeed (5.3) is the solution, as claimed.

⚡ A few technical points: first, in order to justify differentiating the series (5.4) term-by-term, we may apply Theorem B.1. The series (5.4) converges for all t , so this is indeed justified. Second, in order to show (5.3) is *the* solution of (5.2), we need to know that solutions are unique. A basic fact from the theory of ODEs is that the equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ has a unique solution in the neighborhood of $\mathbf{x}(0) = \mathbf{x}_0$ if $\mathbf{f}(\mathbf{x})$ is differentiable at \mathbf{x}_0 [5]. Since $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ is everywhere differentiable (with derivative $D\mathbf{f}(\mathbf{x}_0) = \mathbf{A}$), the linear system (5.2) always has a unique solution.

Example 5.1 (Double integrator). Consider the input-output system

$$\ddot{y} = u,$$

which might model the angular position of a satellite (see Example 4.6). The transfer function for this system is $1/s^2$, but here we are interested in a state-space realization. To convert this second-order ODE into two first-order ODEs, we employ the usual trick of appending a derivative, and set $\mathbf{x} = (x_1, x_2) = (y, \dot{y})$, to obtain the equations

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= u,\end{aligned}$$

with output equation $y = x_1$. We can write this in matrix form as $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We are interested in the zero-input solution, for $u = 0$, with initial condition $\mathbf{x}(0) = (a, b)$. We can calculate the solution by elementary means: since $u = 0$, we have $\dot{x}_2 = 0$, so $x_2 = \text{constant}$. Applying the initial condition $x_2(0) = b$, we find $x_2 = b$. For x_1 , we then have $\dot{x}_1 = b \implies x_1 = bt + \text{constant}$. From the initial condition $x_1(0) = a$, we then have $x_1 = a + bt$.

Now, let us find the solution using the matrix exponential. Note that $\mathbf{A}^2 = 0$, so the sum in (5.4) terminates after a finite number of terms (namely, two terms), and we have

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + 0 + \cdots = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}.$$

This agrees with our earlier solution, since, for $\mathbf{x}(0) = (a, b)$, we have

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a + bt \\ b \end{bmatrix}. \quad \diamond$$

Properties of the matrix exponential

We now discuss some properties of the matrix exponential. It is important to become comfortable with these, so one may know how to manipulate expressions involving the matrix exponential.

1. $e^{\mathbf{A} \cdot 0} = \mathbf{I}$.
2. $e^{\mathbf{A}(t_1+t_2)} = e^{\mathbf{A}t_2}e^{\mathbf{A}t_1} = e^{\mathbf{A}t_1}e^{\mathbf{A}t_2}$.
3. $(e^{\mathbf{A}t})^{-1} = e^{-\mathbf{A}t}$. Note that this implies that $e^{\mathbf{A}t}$ is always invertible (even if \mathbf{A} is not invertible).
4. $(e^{\mathbf{A}t})^\top = e^{\mathbf{A}^\top t}$, where \mathbf{A}^\top denotes the transpose of \mathbf{A} (swapping rows and columns).
5. $\mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A}$.
6. $\frac{d}{dt}e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t}$.
7. $e^{\mathbf{T}\mathbf{A}\mathbf{T}^{-1}t} = \mathbf{T}e^{\mathbf{A}t}\mathbf{T}^{-1}$.

Proof. We have actually already shown properties 1 and 6, in showing that (5.3) is the solution to (5.2). We prove properties 2 and 3, and leave the remainder as exercises. For property 2, consider an initial condition $\mathbf{x}(0) = \mathbf{x}_0$. Then

$$\mathbf{x}(t_1 + t_2) = e^{\mathbf{A}(t_1+t_2)}\mathbf{x}_0.$$

But we also have $\mathbf{x}(t_1) = e^{\mathbf{A}t_1}\mathbf{x}_0$, and thus, taking this as an initial condition and integrating for time t_2 , we have

$$\mathbf{x}(t_1 + t_2) = e^{\mathbf{A}t_2}\mathbf{x}(t_1) = e^{\mathbf{A}t_2}e^{\mathbf{A}t_1}\mathbf{x}_0.$$

Since these two expressions must be equal for any \mathbf{x}_0 , we have $e^{\mathbf{A}(t_1+t_2)} = e^{\mathbf{A}t_2}e^{\mathbf{A}t_1}$. Swapping the roles of t_1 and t_2 then gives $e^{\mathbf{A}(t_1+t_2)} = e^{\mathbf{A}t_1}e^{\mathbf{A}t_2}$.

For property 3, note that

$$e^{\mathbf{A}t}e^{-\mathbf{A}t} = e^{\mathbf{A}t}e^{\mathbf{A}(-t)} = e^{\mathbf{A}(t-t)} = e^{\mathbf{A} \cdot 0} = \mathbf{I},$$

where the second equality follows from property 2, and the final equality follows from property 1. \square

From the above properties, one might hope that any property that holds for the scalar exponential e^{at} also holds for the matrix exponential $e^{\mathbf{A}t}$. However, this is not the case: in particular, if \mathbf{A} and \mathbf{B} are two matrices of the same dimension, then

$$e^{\mathbf{A}t}e^{\mathbf{B}t} = e^{(\mathbf{A}+\mathbf{B})t} \iff \mathbf{A}\mathbf{B} = \mathbf{B}\mathbf{A}. \quad (5.5)$$

Thus, it is *not* true in general that $e^{\mathbf{A}t}e^{\mathbf{B}t} = e^{(\mathbf{A}+\mathbf{B})t}$, unlike in the scalar case. For the matrix case, this expression holds only if \mathbf{A} and \mathbf{B} commute.



There is an even more general way of describing the exponential map, in which one *defines* the exponential map to be the solution of an ODE such as (5.2). In this more general setting, one starts with a *Lie group* G , which is a group that is also a differentiable manifold; one then defines the *Lie algebra* \mathfrak{g} , a vector space that is the tangent space of G at the identity. Given a “direction” $\xi \in \mathfrak{g}$, one first defines a left-invariant (or right-invariant) vector field, by left (or right) translation of ξ to an arbitrary base point $g \in G$. One can then integrate the resulting ODE, starting at the identity, to form a curve $h(t) \in G$, such that $h(0)$ is the identity. Finally, one defines $\exp : \mathfrak{g} \rightarrow G$ to be $h(1)$. The matrix case described in this section is a special case of this theory, for the group $G = \text{GL}(n)$, the space of invertible $n \times n$ matrices; the Lie algebra of this group is $\mathfrak{g} = \mathfrak{gl}(n)$, the vector space of all $n \times n$ matrices. So in (5.2), one thinks of A as an element of $\mathfrak{gl}(n)$; the corresponding (right-invariant) vector field is simply $\dot{X} = AX$, for $X \in \text{GL}(n)$, and \exp maps a matrix $A \in \mathfrak{gl}(n)$ to an invertible matrix $e^A \in \text{GL}(n)$. The theory of Lie groups is quite a powerful theory, with many useful practical applications: for instance, the configuration space of a spinning satellite is properly thought of as a Lie group, the group of orthogonal 3×3 matrices with determinant 1, denoted $\text{SO}(3)$; its Lie algebra $\mathfrak{so}(3)$ is the space of skew-symmetric matrices; the exponential map then maps skew-symmetric matrices to orthogonal matrices. For much more on this topic, see [8, 19].

Computing the matrix exponential

We now know, at least in principle, how to solve linear systems of the form (5.2): the solution is simply (5.3). But how exactly do we compute the matrix exponential $e^{\mathbf{A}t}$? We certainly do not want to evaluate an infinite number of terms in the series (5.4). Here, we discuss some special cases for which computing $e^{\mathbf{A}t}$ is particularly simple. We will discuss more general cases in Section A.1. (Note that there are many different ways of computing the matrix exponential: for an interesting survey of some of them, see [20], by the original author of MATLAB.)

The simplest case is when \mathbf{A} is diagonal. In this case, each term in the series (5.4) is also a diagonal matrix, so if

$$\mathbf{A} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}, \quad \text{then} \quad e^{\mathbf{A}t} = \begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix}.$$

The computation is also easy if \mathbf{A} is *strictly* upper (or lower) triangular (i.e., with zeros on the diagonals):

$$\mathbf{A} = \begin{bmatrix} 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}^2 = \begin{bmatrix} 0 & 0 & * \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}^3 = 0.$$

In this case, $\mathbf{A}^k = 0$, for all $k \geq n$ (such a matrix is called *nilpotent*),

so the series (5.4) terminates after a finite number of terms. For instance, Example 5.1 falls into this category.

Another way is to simply recognize the series expansion, as the following example shows.

Example 5.2. Compute $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (5.6)$$

We do this by simply evaluating the terms in the series (5.4). We calculate

$$\mathbf{A}^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = -\mathbf{I}, \quad \mathbf{A}^3 = -\mathbf{A}, \quad \mathbf{A}^4 = \mathbf{I}.$$

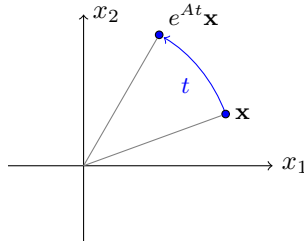
Thus,

$$\begin{aligned} e^{\mathbf{A}t} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -t \\ t & 0 \end{bmatrix} + \begin{bmatrix} -t^2/2! & 0 \\ 0 & -t^2/2! \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & t^3/3! \\ -t^3/3! & 0 \end{bmatrix} + \begin{bmatrix} t^4/4! & 0 \\ 0 & t^4/4! \end{bmatrix} + \cdots \\ &= \begin{bmatrix} 1 - t^2/2! + \cdots & -t + t^3/3! + \cdots \\ t - t^3/3! - \cdots & 1 - t^2/2! + \cdots \end{bmatrix}. \end{aligned}$$

Recognizing these series expansions, we find

$$e^{\mathbf{A}t} = \begin{bmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{bmatrix}.$$

You may recognize this as a rotation matrix: it rotates a vector $\mathbf{x} = (x_1, x_2)$ counterclockwise through an angle t , as shown in the diagram below:



Note that the matrix \mathbf{A} in (5.6) behaves in many ways like the imaginary unit $i = \sqrt{-1}$. For instance, $\mathbf{A}^2 = -\mathbf{I}$, just as $i^2 = -1$. Similarly, given a vector $\mathbf{x} = (x_1, x_2)$, $e^{\mathbf{A}t}\mathbf{x}$ is a counterclockwise rotation by angle t , just like for a complex number $z = re^{i\theta}$, $e^{it}z = re^{i(\theta+t)}$ is a counterclockwise rotation by an angle t . \diamond

Another way to compute the matrix exponential, without resorting to writing the infinite series, is to use *diagonalization*. (See Appendix A.3 for a review of diagonalization.) This method will work as long as \mathbf{A} is diagonalizable (that is, it has n linearly independent eigenvectors). This is true for most matrices—for instance, it is always true if the eigenvalues of \mathbf{A} are distinct—but there are some matrices that arise in practice that are not diagonalizable, such as the double integrator of Example 5.1.

This method works as follows: if we know the eigenvectors and eigenvalues of \mathbf{A} , then we write $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where \mathbf{V} is a matrix whose columns are eigenvectors of \mathbf{A} , and $\mathbf{\Lambda}$ is a diagonal matrix of the corresponding eigenvalues. Then, using Property 7 of the matrix exponential (page 147), we find

$$e^{\mathbf{A}t} = e^{\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}t} = \mathbf{V}e^{\mathbf{\Lambda}t}\mathbf{V}^{-1} = \mathbf{V} \begin{bmatrix} e^{\lambda_1 t} & & \\ & \ddots & \\ & & e^{\lambda_n t} \end{bmatrix} \mathbf{V}^{-1}. \quad (5.7)$$

Thus a systematic procedure for computing $e^{\mathbf{A}t}$ is as follows:

- Find eigenvalues λ_j and eigenvectors \mathbf{v}_j of \mathbf{A} .
- Stack eigenvectors into a matrix $\mathbf{V} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_n]$.
- Find \mathbf{V}^{-1} .
- Compute $e^{\mathbf{A}t}$ from (5.7).

If one simply desires a numerical calculation, the MATLAB commands `expm` or `initial` will do the job. Note that, in MATLAB, `exp(A)` does *not* compute the matrix exponential $e^{\mathbf{A}t}$: it computes a matrix whose entries are $e^{a_{ij}}$ where a_{ij} are the elements of \mathbf{A} . To compute the matrix exponential, be sure to use `expm`.

Input-output response

Now let us return to our original system (5.1) and consider the response for a nonzero input $u(t)$. We claim the following:

The solution to (5.1a) is

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau) d\tau. \quad (5.8)$$

Note that the first term of this expression is the zero-input response (5.3), and the second term is the response assuming the initial state \mathbf{x}_0 is zero. To show that this is the solution of (5.1a), we differentiate, to obtain

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) &= \mathbf{A}e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t \mathbf{A}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau) d\tau + e^{\mathbf{A}(t-t)}\mathbf{B}u(t) \\ &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \end{aligned}$$

so $\mathbf{x}(t)$ satisfies the ODE in (5.8). Furthermore, $\mathbf{x}(0) = \mathbf{x}_0$, so $\mathbf{x}(t)$ given by (5.8) is indeed the (unique) solution to (5.1a). Note that this expression is valid even if the input $\mathbf{u}(t)$ is a vector (i.e., for a system with multiple inputs).

The corresponding solution of the output equation (5.1b) is then

$$\begin{aligned}\mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \\ &= \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau) d\tau + \mathbf{D}\mathbf{u}(t).\end{aligned}$$

This expression may be written

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t \mathbf{H}(t-\tau)\mathbf{u}(\tau) d\tau, \quad (5.9)$$

where

$$\mathbf{H}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B} + \mathbf{D}\delta(t) \quad (5.10)$$

is the impulse response. Like (5.8), this expression is also valid even if \mathbf{u} and \mathbf{y} are vectors; in this case, $\mathbf{H}(t)$ is a matrix for which the number of columns is the number of inputs (components of \mathbf{u}), and the number of rows is the number of outputs (components of \mathbf{y}). If \mathbf{u} and \mathbf{y} are both scalars (a single-input, single-output system), then $\mathbf{H}(t)$ is a scalar, and this is in fact precisely the same impulse response as that defined in Section 2.2, only written in terms of the state-space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$.



You may be wondering why (5.9) differs slightly from our earlier expressions such as (2.4). There are several reasons for this. First, state-space systems are always causal, so the impulse response $h(t)$ in (2.4) is always zero for $t < 0$. Second, the solution (5.9) is valid only for $t > 0$; instead of using an input $u(t)$ that may be defined for $t < 0$, we ignore any inputs prior to $t = 0$ and start with an initial state \mathbf{x}_0 at $t = 0$. But with these two caveats, the two expressions are equivalent.

In order to make some more connections with transfer functions and the frequency-domain methods from the previous chapters, let us briefly consider how these state-space solutions appear in the frequency domain. Consider first the zero-input equation (5.2), given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0. \quad (5.11)$$

Taking the Laplace transform gives

$$sX(s) - \mathbf{x}_0 = \mathbf{A}X(s),$$

where $X(s) = \mathcal{L}\{\mathbf{x}(t)\}$. Solving for $X(s)$, we have

$$\begin{aligned}sX(s) - \mathbf{A}X(s) &= \mathbf{x}_0 \\ \implies s\mathbf{I}X(s) - \mathbf{A}X(s) &= \mathbf{x}_0 \\ \implies (s\mathbf{I} - \mathbf{A})X(s) &= \mathbf{x}_0 \\ \implies X(s) &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}_0.\end{aligned}$$

(Note that, in the second equation, we needed to insert an identity matrix \mathbf{I} to obtain $s\mathbf{I} - \mathbf{A}$ instead of $s - \mathbf{A}$, because one cannot add a scalar and a matrix.) Now, we also know the solution of (5.11) is $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$. But $X(s) = \mathcal{L}\{\mathbf{x}(t)\}$, and hence

$$\mathcal{L}\{e^{\mathbf{A}t}\} = (s\mathbf{I} - \mathbf{A})^{-1}.$$

This expression is simply the matrix generalization of the familiar scalar formula

$$\mathcal{L}\{e^{at}\} = \frac{1}{s-a}.$$

This formula also provides further insight into our expression (2.11) for the transfer function of a state-space system. Recall from Section 2.3 that the transfer function may be defined as the Laplace transform of the impulse response. Taking the Laplace transform of (5.10) gives

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D},$$

which agrees with our earlier formula (2.11) for the transfer function of a state-space system.

5.2 Eigenvectors for linear ODEs

Consider the linear ODE

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{v}.$$

Let us consider what happens when the initial condition \mathbf{v} is an eigenvector of \mathbf{A} with eigenvalue λ :

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}.$$

(For a review of eigenvectors and eigenvalues, see Appendix A.1.) The solution is given by

$$\begin{aligned} \mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{v} = \left(I + t\mathbf{A} + \frac{t^2}{2!}\mathbf{A}^2 + \frac{t^3}{3!}\mathbf{A}^3 + \cdots\right)\mathbf{v} \\ &= \mathbf{v} + t\mathbf{A}\mathbf{v} + \frac{t^2}{2!}\mathbf{A}^2\mathbf{v} + \frac{t^3}{3!}\mathbf{A}^3\mathbf{v} + \cdots \\ &= \mathbf{v} + t\lambda\mathbf{v} + \frac{t^2}{2!}\lambda^2\mathbf{v} + \frac{t^3}{3!}\lambda^3\mathbf{v} + \cdots \\ &= \left(1 + \lambda t + \frac{(\lambda t)^2}{2!} + \frac{(\lambda t)^3}{3!} + \cdots\right)\mathbf{v} \\ &= e^{\lambda t}\mathbf{v}. \end{aligned}$$

Let us take a closer look at what we have just done: $e^{\mathbf{A}t}$ is an $n \times n$ matrix, while $e^{\lambda t}$ is a scalar. For a typical initial condition, $\mathbf{x}(t)$ evolves in a possibly complicated way ($\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$), but if the initial condition is an eigenvector \mathbf{v} , the solution $\mathbf{x}(t)$ never changes “direction” in state space: $\mathbf{x}(t)$ is always a scalar multiple of \mathbf{v} . This is a remarkable result. It may be summarized mathematically in the following theorem:

Theorem 5.3. *If \mathbf{v} is an eigenvector of \mathbf{A} with eigenvalue λ , then \mathbf{v} is an eigenvector of $e^{\mathbf{A}t}$ with eigenvalue $e^{\lambda t}$.*

What if λ and \mathbf{v} are complex? Then it does not make sense to set $\mathbf{x}(0) = \mathbf{v}$, since for a physical system in which the states represent actual physical quantities, $\mathbf{x}(t)$ must be real. In this case, we can let

$$\mathbf{x}(0) = \text{Re}(\mathbf{v}) = \frac{1}{2}(\mathbf{v} + \bar{\mathbf{v}}),$$

where $\bar{\mathbf{v}}$ denotes the complex conjugate of \mathbf{v} . We then have

$$\begin{aligned}\mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) = \frac{1}{2}(e^{\mathbf{A}t}\mathbf{v} + e^{\mathbf{A}t}\bar{\mathbf{v}}) \\ &= \frac{1}{2}(e^{\mathbf{A}t}\mathbf{v} + \overline{e^{\mathbf{A}t}\mathbf{v}}) \\ &= \text{Re}(e^{\mathbf{A}t}\mathbf{v}) \\ &= \text{Re}(e^{\lambda t}\mathbf{v}),\end{aligned}$$

so now $\mathbf{x}(t)$ evolves in a 2-dimensional subspace spanned by the real and imaginary parts of \mathbf{v} .

To summarize:

If \mathbf{v} is an eigenvector, with $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, then

$$\begin{aligned}\mathbf{x}(0) = \mathbf{v} &\implies \mathbf{x}(t) = e^{\lambda t}\mathbf{v} \\ \mathbf{x}(0) = \text{Re}(\mathbf{v}) &\implies \mathbf{x}(t) = \text{Re}(e^{\lambda t}\mathbf{v}).\end{aligned}$$

More generally, if we can write the initial condition as a linear combination of eigenvectors

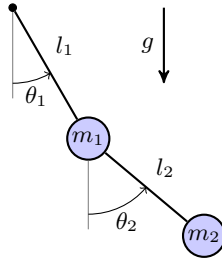
$$\mathbf{x}(0) = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots + \alpha_n\mathbf{v}_n,$$

with $\mathbf{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j$, then the response is

$$\mathbf{x}(t) = \alpha_1 e^{\lambda_1 t}\mathbf{v}_1 + \cdots + \alpha_n e^{\lambda_n t}\mathbf{v}_n. \quad (5.12)$$

The eigenvectors of \mathbf{A} are sometimes referred to as *modes* of the system, and an arbitrary response $\mathbf{x}(t)$ is then a linear combination of modes.

Example 5.4 (Double pendulum). Consider the dynamics of a double pendulum, as shown in the diagram below:



We assume the two point masses m_1 and m_2 are connected by massless rods, with no damping. Linearizing the equations of motion about the equilibrium with both masses hanging down, we obtain the equations of motion

$$\begin{aligned}\ddot{\theta}_1 &= \frac{g(\theta_2 - \mu\theta_1)}{l_1(\mu - 1)} \\ \ddot{\theta}_2 &= \frac{g\mu(\theta_1 - \theta_2)}{l_2(\mu - 1)},\end{aligned}$$

where

$$\mu = \frac{m_1 + m_2}{m_2}$$

is the mass ratio (see, e.g., [21] for a derivation). Writing these in state-space form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, where the state vector is $\mathbf{x} = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$, we have

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{g\mu}{l_1(\mu-1)} & \frac{g}{l_1(\mu-1)} & 0 & 0 \\ \frac{g\mu}{l_2(\mu-1)} & -\frac{g\mu}{l_2(\mu-1)} & 0 & 0 \end{bmatrix}.$$

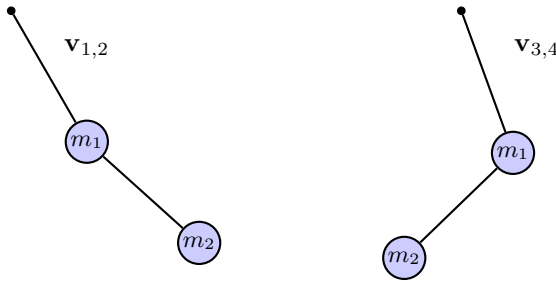
With $\mu = 3$, $g = 9.8$, $l_1 = 0.18$, and $l_2 = 0.15$, the eigenvalues of \mathbf{A} are

$$\lambda_{1,2} = \pm 6.2i, \quad \lambda_{3,4} = \pm 12i,$$

with corresponding eigenvectors

$$\mathbf{v}_{1,2} = \begin{bmatrix} 1.0 \\ 1.6 \\ \pm 6.2i \\ \pm 9.9i \end{bmatrix}, \quad \mathbf{v}_{3,4} = \begin{bmatrix} 1.0 \\ -2.3 \\ \pm 12.0i \\ \mp 27.5i \end{bmatrix}.$$

The corresponding “mode shapes” are shown in the diagram below:



These are simply the real parts of the eigenvectors, so if the initial condition is one of these states, the pendulum will oscillate at 6.2 rad/sec for the left picture, or 12 rad/sec for the right picture. Note that the left picture corresponds to a slow oscillation in which the two pendula are in phase with one another, while the right picture corresponds to a fast oscillation in which the pendula are out of phase with one another. Any initial condition may be represented as a linear combination of these eigenvectors, so the general (linearized) response is a superposition of these two modes of oscillation. \diamond

It is useful to compare (5.12) to our expression (3.2) for the impulse response. Both (5.12) and (3.2) involve sums of exponential terms like $e^{\lambda_j t}$ (where λ_j is an eigenvalue of \mathbf{A}) or $e^{p_j t}$ (where p_j is a pole of the transfer function). It turns out that these are, in fact, the same thing!

Eigenvalues of \mathbf{A} are *poles* of the transfer function

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D.$$

To see why this is the case, consider what happens to the above formula for $G(s)$ if s is an eigenvalue of \mathbf{A} : then $\det(s\mathbf{I} - \mathbf{A}) = 0$, so $s\mathbf{I} - \mathbf{A}$ is not invertible, and $G(s)$ “blows up.” We obtained (3.2) through a partial fraction expansion of the transfer function, while we obtained (5.12) through considering eigenvalues and eigenvectors of \mathbf{A} . Nevertheless, they give us the same result, namely that the transient response is the sum of exponentials e^{pt} , where p is a pole (or eigenvalue of \mathbf{A}).

5.3 Realizations

If we are given a state-space realization (5.1), we know how to construct a transfer function. We simply apply the formula (2.11):

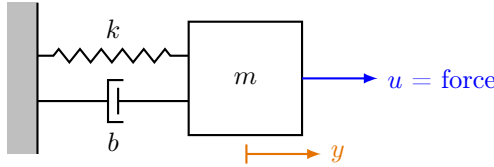
$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D.$$

But suppose we are given a transfer function $G(s)$. Can we construct an equivalent state-space realization? The answer is yes, as long as the transfer

function is *proper*. Recall that a transfer function is proper if the number of poles is greater than or equal to the number of zeros (that is, the relative degree $r \geq 0$). If the transfer function is *strictly proper* (number of poles strictly greater than the number of zeros), then the D matrix in the state-space realization is zero.

How can one construct such a state-space realization? There are systematic procedures for this, but first we illustrate with an example.

Example 5.5 (Spring/mass). Consider a spring-mass system where the mass is subjected to a force $u(t)$ as shown in the diagram below:



If the output $y(t)$ is the position of the mass, the dynamics are given by

$$m\ddot{y} + b\dot{y} + ky = u.$$

The transfer function from u to y is then

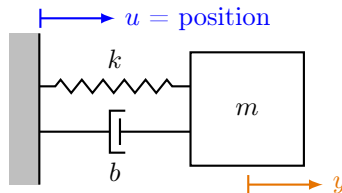
$$\frac{Y(s)}{U(s)} = \frac{1}{ms^2 + bs + k}.$$

It is also straightforward to obtain a state-space realization for this system. Choosing the state $(x_1, x_2) = (y, \dot{y})$, we have

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1/m \end{bmatrix}}_{\mathbf{B}} u \\ y &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{0}_{\mathbf{D}} u \end{aligned}$$

◇

Example 5.6 (Spring/mass forced by moving wall). Now consider a spring-mass system where now the input $u(t)$ is the position of the wall, as in the diagram below:



The equations of motion are now given by

$$\begin{aligned} m\ddot{y} &= -k(y - u) - b(\dot{y} - \dot{u}) \\ m\ddot{y} + b\dot{y} + ky &= ku + b\dot{u}. \end{aligned} \quad (5.13)$$

It is straightforward to calculate the transfer function: we simply take the Laplace transform and solve for the output $Y(s)$, to obtain

$$\frac{Y(s)}{U(s)} = \frac{bs + k}{ms^2 + bs + k}.$$

But how do we write (5.13) in state-space form? If we try the method of the previous example, with the state $\mathbf{x} = (y, \dot{y})$, then there is no way to include the term $b\dot{u}$: a state-space realization can have terms involving u , but not \dot{u} . We appear to be stuck, and need an alternative method. \diamond

One way of determining a state-space realization for Example 5.6 is to use *controllable canonical form*, described below.

Controllable canonical form

For a transfer function

$$G(s) = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0} + d, \quad (5.14)$$

a state-space realization is

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \\ y &= [b_0 \quad b_1 \quad \cdots \quad b_{n-1}] \mathbf{x} + d u. \end{aligned} \quad (5.15)$$

This particular realization is called the *controllable canonical form* (or *reachable canonical form*, or *companion form*).

Where does this come from? Consider a small example: for $n = 2$, the transfer function (5.14) gives

$$Y(s) = \frac{b_1s + b_0}{s^2 + a_1s + a_0}U(s) + dU(s),$$

where $U(s)$ is the Laplace transform of the input $u(t)$, and $Y(s)$ is the Laplace transform of the output $y(t)$. Let us introduce a new variable $X(s)$ that satisfies

$$X(s) = \frac{1}{s^2 + a_1s + a_0}U(s),$$

Exercises

5.1. Consider a linear time-invariant (LTI) system of the form (5.1a).

- (a) Suppose $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ are solutions to (5.1a) with zero input ($\mathbf{u} = 0$), and with initial conditions

$$\mathbf{x}_1(0) = \mathbf{v}_1$$

$$\mathbf{x}_2(0) = \mathbf{v}_2.$$

Show that if the initial state is $\mathbf{x}(0) = k_1\mathbf{v}_1 + k_2\mathbf{v}_2$, then the solution is $\mathbf{x}(t) = k_1\mathbf{x}_1(t) + k_2\mathbf{x}_2(t)$.

- (b) Now suppose $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ are solutions to (5.1a) with zero initial condition ($\mathbf{x}_1(0) = \mathbf{x}_2(0) = \mathbf{0}$) and with inputs $\mathbf{u}_1(t)$ and $\mathbf{u}_2(t)$, respectively. Show that if the input is $\mathbf{u}(t) = k_1\mathbf{u}_1(t) + k_2\mathbf{u}_2(t)$, then the solution is $\mathbf{x}(t) = k_1\mathbf{x}_1(t) + k_2\mathbf{x}_2(t)$.

5.2. An LTI system with two states and one output has the following zero-input responses.

- For $\mathbf{x}(0) = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$, $y(t) = 3e^{-3t} - e^{2t}$.

- For $\mathbf{x}(0) = \begin{bmatrix} -2 \\ 6 \end{bmatrix}$, $y(t) = 3e^{-3t} + e^{2t}$.

- (a) Using the property of linearity and superposition, calculate the zero-input response $y(t)$ if $\mathbf{x}(0) = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$.
- (b) Is this LTI system stable?
- (c) If the system were to be written in state-space form, determine what the eigenvalues and eigenvectors of \mathbf{A} would be. Which eigenvector corresponds to which eigenvalue?

5.3. Prove that, if \mathbf{T} is an invertible matrix, then $e^{\mathbf{T}^{-1}\mathbf{A}\mathbf{T}t} = \mathbf{T}^{-1}e^{\mathbf{A}t}\mathbf{T}$.

5.4. Recall that a matrix \mathbf{A} is *skew-symmetric* if $\mathbf{A}^\top = -\mathbf{A}$, and a matrix \mathbf{M} is *orthogonal* if $\mathbf{M}^\top = \mathbf{M}^{-1}$. Show that if \mathbf{A} is skew-symmetric, then $e^{\mathbf{A}t}$ is orthogonal.

5.5. Let \mathbf{J} be the matrix defined by

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix},$$

where \mathbf{I} denotes the $n \times n$ identity matrix. A $2n \times 2n$ matrix \mathbf{A} is called *Hamiltonian* if $\mathbf{J}\mathbf{A} + \mathbf{A}^\top\mathbf{J} = \mathbf{0}$, and a matrix \mathbf{M} is called *symplectic* if $\mathbf{M}^\top\mathbf{J}\mathbf{M} = \mathbf{J}$. Show that if \mathbf{A} is Hamiltonian, then $e^{\mathbf{A}t}$ is symplectic.

5.6. Compute $e^{\mathbf{A}t}$ for

$$\mathbf{A} = \begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}.$$

5.7. Look up what algorithm MATLAB uses to calculate the matrix exponential, with the `expm` command. Is this one of the algorithms discussed in [20]?

5.8. Consider a state-space system of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}.$$

Define a new state \mathbf{z} , such that $\mathbf{x} = \mathbf{T}\mathbf{z}$ for some invertible matrix \mathbf{T} . Prove that the transfer function for the state space system in \mathbf{x} is the same as the transfer function for the state space system in \mathbf{z} .

5.9. Given a polynomial

$$p(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0,$$

the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}$$

is called its *companion matrix* [15, §3.3]. Show that $p(s)$ is the characteristic polynomial of \mathbf{A} , and thus of its transpose

$$\mathbf{A}^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}$$

5.10. Show that another realization of (5.14) is

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix} \mathbf{x} + \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} u \\ y &= [0 \quad 0 \quad \cdots \quad 0 \quad 1] \mathbf{x} + d u. \end{aligned} \quad (5.17)$$

This realization is called *observable canonical form*.

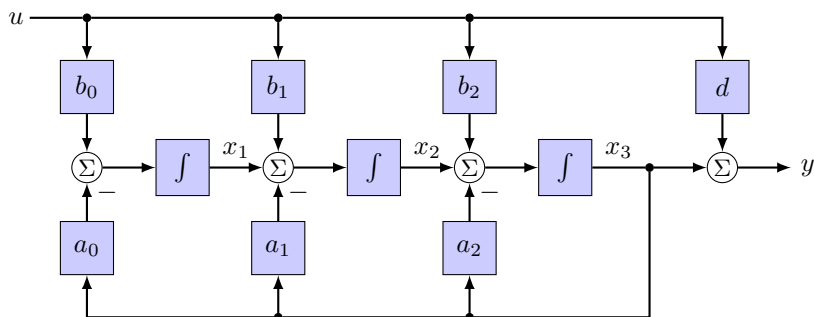


Figure 5.2: Block diagram for Exercise 5.11.

- 5.11.** Determine the transfer function from u to y in Figure 5.2 (see also Exercise 2.13). Then, using the state variables x_1 , x_2 , and x_3 as labeled, find the corresponding state-space realization, with the state $\mathbf{x} = (x_1, x_2, x_3)$. Do you recognize this as one of the canonical forms? If so, which one?
- 5.12.** Compute a state-space realization (of your choice) for the transfer function

$$G(s) = \frac{s+1}{(s+2)(s+3)}.$$

Verify that the eigenvalues of your \mathbf{A} matrix are the poles of $G(s)$.

Chapter 6

State feedback

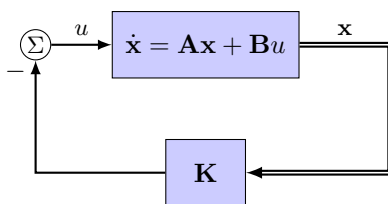
This chapter focuses on a particular type of feedback law commonly used for a state-space system of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \quad (6.1)$$

known as a linear time-invariant (LTI) system. In particular, we choose a feedback law

$$u = -\mathbf{K}\mathbf{x}, \quad \mathbf{K} = [k_1 \quad \cdots \quad k_n],$$

where k_1, \dots, k_n are parameters (the “gains”) we choose in designing our controller. This choice of feedback is called *state feedback*, and corresponds to the block diagram shown below:



In this diagram, the double line is meant to indicate that the state \mathbf{x} is a vector, while u is a scalar. Many of the techniques described in this chapter extend naturally to the multiple-input case, in which the input u is also a vector, but we will focus on the single-input case. We begin with the important concept of *controllability* (or *reachability*), and then discuss some powerful techniques for choosing the gains \mathbf{K} , in particular using pole placement (§6.2) and an optimal control method called the Linear Quadratic Regulator (LQR, §6.5).

6.1 Controllability

Controllability refers to the ability of the input $u(t)$ to control the state $\mathbf{x}(t)$. We will soon introduce a more precise definition, but first we illustrate the idea with an example.

Example 6.1. Consider a linear system described by $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & -2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The poles of the system (eigenvalues of the \mathbf{A} matrix) are 1 and -2 (this can be seen immediately, since the \mathbf{A} matrix is upper triangular), so because one of these is in the right half-plane, the system is unstable. We wish to stabilize the system with the state feedback law

$$u = -\mathbf{K}\mathbf{x}, \quad \mathbf{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}.$$

Let us look at the poles of the closed-loop system. The closed-loop system is given by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}(-\mathbf{K}\mathbf{x}) \\ &= (\mathbf{A} - \mathbf{BK})\mathbf{x}, \end{aligned}$$

so the closed-loop poles are eigenvalues of $\mathbf{A} - \mathbf{BK}$. We find

$$\mathbf{BK} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix},$$

so the closed-loop poles are where

$$\det(s\mathbf{I} - (\mathbf{A} - \mathbf{BK})) = \det \begin{bmatrix} s-1 & -1 \\ k_1 & s+2+k_2 \end{bmatrix} = 0,$$

which gives

$$s^2 + s(1+k_2) - 2 + k_1 - k_2 = 0.$$

Note that, with suitable choice of k_1 and k_2 , we can choose the two coefficients in this equation to be anything we like. For instance, if we desire the closed-loop poles to be -1 and -2 , then the *desired* characteristic polynomial is

$$p(s) = (s+1)(s+2) = s^2 + 3s + 2,$$

so this can be achieved by setting $k_2 = 2$, $k_1 = 6$. ◇

This example illustrates a technique called *pole placement*, in which we select *desired* closed-loop poles, and then solve for the corresponding gains that place the actual closed-loop poles there. Is this always possible? Let us look at another example.

Example 6.2. Consider the same example as before, but now with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The closed-loop dynamics are again described by $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x}$, but now

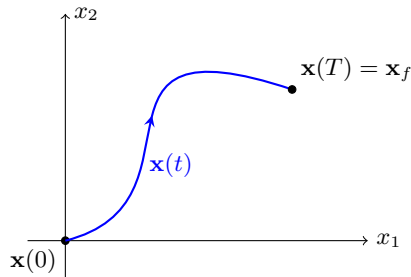
$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 1 & 0 \\ -k_1 & -2 - k_2 \end{bmatrix}.$$

The closed-loop eigenvalues are thus 1 and $-2 - k_2$ (we can read these directly off the diagonal, since this matrix is lower triangular), so we see *we cannot change the pole at +1*: no matter what gains k_1, k_2 we pick, there will always be a closed-loop pole at $s = +1$. This is particularly troublesome, since this pole is in the right half-plane. Thus, it is not even possible to stabilize the system with state feedback. \diamond

We would like a way to characterize when things can “go wrong” as in the previous example. The key property that determines this is *controllability*, for which we now give a more precise definition.

Definition 6.3. The LTI system $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ is *controllable* (or *reachable*) if, for every \mathbf{x}_f and every $T > 0$, there exists an input function $\mathbf{u}(t)$, for $0 < t < T$, such that the system state is taken from $\mathbf{0}$ at $t = 0$ to \mathbf{x}_f at $t = T$.

A system that is not controllable is called *uncontrollable*.



Let us take a closer look at the definition above. It says that, for a system to be controllable, we need to be able to drive the state from the origin at $t = 0$ to any given final state \mathbf{x}_f at time $t = T$. This may seem to be too strict a requirement: after all, the final state \mathbf{x}_f could be chosen to be very large, and the time T could be very small, placing a possibly unreasonable demand on the system. (For instance, it does not sound reasonable to require a spaceship to be able to fly from the Earth to the moon in 0.1 sec.) However, this requirement is actually okay, because we have assumed we have a linear system (6.1). Large final states \mathbf{x}_f and small destination times T will result in large inputs $u(t)$, but otherwise are not a problem.

At first, this definition does not look at all related to the problem that we encountered in Example 6.2: for instance, it says nothing about state feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$ or assigning poles of the closed-loop system. However, we will see in the next section that as long as the system is controllable, in the sense of the above definition, we will be able to place the closed-loop poles wherever we like with state feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$.

It would be helpful to have a convenient test for when a given system is controllable. There are in fact several such tests, as we shall see, but the most common one is the following.

The state-space system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, with $\mathbf{x} \in \mathbb{R}^n$, is controllable if and only if the *controllability matrix*

$$\mathbf{C} := [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (6.2)$$

has full rank ($\text{rank} = n$).

We prove this result later in this section (see Theorem 6.8; also see standard texts such as [12, §5.4] and [9, §2.2]), but first we apply it to some examples. In the controllability test above, the input $\mathbf{u}(t)$ may be a vector (in which case the matrix \mathbf{C} is *rectangular*, not square), although here we will usually be concerned with the single-input case in which u is a scalar, and hence \mathbf{B} is a single column and \mathbf{C} is square.

Let us see what this controllability test says about the examples considered previously.

Example 6.4. Consider again the systems from Examples 6.1 and 6.2. In Example 6.1, the controllability matrix is

$$\mathbf{C} = [\mathbf{B} \quad \mathbf{A}\mathbf{B}] = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix},$$

which has $\text{rank } 2 = n$, so the system is controllable. However, in Example 6.2, the controllability matrix is

$$\mathbf{C} = [\mathbf{B} \quad \mathbf{A}\mathbf{B}] = \begin{bmatrix} 0 & 0 \\ 1 & -2 \end{bmatrix},$$

which has $\text{rank } 1 < n$, so the system is not controllable. \diamond

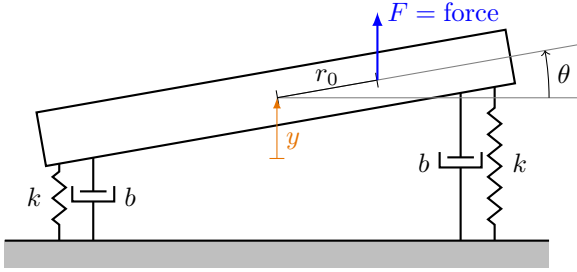
Reachable subspace

The matrix \mathbf{C} actually gives us more information than simply whether or not a given system is controllable: it also tells us precisely which states \mathbf{x}_f can be reached with an input $u(t)$ as described in Definition 6.3, with $\mathbf{x}(0) = \mathbf{0}$. In

particular, the set of all such states \mathbf{x}_f forms a subspace, called the *reachable subspace*, and it is the space spanned by the columns of \mathcal{C} (called the *range* or *image* of \mathcal{C}).

The following example illustrates this idea with a mechanical system.

Example 6.5 (Sprung beam). Consider a beam of mass m and length $2L$ supported by springs at each end, as shown in the diagram below:



Here, y is the height of the center of mass of the beam (relative to its rest position), θ is the angle of the beam relative to the horizontal, and there is a force $F(t)$ applied a distance r_0 from the center of mass. We assume the springs and dampers are located at the ends of the beam, a distance L from the center of mass, and the moment of inertia about the center of mass is J_0 . The equations of motion, linearized about $\theta = 0$, are given by

$$\begin{aligned} m\ddot{y} &= -2ky - 2b\dot{y} + F \\ J_0\ddot{\theta} &= -2kL^2\theta - 2bL^2\dot{\theta} + r_0F. \end{aligned}$$

We may simplify these equations and reduce the number of parameters by nondimensionalizing. Defining dimensionless variables $\tilde{y} = y/L$ and $\tilde{t} = t\sqrt{2k/m}$ (normalized by the undamped natural frequency of the beam in the y -direction), and defining the dimensionless quantities

$$r = \frac{r_0}{L}, \quad J = \frac{J_0}{mL^2}, \quad \delta = \frac{b}{\sqrt{km}/2}, \quad u = \frac{F}{2kL},$$

the dimensionless equations become

$$\begin{aligned} \ddot{\tilde{y}} &= -\tilde{y} - \delta\dot{\tilde{y}} + u \\ J\ddot{\theta} &= -\theta - \delta\dot{\theta} + r\tilde{y}, \end{aligned}$$

where now $(\dot{})$ denotes $d/d\tilde{t}$. In state-space form, then, the equations become

$$\frac{d}{d\tilde{t}} \begin{bmatrix} \tilde{y} \\ \theta \\ \dot{\tilde{y}} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -\delta & 0 \\ 0 & -1/J & 0 & -\delta/J \end{bmatrix} \begin{bmatrix} \tilde{y} \\ \theta \\ \dot{\tilde{y}} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ r/J \end{bmatrix} u.$$

Is this system controllable? Let us first try to use our intuition. It seems clear that, if $r = 0$, then the system should not be controllable, for if the beam starts from rest, with $\theta = 0$, and if the (nondimensional) force u is at the center of mass ($r = 0$), then there will be no way to drive the beam to a nonzero angle. What if $r \neq 0$? Then, the situation is not so clear. It does seem like a challenging task to find an input $u(t)$ to drive the beam from rest to any desired final state $(y, \theta, \dot{y}, \dot{\theta})$, and it is not intuitively clear when (if ever) this might be possible.

Let us see what our controllability test tells us. In particular, consider the case where there is no damping, for $\delta = 0$ (the same conclusions hold for $\delta \neq 0$: see Exercise 6.2). Then the controllability matrix is

$$\mathbf{C} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \mathbf{A}^3\mathbf{B}].$$

We have

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ r/J \end{bmatrix}, \quad \mathbf{AB} = \begin{bmatrix} 1 \\ r/J \\ 0 \\ 0 \end{bmatrix}.$$

For $\mathbf{A}^2\mathbf{B}$, note that we have already calculated \mathbf{AB} , so it is much easier to calculate this as $\mathbf{A}(\mathbf{AB})$ rather than doing the matrix multiplication \mathbf{A}^2 and multiplying this by \mathbf{B} . We find

$$\mathbf{A}^2\mathbf{B} = \mathbf{A}(\mathbf{AB}) = \begin{bmatrix} 0 \\ 0 \\ -1 \\ -r/J^2 \end{bmatrix}, \quad \mathbf{A}^3\mathbf{B} = \mathbf{A}(\mathbf{A}^2\mathbf{B}) = \begin{bmatrix} -1 \\ -r/J^2 \\ 0 \\ 0 \end{bmatrix}.$$

Altogether, then,

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & r/J & 0 & -r/J^2 \\ 1 & 0 & -1 & 0 \\ r/J & 0 & -r/J^2 & 0 \end{bmatrix}.$$

Because \mathbf{C} is square, one way to check that it is full rank (rank = 4) is to check that its determinant is nonzero. (Recall that a square matrix is full rank if and only if it is invertible.) We find

$$\det \mathbf{C} = -\frac{r^2}{J^4}(J-1)^2,$$

and the system is controllable if and only if $\det \mathbf{C} \neq 0$. We see right away that if $r = 0$, then $\det \mathbf{C} = 0$, and thus the system is not controllable, as our intuition predicted. The only other way the system can be uncontrollable is if $J = 1$, or (in the dimensional variables) $J_0 = mL^2$. This represents a

particular relationship between the moment of inertia J and the mass and length of the beam: in particular, all the mass is concentrated at the ends of the beam. If the distribution of mass is such that $J = 1$, then the system will be uncontrollable. Otherwise, it will be controllable.

What is significant about this condition $J = 1$? Can we understand this intuitively? If $u = 0$ and $\theta = \dot{\theta} = 0$, then the y dynamics become $\ddot{y} + y = 0$, which is an undamped oscillator with natural frequency 1. On the other hand, if $y = \dot{y} = 0$, then the θ dynamics become $J\ddot{\theta} + \theta = 0$, an undamped oscillator with natural frequency $\sqrt{1/J}$. So if $J = 1$, then the two frequencies of oscillation (corresponding to bouncing in y , and rocking in θ) are the same. This extra “symmetry” then interferes with the controllability. When these two frequencies coincide, no matter what input $u(t)$ we apply, the rocking in θ will be perfectly in phase with the bouncing in y (when the beam starts from rest), and we cannot control these independently.

We can see this mathematically by looking at the reachable subspace, the space spanned by the columns of \mathcal{C} . When $J = 1$, the third and fourth columns of \mathcal{C} are just -1 times the first and second columns, respectively, so the controllable subspace is spanned by the first two columns. In particular, this means

$$\begin{bmatrix} \tilde{y} \\ \theta \\ \dot{\tilde{y}} \\ \dot{\theta} \end{bmatrix} = c_1(t) \begin{bmatrix} 0 \\ 0 \\ 1 \\ r \end{bmatrix} + c_2(t) \begin{bmatrix} 1 \\ r \\ 0 \\ 0 \end{bmatrix},$$

for some scalars $c_1(t), c_2(t)$, and therefore

$$\theta(t) = r\tilde{y}(t).$$

Thus, the angle and height are always in phase, and cannot be varied independently of one another. \diamond



Uncontrollable states and the controllability Gramian

We now present additional ways to characterize and test for controllability, and use these to prove the controllability test given on page 166. These concepts are somewhat more advanced than those in the rest of this section, and may be skipped if desired.

We consider a system with multiple inputs (i.e., $\mathbf{u}(t)$ may be a vector), starting at the origin:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{0}, \quad (6.3)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$. Recall from (5.8) that the solution is given by

$$\mathbf{x}(t) = \int_0^t e^{\mathbf{A}\tau} \mathbf{B}\mathbf{u}(t - \tau) d\tau. \quad (6.4)$$

We now introduce the notion of an uncontrollable *state*. (Note that this is a different concept from an uncontrollable *system*, as in Definition 6.3.)

Definition 6.6 (Uncontrollable state). A state $\mathbf{x}_0 \neq \mathbf{0}$ is *uncontrollable* if the solution $\mathbf{x}(t)$ of (6.3) is orthogonal to \mathbf{x}_0 for all $t > 0$, and for all input functions $\mathbf{u}(t)$.

That is, an uncontrollable state is orthogonal to the *reachable subspace*, the set of all states that can be reached by an arbitrary input, starting from the origin. (Recall that two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ are *orthogonal* if their dot product is zero, i.e., $\mathbf{v}^\top \mathbf{w} = 0$.) The following theorem establishes conditions for \mathbf{x}_0 to be an uncontrollable state.

Theorem 6.7. *The following are equivalent:*

1. *The system (6.3) has an uncontrollable state \mathbf{x}_0 .*
2. *$\mathbf{x}_0^\top e^{\mathbf{A}t} \mathbf{B} = \mathbf{0}$ for all t .*
3. *$\mathbf{x}_0^\top e^{\mathbf{A}t} \mathbf{B} = \mathbf{0}$ for all t in some open interval (a, b) .*

Proof. First, from (6.4) we see that the dot product of \mathbf{x}_0 with $\mathbf{x}(t)$ is given by

$$\mathbf{x}_0^\top \mathbf{x}(t) = \int_0^t \mathbf{x}_0^\top e^{\mathbf{A}\tau} \mathbf{B} \mathbf{u}(t - \tau) d\tau, \quad (6.5)$$

from which it is clear that (2) \implies (1).

Next, to prove that (1) \implies (3), we will show that if \mathbf{x}_0 is an uncontrollable state, then

$$\mathbf{v}(t) := (\mathbf{x}_0^\top e^{\mathbf{A}t} \mathbf{B})^\top = \mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0$$

is zero for all $t \geq 0$. Suppose, to the contrary, that $\mathbf{v}(t) \neq \mathbf{0}$ for some $t = t_0 \geq 0$. Then, since \mathbf{v} is a continuous function of t , there is an $\varepsilon > 0$ such that $\mathbf{v}(t) \neq \mathbf{0}$ for all t in an interval $(t_0, t_0 + \varepsilon)$. Let $T = t_0 + \varepsilon$ and construct an input function $\mathbf{u}(t)$ such that

$$\mathbf{u}(T - t) = \begin{cases} \mathbf{v}(t), & t \in (t_0, t_0 + \varepsilon) \\ 0, & t \notin (t_0, t_0 + \varepsilon). \end{cases}$$

Then from (6.5), for this input \mathbf{u} ,

$$\begin{aligned} \mathbf{x}_0^\top \mathbf{x}(T) &= \int_{t_0}^{t_0 + \varepsilon} \mathbf{x}_0^\top e^{\mathbf{A}\tau} \mathbf{B} \mathbf{v}(\tau) d\tau \\ &= \int_{t_0}^{t_0 + \varepsilon} \mathbf{v}(\tau)^\top \mathbf{v}(\tau) d\tau = \int_{t_0}^{t_0 + \varepsilon} \|\mathbf{v}(\tau)\|^2 d\tau > 0. \end{aligned}$$

Hence, \mathbf{x}_0 is not orthogonal to $\mathbf{x}(T)$, so \mathbf{x}_0 is not an uncontrollable state, a contradiction. Thus we must have $\mathbf{v}(t) = \mathbf{0}$ for all $t \geq 0$.

Finally, to show (3) \implies (2), note that, from the definition of the matrix exponential in (5.4), $\mathbf{x}_0^\top e^{\mathbf{A}t} \mathbf{B}$ has a power series expansion

$$\mathbf{x}_0^\top e^{\mathbf{A}t} \mathbf{B} = \sum_{k=0}^{\infty} \mathbf{x}_0^\top \mathbf{A}^k \mathbf{B} \frac{t^k}{k!}, \quad (6.6)$$

which converges for all t . Thus, if this quantity is zero in any open interval (a, b) , then all its derivatives must be zero in this interval, and thus every term in the power series must be zero, so the function must be zero for all t . (This is an example of analytic continuation, discussed in Appendix B.2.) \square

Next, we define another matrix that can be even more useful than the controllability matrix \mathbf{C} . The *controllability Gramian* is an $n \times n$ matrix defined by

$$\mathbf{W}_c(T) := \int_0^T e^{\mathbf{A}t} \mathbf{B} \mathbf{B}^\top e^{\mathbf{A}^\top t} dt. \quad (6.7)$$

Observe that $\mathbf{W}_c(T)$ is a square matrix even if the system (6.3) has more than one input (i.e., \mathbf{B} has more than one column), while the controllability matrix \mathbf{C} given by (6.2) is rectangular if there is more than one input. It is easy to see that the matrix $\mathbf{W}_c(T)$ is symmetric, and furthermore it is positive semi-definite, since for any \mathbf{x}_0 ,

$$\begin{aligned} \mathbf{x}_0^\top \mathbf{W}_c(T) \mathbf{x}_0 &= \int_0^T \mathbf{x}_0^\top e^{\mathbf{A}t} \mathbf{B} \mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0 dt \\ &= \int_0^T (\mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0)^\top \mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0 dt \\ &= \int_0^T \|\mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0\|^2 dt \geq 0. \end{aligned} \quad (6.8)$$

We now present the main result of this section.

Theorem 6.8 (Controllability). *The following are equivalent:*

1. The system (6.3) is controllable.
2. The system (6.3) has no uncontrollable states.
3. The controllability Gramian $\mathbf{W}_c(T)$ defined by (6.7) is positive definite for all $T > 0$.
4. The controllability matrix \mathbf{C} defined by (6.2) is full rank (rank = n).

Proof. To show all the required equivalences, we first prove (1) \implies (2) \implies (3) \implies (1), and then show (2) \iff (4).

To see that (1) \implies (2), consider the contrapositive: if (6.3) does have an uncontrollable state $\mathbf{x}_0 \neq \mathbf{0}$, then it is not possible to find an input $\mathbf{u}(t)$ to drive the state from $\mathbf{0}$ to \mathbf{x}_0 (for any time T), so the system is not controllable.

To see that (2) \implies (3), again consider the contrapositive, and suppose there is a $T > 0$ for which $\mathbf{W}_c(T)$ is not positive definite (that is, it is only positive semi-definite). Then there is some \mathbf{x}_0 such that $\mathbf{x}_0^\top \mathbf{W}_c(T) \mathbf{x}_0 = 0$. We will show this \mathbf{x}_0 is an uncontrollable state. Indeed, from the calculation in (6.8), we see

$$\mathbf{x}_0^\top \mathbf{W}_c(T) \mathbf{x}_0 = \int_0^T \|\mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0\|^2 dt,$$

so since the integrand is non-negative, the only way this can be zero is if $\mathbf{B}^\top e^{\mathbf{A}^\top t} \mathbf{x}_0 = \mathbf{0}$ for all $t \in [0, T]$. Thus, by Theorem 6.7, \mathbf{x}_0 is an uncontrollable state.

To show that (3) \implies (1), we explicitly construct an input $\mathbf{u}(t)$ to drive the state from $\mathbf{0}$ to any target state \mathbf{x}_f at time T . Since $\mathbf{W}_c(T)$ is positive definite, it is invertible, so we may define the input

$$\mathbf{u}(t) = \mathbf{B}^\top e^{\mathbf{A}^\top (T-t)} \mathbf{W}_c(T)^{-1} \mathbf{x}_f. \quad (6.9)$$

Evaluating (6.4) for this input then gives

$$\mathbf{x}(T) = \int_0^T e^{\mathbf{A}^\top \tau} \mathbf{B} \mathbf{B}^\top e^{\mathbf{A}^\top \tau} \mathbf{W}_c(T)^{-1} \mathbf{x}_f d\tau = \mathbf{W}_c(T) \mathbf{W}_c(T)^{-1} \mathbf{x}_f = \mathbf{x}_f,$$

as desired, so the system is controllable.

Next, to show (4) \implies (2), suppose that the system (6.3) does have an uncontrollable state $\mathbf{x}_0 \neq \mathbf{0}$; we wish to show that $\text{rank } \mathcal{C} < n$. By Theorem 6.7, we know $\mathbf{x}_0^\top e^{\mathbf{A}^\top t} \mathbf{B} = \mathbf{0}$ for all t . Evaluating this expression at $t = 0$, we find $\mathbf{x}_0^\top \mathbf{B} = \mathbf{0}$. Since this expression is zero for all t , its derivative at $t = 0$ must also be zero, and so

$$\frac{d}{dt} (\mathbf{x}_0^\top e^{\mathbf{A}^\top t} \mathbf{B}) \Big|_{t=0} = \mathbf{x}_0^\top e^{\mathbf{A}^\top 0} \mathbf{A} \mathbf{B} = \mathbf{x}_0^\top \mathbf{A} \mathbf{B} = \mathbf{0}.$$

Furthermore, the second derivative must be zero, which implies $\mathbf{x}_0^\top \mathbf{A}^2 \mathbf{B}$, and indeed all the derivatives must be zero, so $\mathbf{x}_0^\top \mathbf{A}^k \mathbf{B} = \mathbf{0}$ for $k = 0, 1, 2, \dots$. Collecting these expressions in matrix form for $k = 0, \dots, n-1$ then gives

$$\mathbf{x}_0^\top [\mathbf{B} \quad \mathbf{A} \mathbf{B} \quad \dots \quad \mathbf{A}^{n-1} \mathbf{B}] = \mathbf{x}_0^\top \mathcal{C} = \mathbf{0}.$$

Thus, \mathbf{x}_0 is a nonzero vector orthogonal to the range of \mathcal{C} , so the range of \mathcal{C} cannot be all of \mathbb{R}^n , and hence the rank of \mathcal{C} must be less than n .

Finally, to show (2) \implies (4), we again show the contrapositive: we assume \mathcal{C} is not full rank, and find an uncontrollable state \mathbf{x}_0 . Since the range of \mathcal{C} is not all of \mathbb{R}^n , there is a nonzero vector $\mathbf{x}_0 \in \mathbb{R}^n$ such that $\mathbf{x}_0^\top \mathcal{C} = \mathbf{0}$. That is,

$$\mathbf{x}_0^\top \mathbf{B} = \mathbf{0}, \quad \mathbf{x}_0^\top \mathbf{A} \mathbf{B} = \mathbf{0}, \quad \dots, \quad \mathbf{x}_0^\top \mathbf{A}^{n-1} \mathbf{B} = \mathbf{0}.$$

By the Cayley-Hamilton theorem (Corollary A.7), we may write \mathbf{A}^n as a linear combination of $\mathbf{A}^0, \dots, \mathbf{A}^{n-1}$, from which it follows $\mathbf{x}_0^\top \mathbf{A}^n \mathbf{B} = \mathbf{0}$, and proceeding recursively, we find $\mathbf{x}_0^\top \mathbf{A}^k \mathbf{B} = \mathbf{0}$ for all $k = 0, 1, 2, \dots$. Thus, all the terms in the power series expansion (6.6) are zero, and thus \mathbf{x}_0 is an uncontrollable state. \square

An interesting feature of the proof above is that we explicitly constructed an input (6.9) that drives the state from the origin to \mathbf{x}_f . This input function is not unique: if a system is controllable, there are many choices of $\mathbf{u}(t)$ that will drive the state to \mathbf{x}_f . But it turns out that the particular choice given by (6.9) is the *minimum-energy* input that accomplishes this: that is,

$$\|\mathbf{u}\|^2 := \int_0^T \mathbf{u}(t)^\top \mathbf{u}(t) dt$$

is the smallest, over all input functions \mathbf{u} that drive the state to \mathbf{x}_f at time T . In fact, for \mathbf{u} given by (6.9), we have

$$\begin{aligned} \|\mathbf{u}\|^2 &= \int_0^T \mathbf{x}_f^\top \mathbf{W}_c(T)^{-1} e^{\mathbf{A}(T-t)} \mathbf{B} \mathbf{B}^\top e^{\mathbf{A}^\top(T-t)} \mathbf{W}_c(T)^{-1} \mathbf{x}_f dt \\ &= \mathbf{x}_f^\top \mathbf{W}_c(T)^{-1} \mathbf{W}_c(T) \mathbf{W}_c(T)^{-1} \mathbf{x}_f \\ &= \mathbf{x}_f^\top \mathbf{W}_c(T)^{-1} \mathbf{x}_f. \end{aligned}$$

As a consequence, the controllability Gramian $\mathbf{W}_c(T)$ can be used to quantify “how controllable” a particular state is. From the calculation above, the quantity $\mathbf{x}_f^\top \mathbf{W}_c(T)^{-1} \mathbf{x}_f$ tells us how much input energy or actuator effort is required to drive to any given state \mathbf{x}_f . If this quantity is large, then a large input is required to drive the state to \mathbf{x}_f , and if this quantity is extremely large, then it may be extremely difficult in practice to drive the state to \mathbf{x}_f , so we might say that the state is “nearly uncontrollable”. Conversely, if this quantity is small, then a small actuator effort is required to drive the state to \mathbf{x}_f , so such states can be considered “more controllable.”

If the system (6.3) is stable (all eigenvalues of \mathbf{A} in the open left half-plane), then it is often convenient to consider $\mathbf{W}_c(T)$ in the limit as $T \rightarrow \infty$, for which the limit \mathbf{W}_c can be found by solving the linear matrix equation

$$\mathbf{A} \mathbf{W}_c + \mathbf{W}_c \mathbf{A}^\top + \mathbf{B} \mathbf{B}^\top = \mathbf{0},$$

called a Lyapunov equation (see Theorem 6.17).

6.2 Pole placement

In the previous section, we saw a definition of controllability and a useful test for it, but it was not clear how this definition relates to state feedback. Recall that, given a system described by

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} u,$$

with $\mathbf{x} \in \mathbb{R}^n$, state feedback refers to a controller of the form

$$u = -\mathbf{K}\mathbf{x},$$

where $\mathbf{K} = [k_1 \ \cdots \ k_n]$ is a matrix of “gains.” Such a feedback law changes the dynamics of the system—more precisely, it changes the poles. The open-loop poles (without feedback) are simply eigenvalues of \mathbf{A} . The closed-loop system is given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(-\mathbf{K}\mathbf{x}) = (\mathbf{A} - \mathbf{BK})\mathbf{x},$$

so the closed-loop poles are the eigenvalues of $\mathbf{A} - \mathbf{BK}$. If $\mathbf{K} = \mathbf{0}$, then there is no feedback, and these poles are the same as the open-loop poles. As we change the gains \mathbf{K} , we change the eigenvalues of $\mathbf{A} - \mathbf{BK}$. How much flexibility do we have, with different choices of k_1, \dots, k_n to change these eigenvalues? We will see in this section that if the system is controllable, we have quite a bit of flexibility. In fact, we can place the closed-loop poles anywhere we like in the complex plane, with an appropriate choice of \mathbf{K} .

To state this result more precisely, we first give some basic definitions. The *characteristic polynomial* of a matrix \mathbf{M} is

$$p(s) = \det(s\mathbf{I} - \mathbf{M}).$$

If \mathbf{M} is $n \times n$, then $p(s)$ is a polynomial of degree n :

$$p(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0.$$

Furthermore, its leading coefficient is 1: this is called a *monic* polynomial. We can now state the main result of this section.

Theorem 6.9 (Pole placement). *Let $p(s)$ be a monic polynomial of degree n . If the single-input system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ is controllable (where $\mathbf{x} \in \mathbb{R}^n$), then there is a unique gain vector \mathbf{K} such that the characteristic polynomial of $\mathbf{A} - \mathbf{BK}$ is $p(s)$.*

The $p(s)$ of this theorem is the *desired* characteristic polynomial, so the theorem says that as long as the system (6.1) is controllable, we can choose \mathbf{K} such that the closed-loop poles (eigenvalues of $\mathbf{A} - \mathbf{BK}$) match the roots of any desired characteristic polynomial $p(s)$. The proof of this theorem is given at the end of this section (see also [12, §6.2]), but first we give some examples of its use.

Example 6.10. Suppose we want closed-loop poles at $s = -1, -3, -7$. Then the desired characteristic polynomial is

$$p(s) = (s + 1)(s + 3)(s + 7) = s^3 + 11s^2 + 31s + 21. \quad \diamond$$

The theorem suggests the following design procedure for state feedback controllers:

Design procedure for pole placement:

- Decide on desired closed-loop poles.
- Find the corresponding characteristic polynomial $p(s)$.
- Solve for the corresponding gains \mathbf{K} such that

$$\det(s\mathbf{I} - (\mathbf{A} - \mathbf{BK})) = p(s).$$

This procedure is, in fact, exactly what we used to determine the gains \mathbf{K} in Example 6.1. Now we see that it was not just good fortune that we were able to find gains \mathbf{K} to place the poles arbitrarily in that example: in fact, this was guaranteed, as long as the pair (\mathbf{A}, \mathbf{B}) was controllable.

Example 6.11. Consider a state-space realization $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

The open-loop poles are immediately seen to be $s = 1, -2$ (since the \mathbf{A} matrix is diagonal), so the open-loop system is unstable. Let us try to stabilize the system with state feedback $u = -\mathbf{K}\mathbf{x}$, placing the closed-loop poles at $s = -1, -2$.

First, we determine whether this is possible by checking the controllability matrix

$$\mathbf{C} = [\mathbf{B} \quad \mathbf{AB}] = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix}.$$

We find $\det \mathbf{C} = 3 \neq 0$, so the system is controllable, and we can assign the closed-loop poles arbitrarily. Next, we find the desired characteristic polynomial: for the closed-loop poles to be at $-1, -2$, we have

$$p(s) = (s + 1)(s + 2) = s^2 + 3s + 2.$$

Next, we find the actual characteristic polynomial of $\mathbf{A} - \mathbf{BK}$:

$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} = \begin{bmatrix} 1 - k_1 & -k_2 \\ k_1 & -2 + k_2 \end{bmatrix},$$

so the actual characteristic polynomial is

$$\begin{aligned} \det(s\mathbf{I} - (\mathbf{A} - \mathbf{BK})) &= \det \begin{bmatrix} s - 1 + k_1 & k_2 \\ -k_1 & s + 2 - k_2 \end{bmatrix} \\ &= s^2 + (1 + k_1 - k_2)s - 2 + 2k_1 + k_2. \end{aligned}$$

Finally, we equate coefficients between the actual and desired characteristic polynomials, to get

$$\begin{aligned} 1 + k_1 - k_2 &= 3 \\ -2 + 2k_1 + k_2 &= 2. \end{aligned}$$

Adding these equations, we get $3k_1 = 6$, so $k_1 = 2$; substituting this into the first equation then gives $k_2 = 0$, so overall the gains are

$$\mathbf{K} = \begin{bmatrix} 2 & 0 \end{bmatrix}. \quad \diamond$$

In MATLAB, the gains \mathbf{K} can be determined with the command

```
K = place(A,B,p)
```

where \mathbf{p} is a vector of desired closed-loop pole locations.



Proof of the pole placement theorem

We now prove the pole placement theorem, Theorem 6.9. The idea of the proof is to first change coordinates so that the system is in controllable canonical form (5.15); in these coordinates, the result is then obvious. We first establish the following lemma, which shows that any controllable single-input system can, in fact, be transformed into controllable canonical form with a suitable change of coordinates.

Lemma 6.12. *Suppose two single-input LTI systems given by*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \quad \dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}u$$

are both controllable, and suppose that \mathbf{A} and $\tilde{\mathbf{A}}$ have the same characteristic polynomial. Then the two systems are related by a change of coordinates $\mathbf{x} = \mathbf{T}\tilde{\mathbf{x}}$. In particular, letting \mathbf{C} and $\tilde{\mathbf{C}}$ denote the controllability matrices of the two systems, and defining $\mathbf{T} = \mathbf{C}\tilde{\mathbf{C}}^{-1}$, then

$$\mathbf{A} = \mathbf{T}\tilde{\mathbf{A}}\mathbf{T}^{-1}, \quad \mathbf{B} = \mathbf{T}\tilde{\mathbf{B}}.$$

Proof. First, note that

$$\mathbf{B} = [\mathbf{B} \quad \mathbf{AB} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{C}\mathbf{e}_1,$$

where $\mathbf{e}_1 = (1, 0, \dots, 0)$, and similarly, $\tilde{\mathbf{B}} = \tilde{\mathbf{C}}\mathbf{e}_1$. Therefore, $\mathbf{C}^{-1}\mathbf{B} = \tilde{\mathbf{C}}^{-1}\tilde{\mathbf{B}}$, and hence

$$\mathbf{B} = \mathbf{C}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{B}} = \mathbf{T}\tilde{\mathbf{B}}.$$

Next, let

$$p(\lambda) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$$

denote the characteristic polynomial of \mathbf{A} . Then, by the Cayley-Hamilton theorem (Theorem A.6), it follows

$$\mathbf{A}^n = -c_0\mathbf{I} - c_1\mathbf{A} - \cdots - c_{n-1}\mathbf{A}^{n-1},$$

and hence

$$\begin{aligned}
 \mathbf{A}\mathbf{C} &= [\mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \mathbf{A}^3\mathbf{B} \quad \cdots \quad \mathbf{A}^n\mathbf{B}] \\
 &= [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \cdots & 0 & -c_2 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{bmatrix} \\
 &= \mathbf{C}\mathbf{P},
 \end{aligned}$$

where \mathbf{P} denotes the companion matrix of $p(\lambda)$ (see Exercise 5.9). Similarly, $\tilde{\mathbf{A}}\tilde{\mathbf{C}} = \tilde{\mathbf{C}}\mathbf{P}$ for the same companion matrix \mathbf{P} (since \mathbf{A} and $\tilde{\mathbf{A}}$ have the same characteristic polynomial). Therefore,

$$\mathbf{A}\mathbf{T} = \mathbf{A}\mathbf{C}\tilde{\mathbf{C}}^{-1} = \mathbf{C}\mathbf{P}\tilde{\mathbf{C}}^{-1} = \mathbf{C}\tilde{\mathbf{C}}^{-1}\tilde{\mathbf{A}} = \mathbf{T}\tilde{\mathbf{A}},$$

so $\mathbf{A} = \mathbf{T}\tilde{\mathbf{A}}\mathbf{T}^{-1}$. □

We can now prove the pole placement theorem.

Proof of Theorem 6.9. By the above lemma, if the system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ is controllable, then there is a change of coordinates $\mathbf{x} = \mathbf{T}\tilde{\mathbf{x}}$ such that the transformed system is in controllable canonical form:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -c_0 & -c_1 & -c_2 & \cdots & -c_{n-1} \end{bmatrix} \tilde{\mathbf{x}} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}u,$$

where the characteristic polynomial of \mathbf{A} (the open-loop characteristic polynomial) is

$$p_{ol}(s) = s^n + c_{n-1}s^{n-1} + \cdots + c_1s + c_0.$$

Considering a state feedback of the form

$$u = -\tilde{\mathbf{K}}\tilde{\mathbf{x}} = -[k_0 \quad k_1 \quad \cdots \quad k_{n-1}] \tilde{\mathbf{x}},$$

the closed-loop equations are

$$\dot{\tilde{\mathbf{x}}} = (\tilde{\mathbf{A}} - \tilde{\mathbf{B}}\tilde{\mathbf{K}})\tilde{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -c_0 - k_0 & -c_1 - k_1 & -c_2 - k_2 & \cdots & -c_{n-1} - k_{n-1} \end{bmatrix} \tilde{\mathbf{x}}.$$

The characteristic polynomial of $\tilde{\mathbf{A}} - \tilde{\mathbf{B}}\tilde{\mathbf{K}}$ (i.e., the closed-loop characteristic polynomial) is then

$$p_{cl}(s) = s^n + (c_{n-1} + k_{n-1})s^{n-1} + \cdots + (c_1 + k_1)s + (c_0 + k_0).$$

Denoting the desired characteristic polynomial by

$$p(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0,$$

then we see we can make the closed-loop characteristic polynomial match the desired characteristic polynomial by choosing

$$k_j = a_j - c_j, \quad j = 0, \dots, n-1.$$

Finally, the feedback law can be written equivalently as

$$u = -\tilde{\mathbf{K}}\tilde{\mathbf{x}} = -\tilde{\mathbf{K}}\mathbf{T}^{-1}\mathbf{x} = \mathbf{K}\mathbf{x},$$

with $\mathbf{K} = \tilde{\mathbf{K}}\mathbf{T}^{-1}$. Finally, we find

$$\mathbf{A} - \mathbf{B}\mathbf{K} = \mathbf{T}\tilde{\mathbf{A}}\mathbf{T}^{-1} - (\mathbf{T}\tilde{\mathbf{B}})(\tilde{\mathbf{K}}\mathbf{T}^{-1}) = \mathbf{T}(\tilde{\mathbf{A}} - \tilde{\mathbf{B}}\tilde{\mathbf{K}})\mathbf{T}^{-1}.$$

Since $\mathbf{A} - \mathbf{B}\mathbf{K}$ and $\tilde{\mathbf{A}} - \tilde{\mathbf{B}}\tilde{\mathbf{K}}$ are related by a similarity transformation, they have the same characteristic polynomial $p(s)$, as desired. \square

6.3 Tracking with state feedback

We have seen how to use state feedback in order to modify the dynamics of a system, but could we use state feedback for other control objectives, such as tracking a desired reference signal? We will see in this section that we can.

First, let us introduce some convenient notation. We will denote the state-space system

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\end{aligned}$$

by

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right].$$

In MATLAB, this state-space system is specified as `ss(A,B,C,D)`. The advantage of this notation is that it avoids awkward and confusing statements such as “the \mathbf{A} matrix is $\mathbf{A} - \mathbf{B}\mathbf{K}$ ” or “the \mathbf{C} matrix is \mathbf{K} .” In this section, we will consider a single-input single-output plant described by the realization

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & 0 \end{array} \right]. \quad (6.10)$$

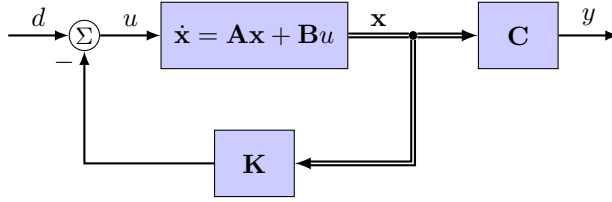


Figure 6.1: Typical block diagram for disturbance rejection with state feedback.

That is, the \mathbf{D} matrix is zero: this is true for every real system (since the response must roll off at high frequency), and it simplifies the formulas and block diagrams; however, all of the methods that follow also work for a nonzero \mathbf{D} matrix.

Before addressing the problem of tracking a reference signal, we first discuss the slightly simpler problem of disturbance rejection. As in §2.6, we model an external disturbance d as entering in the same way as the control input u , as shown in the block diagram in Figure 6.1. Thus, for disturbance rejection, the input to the plant becomes

$$u = -\mathbf{K}\mathbf{x} + d$$

and the state equation becomes

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ &= \mathbf{A}\mathbf{x} + \mathbf{B}(-\mathbf{K}\mathbf{x} + d) \\ &= (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}d.\end{aligned}$$

The output is still $y = \mathbf{C}\mathbf{x}$, so the overall state-space realization of the closed-loop transfer function from d to y is

$$\left[\begin{array}{c|c} \mathbf{A} - \mathbf{B}\mathbf{K} & \mathbf{B} \\ \hline \mathbf{C} & 0 \end{array} \right].$$

Note that this is identical to the original (open-loop) plant (6.10), except the \mathbf{A} matrix has been replaced by $\mathbf{A} - \mathbf{B}\mathbf{K}$.

Now, let us look at the problem of tracking a reference input. More precisely, we want to design a feedback controller such that the output $y(t)$ tracks the given reference $r(t)$. One way to do this is shown in Figure 6.2, where we have introduced an additional (scalar) parameter k_r in our controller. Note that the controller now has two inputs: the state vector \mathbf{x} as well as the reference signal r to be tracked. The diagram corresponds to the feedback law

$$u = -\mathbf{K}\mathbf{x} + k_r r,$$

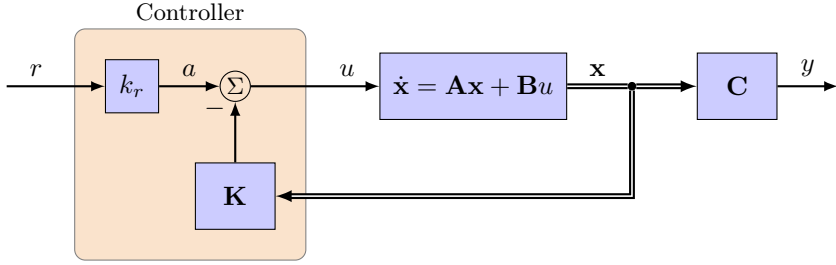


Figure 6.2: Typical block diagram for reference tracking with state feedback. The controller has two inputs, the reference r and the state vector \mathbf{x} , and has a single output u .

and the corresponding state equations are

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}(-\mathbf{K}\mathbf{x} + k_r r) \\ &= (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{B}k_r r,\end{aligned}$$

so a state-space realization for the closed-loop system with input r and output y is given by

$$\left[\begin{array}{c|c} \mathbf{A} - \mathbf{BK} & \mathbf{B}k_r \\ \hline \mathbf{C} & 0 \end{array} \right]. \quad (6.11)$$

We could choose the gain matrix \mathbf{K} by pole placement (or LQR, to be discussed in §6.5), but how should we choose the scalar gain k_r ? We want to choose it so that at steady state, the output y matches the reference r . At steady state, $\dot{\mathbf{x}} = \mathbf{0}$, so the steady-state value \mathbf{x}_e satisfies

$$\begin{aligned}0 &= (\mathbf{A} - \mathbf{BK})\mathbf{x}_e + \mathbf{B}k_r r \\ \implies \mathbf{x}_e &= -(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{B}k_r r.\end{aligned}$$

Since $y = \mathbf{C}\mathbf{x}$, the steady-state value of y is therefore

$$y_e = \mathbf{C}\mathbf{x}_e = -\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{B}k_r r.$$

Since we want $y_e = r$, we therefore choose

$$k_r = \frac{-1}{\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{B}}. \quad (6.12)$$

Note that the (scalar) quantity $-\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1}\mathbf{B}$ is the zero-frequency gain of the closed-loop transfer function from a to y in Figure 6.2. Thus, k_r is simply the reciprocal of this zero-frequency gain; this choice of k_r thus makes the zero-frequency gain of the transfer function from r to y exactly 1, as desired.

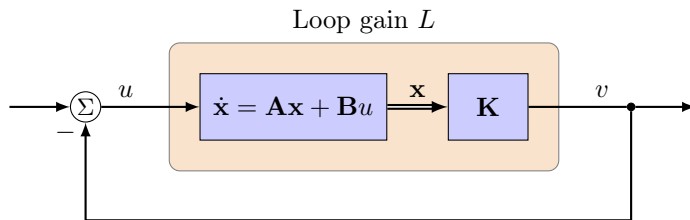


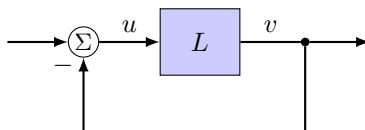
Figure 6.3: Block diagram of a typical state feedback controller, showing the loop gain L .

6.4 Loop gain

We now have some powerful tools for designing controllers using state feedback. In particular, pole placement gives us the ability to place the closed-loop poles anywhere we like (provided our system is controllable), and we now know how to use a state feedback controller for disturbance rejection or reference tracking. But how do we evaluate *how good* a particular state feedback controller is? Over what frequency range can we expect good performance (e.g., good disturbance rejection or good tracking)? Given two different controllers, with different closed-loop pole locations, which is preferable? In order to answer these questions properly, we need to return to the concept of the loop gain (or loop transfer function), discussed in §4.4.

We have seen previously that the loop gain, an open-loop transfer function, tells us a great deal about the closed-loop system. In particular, the loop gain $L(s)$ can tell us both about performance (good performance for frequencies ω where $|L(i\omega)| \gg 1$) and about robustness (from the Nyquist plot of L , e.g., gain and phase margins). From it, we can also determine the sensitivity function $S = 1/(1 + L)$ and its complement $T = L/(1 + L)$, which are excellent measures of closed-loop performance, as we saw in §2.6. Can we apply these ideas to state feedback controllers?

It turns out we can, at least when we have a single input u . Consider the diagram for a typical state-feedback controller, as shown in Figure 6.3. The loop gain is indicated by the orange box in the figure, and the important thing to note about it is that, although \mathbf{x} is a vector, the input u and output v of the orange box are both *scalars*. Thus, the loop gain is actually a single-input single-output (SISO) system, as in the following diagram:



A state-space realization of the loop gain is then

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ v &= \mathbf{K}\mathbf{x}\end{aligned}$$

or, in our compact notation,

$$L = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{K} & 0 \end{array} \right]. \quad (6.13)$$

If desired, we can determine the corresponding (scalar) transfer function $L(s) = \mathbf{K}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, and plot Nyquist plots or Bode plots exactly as we did in classical feedback design, to evaluate performance or stability margins. This turns out to be a valuable tool for evaluating state feedback controllers, as we illustrate with the examples below.

Example 6.13 (Motor angle). Consider a servo motor described by the transfer function

$$P(s) = \frac{1}{s(s+1)(s/a+1)}, \quad a = 100.$$

Here, the input u is the motor voltage, the output y is the shaft angle, and we have included both a mechanical lag (the pole at $s = -1$), due to the inertia of the motor, as well as an electrical lag (the pole at $s = -a$), due to the inductance of the motor coil. (See Franklin et al. [11, Example 2.13] for a good discussion of the dynamics of a DC motor.) Suppose we are asked to design a controller to track a given reference angle, with a bandwidth of around 2 rad/sec. We will attempt to do this using state feedback $u = -\mathbf{K}\mathbf{x} + k_r r$, where \mathbf{K} is designed using pole placement.

We first determine a state-space realization for the given transfer function: choosing a state $\mathbf{x} = (y, \dot{y}, \ddot{y})$, a realization is (see §5.3)

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -a & -a-1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} u \\ y &= [1 \quad 0 \quad 0] \mathbf{x} + 0u.\end{aligned}$$

For pole placement, we need to choose closed-loop pole locations. Where shall we choose these? We are given a bandwidth specification of 2 rad/sec, so it seems reasonable to place the closed-loop poles at a radius of about 2 from the origin, perhaps a little further out to be on the safe side. So for our first try we determine a gain matrix $\mathbf{K} = \mathbf{K}_1$ with

$$\text{poles at } -2 \pm 2i, -4 \implies \mathbf{K}_1 = [0.32 \quad -0.76 \quad -0.93].$$

These may be calculated in MATLAB with the following commands:

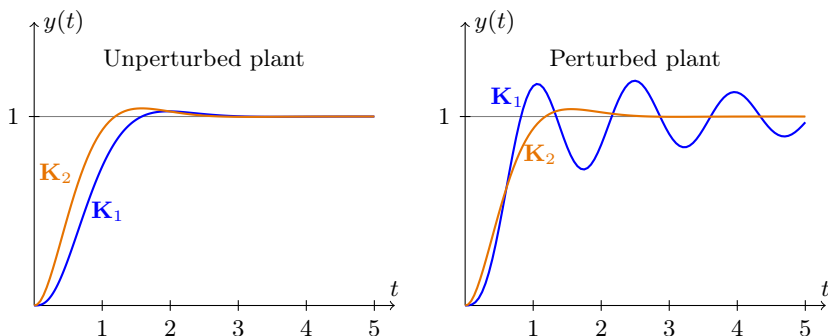


Figure 6.4: Step responses for Example 6.13.

```
poles = [-2+2i, -2-2i, -4];
K1 = place(A,B,poles);
```

We then calculate the appropriate value of k_r from (6.12), and the closed-loop transfer function from r to y is then given by (6.11). The resulting step response is plotted in Figure 6.4 (left), and looks perfectly reasonable.

For comparison, consider a different choice of pole locations, with corresponding gains $\mathbf{K} = \mathbf{K}_2$ given by

$$\text{poles at } -2 \pm 2i, -100 \implies \mathbf{K}_2 = \begin{bmatrix} 8.0 & 3.08 & 0.03 \end{bmatrix}.$$

The corresponding closed-loop step response is also shown in Figure 6.4. The response looks a little faster than the response for gains \mathbf{K}_1 , but otherwise there does not appear to be much difference, at least from the step response. Which is the better choice of pole locations and gains?

In order to better understand how these two controllers would work on a real system, suppose that our plant model is slightly wrong: in particular, suppose the *actual* plant has an overall gain that is 6% larger than our model. If we use the same gains \mathbf{K}_1 and \mathbf{K}_2 in a feedback loop with this perturbed plant, we obtain the closed-loop step responses shown in Figure 6.4 (right): the response with $\mathbf{K} = \mathbf{K}_2$ is almost identical to the response for the unperturbed plant, but the response with $\mathbf{K} = \mathbf{K}_1$ has changed dramatically: it has oscillations, and is on the verge of instability. In fact, if the overall gain of the plant is increased by 7%, then the closed-loop system for $\mathbf{K} = \mathbf{K}_1$ is unstable! The controller \mathbf{K}_1 apparently has serious problems with robustness, while the controller \mathbf{K}_2 appears to be much better behaved. But both were designed with the same method, pole placement, with closed-loop pole locations that seemed reasonable.

To understand what is happening with the example, let us compare the loop gain for the two systems. Let

$$L_1 = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{K}_1 & 0 \end{array} \right], \quad L_2 = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{K}_2 & 0 \end{array} \right].$$

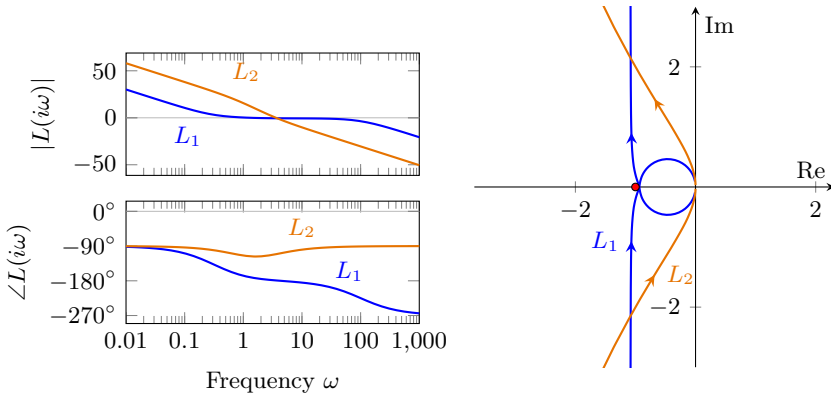


Figure 6.5: Bode plot (left) and Nyquist plot (right) of loop gain for Example 6.13.

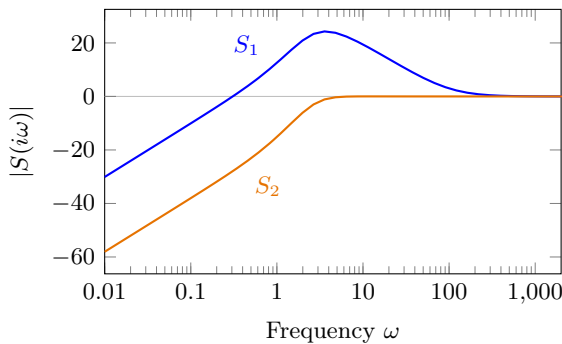


Figure 6.6: Sensitivity function corresponding to loop gains L_1 and L_2 for Example 6.13.

The Bode and Nyquist plots of these two systems are shown in Figure 6.5. Note that the gain and phase margins for L_1 are extremely small (the Nyquist plot comes very close to the -1 point), while L_2 has healthy margins. Furthermore, from the Bode magnitude plot, we see that at low frequencies, L_2 is much larger than L_1 , so we should expect L_2 to have better performance at low frequencies; also, at high frequencies, the magnitude of L_2 is much smaller than that of L_1 , so we would expect L_2 to amplify high-frequency noise much less than L_1 . Indeed, L_2 seems much better than L_1 in every respect.

What has gone wrong with L_1 ? We designed the gains \mathbf{K}_1 and \mathbf{K}_2 using the same procedure, pole placement, both with closed-loop poles that seemed reasonable. We can best understand the problem by looking at the sensitivity

function

$$S(s) = \frac{1}{1 + L(s)}.$$

The Bode magnitude plot of the sensitivity function corresponding to the two loop gains is shown in Figure 6.6. Note that S_1 has a large peak, which implies the bad robustness observed for L_1 (recall from §4.7 that the distance from the Nyquist plot to the -1 point is $1/|S(i\omega)|$, so a large peak in $|S|$ corresponds to a point very close to the -1 point). In contrast, S_2 has no such peak. This is consistent with our observation that \mathbf{K}_1 has poor robustness, but we have still not explained *why* S_1 has the large peak.

💡 To understand this, first note that if $L(s)$ has a pole at $s = p$, then $S(p) = 0$. Thus, open-loop poles (of L) are *zeros* of S , while closed-loop poles (which we specify with pole placement) are *poles* of S . Furthermore, note that $|S(i\omega)| \rightarrow 1$ as $\omega \rightarrow \infty$, since $|L(i\omega)| \rightarrow 0$ as $\omega \rightarrow \infty$. For this example, L_1 and L_2 have the same poles (since they have the same A matrix), and in particular there is an open-loop pole at $s = -100$. Thus, S_1 has a *zero* at $s = -100$, and thus the Bode plot of S_1 “breaks up” (slope increases by one) at the corner frequency $\omega = 100$ rad/sec. Starting from the high frequency portion of the Bode plot, then, $|S_1|$ starts at a magnitude of 1, and then as the frequency decreases, $|S_1|$ starts to increase at $\omega = 100$ because of this zero. In our pole placement procedure, we specified the poles of S_1 (these are the closed-loop poles), and these have corner frequencies of 4 rad/sec and $2\sqrt{2} \approx 2.8$ rad/sec. Thus, as frequency decreases, the Bode magnitude plot continues to increase until we reach these corner frequencies, at which the slope then decreases again. By contrast, for S_2 , we chose a closed-loop pole at $s = -100$, which *cancels* with the closed-loop zero at $s = 100$. So for S_2 , the Bode magnitude plot does not break up at $s = 100$, and we avoid the large peak in $|S|$ that caused the lack of robustness for gains \mathbf{K}_1 . ♦

The previous example and corresponding explanation motivates the following guideline for pole placement:

Fast poles: If the plant has a fast pole, place a closed-loop pole at the same location.

Here, “fast” means relative to the timescale of the dynamics you want to control, for instance the desired bandwidth of the system.

One might think that, if one encounters robustness problems such as that observed in Example 6.13, one could simply move the closed-loop poles farther in the left half-plane, until the robustness problems go away. The next example shows that this is not always a good strategy either.

Example 6.14 (Vehicle steering). The following transfer function models the lateral dynamics of a car:

$$P(s) = \frac{\alpha s + 1}{s^2}, \quad \alpha = 0.5. \quad (6.14)$$

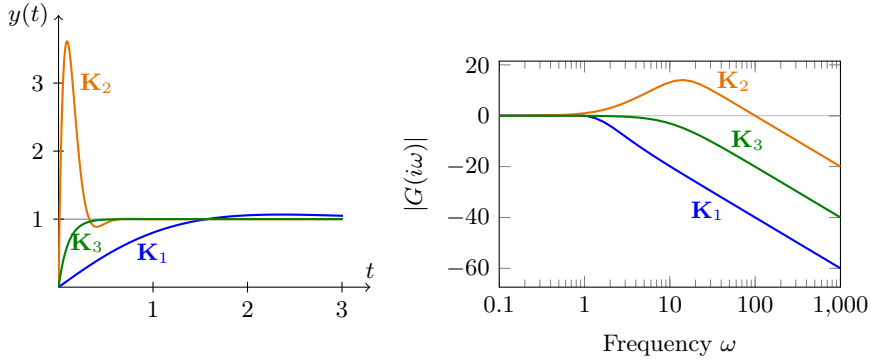


Figure 6.7: Step response (left) and Bode plot (right) of closed-loop transfer function from r to y for Example 6.14.

Here the input u is the steering angle, the output y is the lateral position, and the dynamics are linearized about small steering angles, and small angles of the vehicle's orientation. (See Åström and Murray [1, Example 5.12] for a derivation of these dynamics.) We wish to design a controller, again using state feedback, to track a reference position r . This control objective could be thought of as a “lane change” maneuver: we specify the desired lateral position r , and the controller should drive the output y to that position.

A state-space realization of this transfer function is

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & 0 \end{array} \right] = \left[\begin{array}{cc|c} 0 & 1 & \alpha \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \end{array} \right].$$

As with the previous example, we will use a control law $u = -\mathbf{K}\mathbf{x} + k_r r$, where \mathbf{K} is determined by pole placement. First, let us choose $\mathbf{K} = \mathbf{K}_1$, with

$$\text{poles at } -1 \pm i \implies \mathbf{K}_1 = \begin{bmatrix} 2 & 1 \end{bmatrix}.$$

We calculate k_r from (6.12) and the closed-loop transfer function from r to y is again given by (6.11). Its step response is shown in Figure 6.7.

This step response looks reasonable, and in fact the steady-state value settles out to a final value of exactly 1 after about $t = 5$. But now suppose we wish to have a faster response. In order to achieve this, we should simply place the closed-loop poles further in the left half-plane, for instance choosing $\mathbf{K} = \mathbf{K}_2$, with

$$\text{poles at } -10 \pm 10i \implies \mathbf{K}_2 = \begin{bmatrix} 200 & -80 \end{bmatrix}.$$

The resulting closed-loop step response is also shown in Figure 6.7, and we now see something unexpected: indeed, the response is faster, and the step

response settles to a final value of 1 after about $t = 0.5$, as desired. However, the transient response shows a very large overshoot. This is a major problem: if this were a lane change maneuver, with the step response corresponding to shifting over by 1 lane, this response corresponds to first shifting over by more than 3 lanes, and then moving back to the desired lane.

Now suppose we consider yet another choice of gains $\mathbf{K} = \mathbf{K}_3$, with

$$\text{poles at } -10, -2 \implies \mathbf{K}_3 = \begin{bmatrix} 20 & 2 \end{bmatrix}.$$

This step response is also shown in Figure 6.7, and now the response is faster, as desired, and the overshoot has been eliminated. It is perhaps surprising that \mathbf{K}_3 has a better response than \mathbf{K}_2 : after all, \mathbf{K}_2 has two poles *farther* in the left half plane than \mathbf{K}_3 , so it would make sense that \mathbf{K}_2 should have the faster, better response. Why does \mathbf{K}_2 have such a large overshoot?

The answer is that the plant has a *slow zero*. Recall from §3.4 that zeros in the left half-plane make the transient response worse, for instance in terms of overshoot. The “slower” the zero is (that is, the closer to the imaginary axis it is), the worse this overshoot is. Here, the closed-loop transfer function can be written

$$G(s) = \frac{k_r P(s)}{1 + L(s)},$$

(see Exercise 6.8), and note that the plant P in (6.14) has a zero at $s = -1/\alpha = -2$, so the closed-loop transfer function $G(s)$ also has a zero at $s = -2$. When controller \mathbf{K}_1 is used, with closed-loop poles at $-1 \pm i$, this is not a “slow” zero (since it is farther to the left than our closed-loop poles). However, when \mathbf{K}_2 is used, now the closed-loop poles are at $-10 \pm 10i$, and now the zero is much closer to the imaginary axis than the closed-loop poles, and has a large effect on the step response. When \mathbf{K}_3 is used, however, we placed a closed-loop pole at $s = -2$, and this cancels with the zero in $P(s)$, so does not affect the step response.

The effect of this zero at $s = -2$ is perhaps best illustrated by the Bode plot of the closed-loop transfer function $G(s)$, which is shown in Figure 6.7. Note that at low frequencies, the gain is 1, as desired for good tracking. For \mathbf{K}_1 , we have placed the closed-loop poles at $1 \pm i$, for a corner frequency of $\sqrt{2} \approx 1.4$ rad/sec, so the Bode magnitude plot “breaks down” (decreases slope) just before the corner frequency of the zero at 2 rad/sec. However, for \mathbf{K}_2 , we have placed the closed-loop poles at $-10 \pm 10i$, so the corner frequency is $10\sqrt{2} \approx 14$ rad/sec. The Bode plot thus first “breaks up” (increases slope) at 2 rad/sec, and does not “break down” again until about 14 rad/sec. This results in a large peak in the Bode plot, corresponding to very bad tracking at these frequencies, and overshoot in the step response. Controller \mathbf{K}_2 solves this problem by placing a closed-loop pole at $s = -2$, which cancels with the zero at $s = -2$, avoiding the large peak in the Bode plot. \diamond

We can summarize the lesson learned from this example as follows:

Slow zeros: If the plant has a slow zero, place a closed-loop pole at the same location.

Here, as with the example of fast poles, “slow” is relative to the timescale of the dynamics you want to control, for instance the desired bandwidth of the system.

We have seen that Bode and Nyquist plots can be valuable tools for analyzing the performance and robustness of state-feedback controllers. In fact, while pole placement is a powerful tool, and can be a very useful method for designing state feedback controllers, if the poles are not chosen wisely, the resulting controllers can have very poor robustness (e.g., stability margins), or large transients in the step response. Nyquist and Bode plots are effective tools at identifying these situations. However, it would be nice to have a method that is more reliable than pole placement, for obtaining “good” state feedback controllers. The next section introduces a different method for doing precisely this, and is typically much more reliable than pole placement.

6.5 Linear Quadratic Regulator (LQR)

The Linear Quadratic Regulator (LQR) is an optimal controller for a linear system of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (6.15)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$. In particular, the goal is to find an optimal state feedback law of the form $\mathbf{u} = -\mathbf{K}\mathbf{x}$, to minimize the quadratic cost function

$$J = \int_0^\infty (\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt, \quad (6.16)$$

where \mathbf{Q} and \mathbf{R} are symmetric, positive-definite matrices. (Recall that a matrix \mathbf{M} is positive definite if $\mathbf{v}^\top \mathbf{M} \mathbf{v} > 0$ for all $\mathbf{v} \neq 0$ —see Appendix A.2.) The expression (6.16) for the cost is just a general way of writing a cost function that is *quadratic* in both the state \mathbf{x} and the input \mathbf{u} . Thus, the Linear Quadratic Regulator is an optimal *linear* controller to minimize a *quadratic* cost function.

In the above statement of the LQR problem, the control objective is to drive the state $\mathbf{x}(t)$ to zero, from an initial state \mathbf{x}_0 . In practice, however, the same controller can be used for disturbance rejection or reference tracking problems, precisely as we did in Section 6.3 for state-feedback controllers designed using pole placement.

Before looking at how to solve this optimal control problem, let us consider an example of a typical cost function. Suppose $\mathbf{x} = (x, \theta, \dot{x}, \dot{\theta})$ and $\mathbf{u} =$

(u_1, u_2) , and suppose \mathbf{Q} and \mathbf{R} are diagonal matrices with

$$\mathbf{Q} = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & q_3 & \\ & & & q_4 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_1 & \\ & r_2 \end{bmatrix}.$$

(Note that \mathbf{Q} and \mathbf{R} do not need to be diagonal—they need only be symmetric—though they often are chosen to be diagonal.) Then

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix} \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & q_3 & \\ & & & q_4 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = q_1 x^2 + q_2 \theta^2 + q_3 \dot{x}^2 + q_4 \dot{\theta}^2.$$

Similarly,

$$\mathbf{u}^T \mathbf{R} \mathbf{u} = r_1 u_1^2 + r_2 u_2^2.$$

Thus, the terms $\mathbf{x}^T \mathbf{Q} \mathbf{x}$ in the cost function (6.16) represent a penalty on the states $(x, \theta, \dot{x}, \dot{\theta})$ being different from zero, and the terms $\mathbf{u}^T \mathbf{R} \mathbf{u}$ represent a penalty on the inputs (u_1, u_2) being different from zero. The conditions on positive definiteness of \mathbf{Q} and \mathbf{R} imply the diagonal elements q_j, r_j must be positive. These parameters q_j, r_j are our design parameters, the knobs we tweak in designing a state feedback using LQR. It should be clear why they must be positive: if any of these coefficients is negative, then we could *reduce* the cost, and thereby obtain an “improved” controller, by making the corresponding state or control input *larger*. This is contrary to our control objective, which is to drive the states to zero, with small inputs.

The matrices \mathbf{Q}, \mathbf{R} are weights that specify how much we care about good performance, versus control effort:

- \mathbf{Q} provides a penalty on tracking error.
- \mathbf{R} provides a penalty on control effort.

The gain matrix \mathbf{K} that minimizes J in (6.16) thus provides an optimal compromise between good regulation ($\mathbf{x}^T \mathbf{Q} \mathbf{x}$ small) and small inputs ($\mathbf{u}^T \mathbf{R} \mathbf{u}$ small). If we want an “aggressive” controller that drives the state to zero rapidly, and are not so concerned with the actuator effort required, then we should choose the entries in \mathbf{Q} to be large, providing a large penalty on the states, and the entries in \mathbf{R} to be small, for a relatively small penalty on the inputs. Conversely, if we want a less aggressive controller that uses smaller actuator effort, we should decrease the weights in \mathbf{Q} , or increase the weights in \mathbf{R} . Note that scaling both \mathbf{R} and \mathbf{Q} by the same factor simply scales the overall cost function (6.16) by that factor, and so does not affect the optimal gain \mathbf{K} .

We will derive the solution of this optimization problem later, but for now let us just state the solution: the optimal gain matrix \mathbf{K} is given by

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}, \quad (6.17)$$


where \mathbf{P} is an $n \times n$ symmetric positive-definite matrix that satisfies the matrix equation

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0, \quad (6.18)$$

which is called an *algebraic Riccati equation*. This equation is *quadratic* in \mathbf{P} , so even in the scalar case ($n = 1$, for which all the matrices in (6.18) are scalars), it is clear that there can be more than one solution. It is important that we pick a solution that is *positive definite*. One can show that there is a *unique* positive-definite solution as long as all of the following conditions are met:

- The pair (\mathbf{A}, \mathbf{B}) is controllable.
- The matrix \mathbf{R} is positive definite.
- The matrix \mathbf{Q} is positive definite.

The resulting controller one obtains from LQR is guaranteed to stabilize the system: that is, as long as the above conditions are met, the eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}$ will always be in the left half-plane.

 Actually, the third condition above is a little stronger than necessary: \mathbf{Q} can be positive semi-definite, as long as it satisfies a certain additional condition. In particular, any symmetric positive semi-definite matrix may be written $\mathbf{Q} = \mathbf{C}^T\mathbf{C}$ for some matrix \mathbf{C} , and in order for there to be a unique positive-definite solution of the Riccati equation, the pair $(\mathbf{A}^T, \mathbf{C}^T)$ must be controllable, or, alternatively, the pair (\mathbf{A}, \mathbf{C}) must be observable.

Solving (6.18) by hand is challenging, but in MATLAB, one can compute the solution easily, with

```
K = lqr(A,B,Q,R)
```

where \mathbf{A} and \mathbf{B} are the matrices from the dynamics (6.15), and \mathbf{Q} and \mathbf{R} are the matrix weights in the cost function (6.16).

Guaranteed robustness

There is a remarkable fact about LQR that makes it a particularly reliable tool for control design: not only is the closed-loop system guaranteed to be *stable* (i.e., eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}$ in the left half-plane), but also one gets guaranteed *robustness*, in a way that we will quantify in this section.

We consider a single-input system, so u is a scalar, \mathbf{B} is a single column, and the state feedback gain \mathbf{K} is a single row. Recall from Figure 6.3 that the loop gain for a feedback system with state feedback $u = -\mathbf{K}\mathbf{x}$ is given by

$$L = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{K} & 0 \end{array} \right],$$

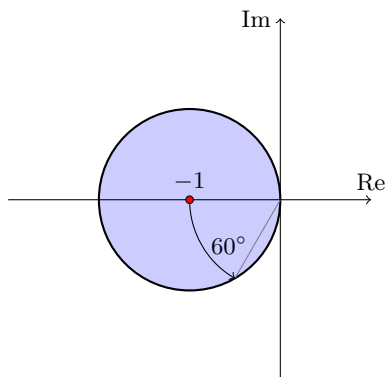


Figure 6.8: Guaranteed stability margins for LQR controllers. The Nyquist plot of the loop gain L must lie outside the shaded region. This implies a phase margin of at least 60° , an infinite gain margin, and a gain reduction margin of at least 2 (i.e., the gain can be reduced by at least a factor of 2 without making the closed-loop system unstable).

so the transfer function is $L(s) = \mathbf{K}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, a scalar transfer function (single-input, single-output). The robustness guarantees follow from the following remarkable result:

If the gain matrix \mathbf{K} is the solution of an LQR problem, then

$$|1 + L(i\omega)| \geq 1, \quad \forall \omega. \quad (6.19)$$

This expression can be obtained using (6.17), by suitably manipulating the Riccati equation (see Exercise 6.11). Why does (6.19) imply that the feedback system is robust? Recall from §4.7 that $|1 + L(i\omega)|$ is the distance from the -1 point to the point $L(i\omega)$ on the Nyquist plot of L . Thus, (6.19) implies that *this distance is always at least 1*. The Nyquist plot of L is therefore guaranteed to lie outside the unit circle centered about the -1 point, as shown in Figure 6.8. This implies that we always have healthy stability margins: in particular, the stability margin s_m from (4.16) must be at least 1; the phase margin must be at least 60° ; the gain margin is infinite, and there is a gain reduction margin of at least 2 (i.e., the gain can be reduced by at least a factor of two without making the closed-loop system unstable).

Note that the condition (6.19) also implies that $|S(i\omega)| \leq 1$ for all ω , and thus there are no frequencies at which feedback is detrimental. (See §4.7, in particular Figure 4.23.)



The astute reader might object that this result appears to violate Bode's integral formula (Theorem 4.20). However, here the loop gain L has relative degree 1, so does not satisfy the conditions of that theorem. This will, however, come back to bite us when we try to connect an LQR controller with an observer, in which case Theorem 4.20 will apply. In fact, there are no such robustness guarantees when an observer is used with an LQR regulator, as shown in a famous paper by Doyle [6].

Example 6.15. Recall the problem of a cart driving on a seesaw: the equations of motion for this system were derived in Example 1.11, where the control input is a force applied to the cart. We will design a controller to stabilize the linearized system (1.45) with a full-state feedback law of the form $u = -\mathbf{K}\mathbf{x}$, in particular comparing controllers designed using pole placement, and using LQR.

We choose (dimensionless) parameters $a = b = 0.1$, $\mu = 2$, $\delta = 10$, for which the \mathbf{A} matrix defined by (1.45) has eigenvalues

$$0.65, \quad -0.34 \pm 0.20i, \quad -10.1.$$

Note that the equilibrium is unstable, since one of the eigenvalues has positive real part. We first choose the gains \mathbf{K} using pole placement, specifying closed-loop poles to be in the left half-plane. Here, we specify the closed-loop poles to lie at

$$-0.6 \pm 0.6i, \quad -1, \quad -1.5,$$

for which the resulting gains are

$$\mathbf{K} = \begin{bmatrix} 5.6 & 3.8 & -5.9 & 4.8 \end{bmatrix}.$$

We compare these to a controller designed using LQR, with

$$\mathbf{Q} = \begin{bmatrix} 10 & & & \\ & 10 & & \\ & & 0 & \\ & & & 0 \end{bmatrix}, \quad R = 1,$$

for which the optimal gains are

$$\mathbf{K} = \begin{bmatrix} 19.6 & 10.3 & 3.3 & 16.0 \end{bmatrix}.$$

These two controllers are compared in Figure 6.9, which shows the response (both the cart position x and seesaw angle θ) for an initial state $(x, \theta, \dot{x}, \dot{\theta}) = (0.1, 0, 0, 0)$. The time responses of the two controllers look similar, and both require comparable control inputs u , so based on these criteria, one might conclude that both controllers should work well. However, the Nyquist plots for two controllers tell a different story: the controller designed using pole placement comes quite close to the -1 point (with a phase margin of only 14°), so unless the plant model is very accurate, this controller is likely not to work in practice. On the other hand, the LQR controller stays well away from the -1 point (with a phase margin of 61°), as is guaranteed by (6.19), so this controller should be more robust to uncertainties in the plant. In this example, the pole placement controller can be improved by heeding the guidelines from Section 6.4: in particular, if we place a closed-loop pole at the location of the “fast” pole at -10.1 , the robustness improves dramatically. However, the point is that LQR *always* gives a controller with good stability margins, no matter what we choose for \mathbf{Q}, R in the cost function. \diamond

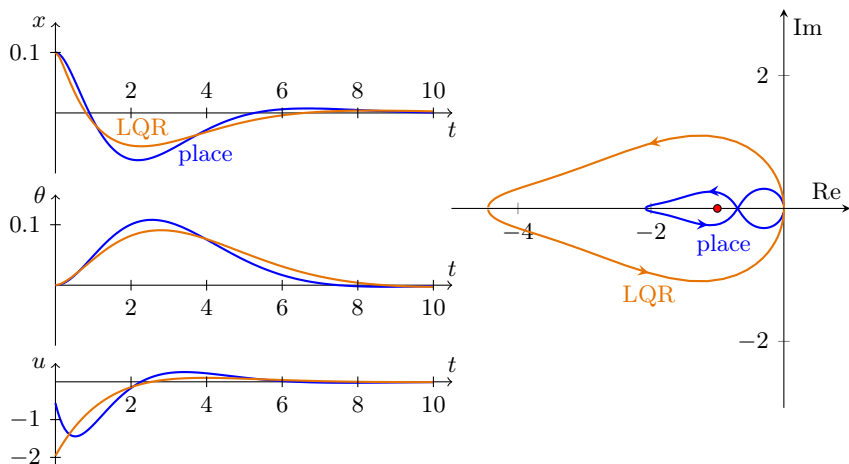


Figure 6.9: Time responses and Nyquist plots for Example 6.15, showing controllers designed by pole-placement (labeled ‘place’) and LQR. The time responses are similar, but the LQR controller has much better robustness, as shown by the Nyquist plots.

Optimal pole locations and the symmetric root locus

In Section 6.2, we selected state feedback gains \mathbf{K} using pole placement: we specified the desired closed-loop pole locations, and then solved for the corresponding gains \mathbf{K} . However, we saw that it can be tricky to choose pole locations to give us a “robust” controller (see Examples 6.13 and 6.14). We now have an alternative method, LQR, for finding the gains \mathbf{K} , that has some very desirable properties: in particular, optimality, and guaranteed robustness. We might therefore ask whether this optimal control approach can give us insight into how to select closed-loop poles when using pole placement. Fortunately, it can, as we now explore.

We once again assume we have a single input, so u is a scalar and \mathbf{B} is a single column. Furthermore, suppose we have a single output $y = \mathbf{C}\mathbf{x}$, which we use to define a cost function for LQR, as

$$J = \int_0^\infty (qy(t)^2 + u(t)^2) dt \quad (6.20)$$

(that is, $\mathbf{Q} = q\mathbf{C}^T\mathbf{C}$ and $\mathbf{R} = 1$), so now the scalar q is the only parameter we use for control design. A small value of q should result in a controller that uses small energy (u small), while a large value of q should give a more aggressive controller (y small). The following result follows from Exercise 6.11:

Let $G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ be the transfer function from u to y , and let the state feedback gains \mathbf{K} be determined by the LQR problem, with the cost function (6.20). Then the closed-loop poles are the left-half-plane solutions of

$$1 + qG(s)G(-s) = 0. \quad (6.21)$$

Note that this expression is in *root-locus form*, as in (4.5). In particular, for any value of $q > 0$, the closed-loop poles lie on the root locus plot of $G(s)G(-s)$. This is called the *symmetric root locus* plot of $G(s)$: observe that the poles and zeros of $G(s)G(-s)$ are precisely those of $G(s)$, together with their reflections about the imaginary axis. Thus, the entire root locus plot is symmetric not just about the *real* axis (as any root locus plot must be, since poles and zeros come in complex-conjugate pairs), but also about the *imaginary* axis.

Example 6.16 (Vehicle steering revisited). Figure 6.10 shows an example of a symmetric root locus plot for the plant $P(s)$ given by (6.14), from Example 6.14. The conventional root locus plot of $P(s)$ on the left shows the closed-loop poles if a simple proportional controller $u = -ky$ is used, as k varies from 0 to ∞ . The symmetric root locus plot, on the other hand, shows the closed-loop poles resulting from an optimal controller $u = -\mathbf{K}\mathbf{x} = -k_1x_1 - k_2x_2$, as the parameter q in the cost function (6.20) varies from 0 to ∞ .

Note that the symmetric root locus plot has four poles (since $P(s)$ has two poles), and the optimal closed-loop poles are only the two poles in the left half-plane. We see that, as $q \rightarrow 0$, the optimal closed-loop poles approach the open-loop poles at $s = 0$, and as q increases, the optimal closed-loop poles move away from the origin at an angle of about 45° with the real axis. For large values of q , one pole goes to the open-loop zero at $s = -1/\alpha$, and the other pole goes off to $-\infty$ along the negative real axis. This is consistent with what we observed in Example 6.14: for a robust controller, we should place a closed-loop pole at a location of the open-loop zero. \diamond

A useful consequence of (6.21) is that we can immediately identify the closed-loop pole locations for the *minimum-energy controller*, which corresponds to the limit $q \rightarrow 0$. As $q \rightarrow 0$ in (6.21), we know from the rules from sketching root locus plots (§4.2) that the roots of (6.21) go to the poles of $G(s)G(-s)$. Thus, the optimal closed-loop poles will be the same as the poles of $G(s)$, *but with any unstable poles reflected about the imaginary axis*, into the left half-plane. This is often a useful starting point when designing a state-feedback controller using pole placement.

It is also instructive to consider the limit as $q \rightarrow \infty$, which corresponds to an aggressive controller for which we are not concerned with control effort. In this case, some of the closed-loop poles go to the zeros of $G(s)$, again

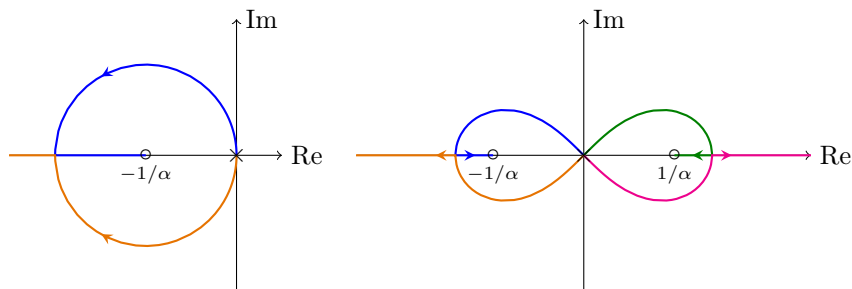


Figure 6.10: *Left:* Root locus plot of $P(s)$ for vehicle steering problem (6.14); *Right:* Symmetric root locus plot $P(s)P(-s)$.

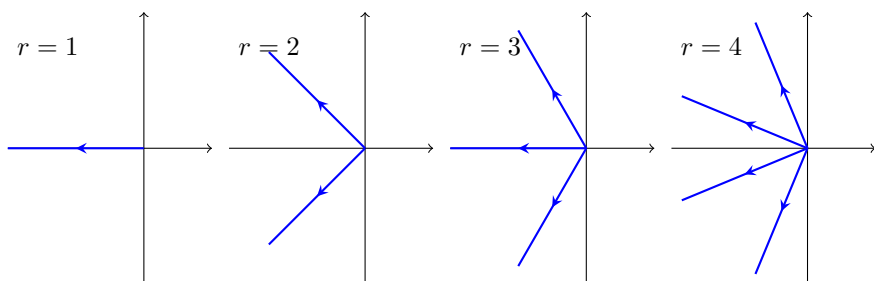


Figure 6.11: Asymptotes for optimal pole locations as $q \rightarrow \infty$ in (6.20), for systems with relative degree $r = 1, 2, 3, 4$, as given by (6.22).

reflecting any right-half-plane zeros into the left half-plane. The remaining r poles (where r is the relative degree of $G(s)$, $\#$ poles $- \#$ zeros) will go towards infinity in the left half-plane, at angles determined by the root locus asymptotes:

$$\theta_k = \frac{(2k + r - 1)\pi}{2r}, \quad k = 1, \dots, r. \quad (6.22)$$

These asymptotes are shown in Figure 6.11 for various values of r . This pattern of poles, lying on a circular arc with angles given by (6.22), is the same as the pole locations in a *Butterworth filter*, so is often referred to as a *Butterworth pattern*. This arrangement of poles can be another useful starting point when designing controllers using pole placement.



Solution of the LQR problem

There are several ways to solve the optimization problem for the linear quadratic regulator, and here we give two different derivations: first, we give a simpler derivation that is specific to linear time-invariant systems, and

then we give a more general derivation for a nonlinear system, and show that the LQR solution arises as a special case. For alternative derivations, see Friedland [12, Chap. 9] or Stengel [24, Sec. 3.4, 5.4].

The main idea is straightforward: we differentiate the cost function with respect to the gain \mathbf{K} , and solve for the gain \mathbf{K} for which the derivative is equal to zero. The details become somewhat tricky, because \mathbf{K} is a matrix, but this is the main idea.

First, we incorporate the gain \mathbf{K} into the cost function. Letting $\mathbf{u} = -\mathbf{K}\mathbf{x}$, the cost function (6.16) becomes

$$J = \int_0^\infty (\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{x}^\top \mathbf{K}^\top \mathbf{R} \mathbf{K} \mathbf{x}) dt = \int_0^\infty \mathbf{x}^\top (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) \mathbf{x} dt. \quad (6.23)$$

Since we have set $\mathbf{u} = -\mathbf{K}\mathbf{x}$, the dynamics (6.15) become $\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}$, and $\mathbf{x}(t)$ is given by

$$\mathbf{x}(t) = e^{(\mathbf{A} - \mathbf{B}\mathbf{K})t} \mathbf{x}_0.$$

Substituting into (6.23), we have

$$\begin{aligned} J &= \int_0^\infty \mathbf{x}_0^\top e^{(\mathbf{A} - \mathbf{B}\mathbf{K})^\top t} (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) e^{(\mathbf{A} - \mathbf{B}\mathbf{K})t} \mathbf{x}_0 dt \\ &= \mathbf{x}_0^\top \mathbf{P} \mathbf{x}_0 \end{aligned}$$

where the $n \times n$ symmetric matrix \mathbf{P} is given by

$$\mathbf{P} = \int_0^\infty e^{(\mathbf{A} - \mathbf{B}\mathbf{K})^\top t} (\mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K}) e^{(\mathbf{A} - \mathbf{B}\mathbf{K})t} dt. \quad (6.24)$$

Next, we require the following theorem, due to Lyapunov, the proof of which is given at the end of this section:

Theorem 6.17. *Suppose all the eigenvalues of \mathbf{A} have negative real parts. Then the linear matrix equation*

$$\mathbf{A}^\top \mathbf{X} + \mathbf{X} \mathbf{A} + \mathbf{Q} = 0 \quad (6.25)$$

(called a Lyapunov equation) has the unique solution

$$\mathbf{X} = \int_0^\infty e^{\mathbf{A}^\top t} \mathbf{Q} e^{\mathbf{A}t} dt. \quad (6.26)$$

Thus, assuming the closed-loop is stable (so that eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}$ are in the left half plane) the matrix \mathbf{P} defined by (6.24) is the unique solution to the linear equation

$$(\mathbf{A} - \mathbf{B}\mathbf{K})^\top \mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{B}\mathbf{K}) + \mathbf{Q} + \mathbf{K}^\top \mathbf{R} \mathbf{K} = 0. \quad (6.27)$$

Differentiating this expression with respect to \mathbf{K} , in the direction of a matrix $\delta\mathbf{K}$, and denoting the corresponding variation in \mathbf{P} by $\delta\mathbf{P}$, we have

$$-(\mathbf{B} \cdot \delta\mathbf{K})^\top \mathbf{P} + (\mathbf{A} - \mathbf{BK})^\top \delta\mathbf{P} + \delta\mathbf{P}(\mathbf{A} - \mathbf{BK}) - \mathbf{PB} \cdot \delta\mathbf{K} + (\delta\mathbf{K})^\top \mathbf{RK} + \mathbf{K}^\top \mathbf{R} \cdot \delta\mathbf{K} = 0.$$

Setting $\delta\mathbf{P} = \mathbf{0}$ (which must be true where J is a minimum), we have

$$(\delta\mathbf{K})^\top (-\mathbf{B}^\top \mathbf{P} + \mathbf{RK}) + (-\mathbf{PB} + \mathbf{K}^\top \mathbf{R}) \delta\mathbf{K} = 0.$$

(Note that the second expression is the transpose of the first, since \mathbf{P} and \mathbf{R} are symmetric.) We wish this equality to hold for all directions of perturbation $\delta\mathbf{K}$, and so we require

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}. \quad (6.28)$$

Finally, we substitute this expression for \mathbf{K} back into (6.27), to obtain

$$(\mathbf{A} - \mathbf{BR}^{-1} \mathbf{B}^\top \mathbf{P})^\top \mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{BR}^{-1} \mathbf{B}^\top \mathbf{P}) + \mathbf{Q} + (\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P})^\top \mathbf{R} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P} = \mathbf{0}.$$

Since \mathbf{P} is symmetric, this simplifies to

$$\mathbf{A}^\top \mathbf{P} + \mathbf{PA} - \mathbf{PBR}^{-1} \mathbf{B}^\top \mathbf{P} + \mathbf{Q} = \mathbf{0}, \quad (6.29)$$

which is called an *algebraic Riccati equation*. It can be shown that as long as the system (6.15) is controllable and the matrices \mathbf{Q} and \mathbf{R} are positive definite, there is a unique positive-definite solution for the matrix \mathbf{P} . (Actually, as mentioned on page 190, \mathbf{Q} does not need to be positive definite: it can be positive semi-definite, as long as a certain observability condition is satisfied: in particular, writing $\mathbf{Q} = \mathbf{CC}^\top$, the pair (\mathbf{A}, \mathbf{C}) must be observable.)

Overall, to solve the optimal control problem, one first finds a positive-definite solution \mathbf{P} to the Riccati equation (6.29), and then the optimal gain \mathbf{K} is given by (6.28).

Proof of Theorem 6.17

Proof. First, we show that \mathbf{X} given by (6.26) satisfies the Lyapunov equation (6.25). Let $\mathbf{Z}(t) = e^{\mathbf{A}^\top t} \mathbf{Q} e^{\mathbf{A}t}$. Then

$$\frac{d\mathbf{Z}}{dt} = \mathbf{A}^\top e^{\mathbf{A}^\top t} \mathbf{Q} e^{\mathbf{A}t} + e^{\mathbf{A}^\top t} \mathbf{Q} e^{\mathbf{A}t} \mathbf{A} = \mathbf{A}^\top \mathbf{Z} + \mathbf{ZA}.$$

Integrating from zero to infinity, we have

$$\mathbf{Z}(\infty) - \mathbf{Z}(0) = \int_0^\infty (\mathbf{A}^\top \mathbf{Z} + \mathbf{ZA}) dt.$$

Defining $\mathbf{X} = \int_0^\infty \mathbf{Z}(t) dt$ and noting that $\mathbf{Z}(\infty) = 0$ (since the eigenvalues of \mathbf{A} are in the left half-plane), we have

$$-\mathbf{Q} = \mathbf{A}^\top \mathbf{X} + \mathbf{X} \mathbf{A},$$

and so \mathbf{X} satisfies (6.25).

To establish uniqueness, consider the linear operator $L : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ given by $L(\mathbf{X}) = \mathbf{A}^\top \mathbf{X} + \mathbf{X} \mathbf{A}$. We have shown above that, for any matrix \mathbf{Q} , the equation $L(\mathbf{X}) = -\mathbf{Q}$ has a solution given by (6.26), so the range of L is all of $\mathbb{R}^{n \times n}$, and therefore its rank is n^2 . Since the domain of L has dimension n^2 as well, the nullspace of L must be trivial, by the rank and nullity theorem, and so the solution is unique. \square



Optimal control of nonlinear systems

Next, we give a more general derivation of the optimal control for a nonlinear system given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (6.30)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$. Suppose we want to minimize the functional

$$J = \int_0^T \frac{1}{2} (\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt + \frac{1}{2} \mathbf{x}(T)^\top \mathbf{Q}_f \mathbf{x}(T) \quad (6.31)$$

where \mathbf{Q} , \mathbf{Q}_f , and \mathbf{R} are symmetric, positive-definite matrices. This cost is the same as that used in (6.16) for the LQR problem, except that we consider a finite time interval $t \in [0, T]$, and the term involving \mathbf{Q}_f represents a *terminal cost*, or a penalty on the state not being zero at time T . Consider this as a constrained optimization problem, to minimize (6.31) subject to the constraint (6.30). To solve this, introduce a Lagrange multiplier $\boldsymbol{\lambda}(t) \in \mathbb{R}^n$, defined for $t \in [0, T]$, and consider the augmented cost function

$$H = J + \int_0^T \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}}) dt.$$

Then by the Lagrange multiplier theorem, necessary conditions for optimality are

$$\frac{\partial H}{\partial \boldsymbol{\lambda}} = 0, \quad \frac{\partial H}{\partial \mathbf{x}} = 0, \quad \frac{\partial H}{\partial \mathbf{u}} = 0.$$

Taking directional derivatives, we have

$$\begin{aligned} \frac{\partial H}{\partial \boldsymbol{\lambda}} \cdot \delta \boldsymbol{\lambda} &= \int_0^T \delta \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}}) dt = 0, \quad \forall \delta \boldsymbol{\lambda} \\ \iff \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) - \dot{\mathbf{x}}(t) &= 0 \quad \forall t \in [0, T]. \end{aligned} \quad (6.32)$$

Similarly,

$$\begin{aligned}
 \frac{\partial H}{\partial \mathbf{u}} \cdot \delta \mathbf{u} &= \int_0^T \left(\frac{1}{2} (\delta \mathbf{u}^\top \mathbf{R} \mathbf{u} + \mathbf{u}^\top \mathbf{R} \delta \mathbf{u}) + \boldsymbol{\lambda}^\top \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \delta \mathbf{u} \right) dt \\
 &= \int_0^T \delta \mathbf{u}^\top \left(\mathbf{R} \mathbf{u} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda} \right) dt = 0, \quad \forall \delta \mathbf{u} \\
 &\iff \mathbf{R} \mathbf{u}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \bigg|_{(\mathbf{x}(t), \mathbf{u}(t))}^\top \cdot \boldsymbol{\lambda}(t) = 0 \quad \forall t \in [0, T], \quad (6.33)
 \end{aligned}$$

and

$$\begin{aligned}
 \frac{\partial H}{\partial \mathbf{x}} \cdot \delta \mathbf{x} &= \int_0^T \frac{1}{2} (\delta \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{x}^\top \mathbf{Q} \delta \mathbf{x}) + \boldsymbol{\lambda}^\top \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} - \dot{\delta \mathbf{x}} \right) dt \\
 &\quad + \frac{1}{2} (\delta \mathbf{x}(T)^\top \mathbf{Q}_f \mathbf{x}(T) + \mathbf{x}(T)^\top \mathbf{Q}_f \delta \mathbf{x}(T)) \\
 &= \int_0^T \delta \mathbf{x}^\top \left(\mathbf{Q} \mathbf{x} + \dot{\boldsymbol{\lambda}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda} \right) dt - \boldsymbol{\lambda}^\top \delta \mathbf{x} \bigg|_0^T + \mathbf{x}(T)^\top \mathbf{Q}_f \delta \mathbf{x}(T) = 0
 \end{aligned}$$

for all $\delta \mathbf{x}$ with $\delta \mathbf{x}(0) = 0$. The boundary terms vanish if we choose $\boldsymbol{\lambda}(T) = \mathbf{Q}_f \mathbf{x}(T)$, and we are left with

$$\dot{\boldsymbol{\lambda}}(t) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \bigg|_{\mathbf{x}(t), \mathbf{u}(t)}^\top \boldsymbol{\lambda}(t) + \mathbf{Q} \mathbf{x}(t) = 0, \quad \boldsymbol{\lambda}(T) = \mathbf{Q}_f \mathbf{x}(T). \quad (6.34)$$

To summarize, equations (6.32–6.34) may be written

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (6.35)$$

$$-\dot{\boldsymbol{\lambda}} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^\top \boldsymbol{\lambda} + \mathbf{Q} \mathbf{x}, \quad \boldsymbol{\lambda}(T) = \mathbf{Q}_f \mathbf{x}(T) \quad (6.36)$$

$$\mathbf{u} = -\mathbf{R}^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)^\top \boldsymbol{\lambda}, \quad (6.37)$$

where at each time, derivatives of \mathbf{f} are evaluated at the point $(\mathbf{x}(t), \mathbf{u}(t))$.



Solving the equations

In general, these equations must be solved iteratively, for instance using a gradient descent method, as follows: guess an input $\mathbf{u}_k(t)$, integrate $\mathbf{x}(t)$ from time $t = 0$ to T using (6.35), then linearize about this trajectory and integrate (6.36) backward from time $t = T$ to 0, and then guess a new input $\mathbf{u}_{k+1}(t)$ using (6.37). In fact, this procedure corresponds to a Newton-Raphson iteration to find zeros of the gradient $(\partial H / \partial \mathbf{u})^*$. The Newton-Raphson update is given by

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \left(\frac{\partial}{\partial \mathbf{u}} \frac{\partial H}{\partial \mathbf{u}} \bigg|_k^* \right)^{-1} \cdot \frac{\partial H}{\partial \mathbf{u}} \bigg|_k^*$$

where

$$\left. \frac{\partial H}{\partial \mathbf{u}} \right|_k^* = \mathbf{R}\mathbf{u}_k + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_k^\top \cdot \boldsymbol{\lambda}_k.$$

If \mathbf{f} is linear in \mathbf{u} , then this becomes

$$\begin{aligned} \mathbf{u}_{k+1}(t) &= \mathbf{u}_k(t) - \mathbf{R}^{-1} \left(\mathbf{R}\mathbf{u}_k(t) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_k^\top \boldsymbol{\lambda}_k(t) \right) \\ &= -\mathbf{R}^{-1} \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_k^\top \boldsymbol{\lambda}_k, \end{aligned}$$

which corresponds to updating $\mathbf{u}(t)$ using (6.37). Note that, if \mathbf{f} is linear in \mathbf{u} , the second derivative is

$$\frac{\partial}{\partial \mathbf{u}} \left(\frac{\partial H}{\partial \mathbf{u}} \right)^* = \mathbf{R}$$

so if \mathbf{R} is not positive definite, then we do not have a local minimum. For instance, this procedure does not work if there is no weight on the input (i.e., $\mathbf{R} = \mathbf{0}$).

For many control problems, the iteration above is too computationally intensive to be practical. Fortunately, if the dynamics are linear, then the solution is simpler and iteration is not needed.



Special case: linear systems

As a special case, if the dynamics are linear, of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad (6.38)$$

then $\partial \mathbf{f} / \partial \mathbf{x} = \mathbf{A}(t)$ and $\partial \mathbf{f} / \partial \mathbf{u} = \mathbf{B}(t)$. If we then write $\boldsymbol{\lambda}(t) = \mathbf{P}(t)\mathbf{x}(t)$, where $\mathbf{P}(t)$ is a symmetric, positive-definite matrix, then (6.36) becomes

$$-\dot{\mathbf{P}}\mathbf{x} - \mathbf{P}\dot{\mathbf{x}} = \mathbf{A}^\top \mathbf{P}\mathbf{x} + \mathbf{Q}\mathbf{x}. \quad (6.39)$$

The optimal control law (6.37) is then $\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}(t)^\top \mathbf{P}(t)\mathbf{x}(t)$, so the dynamics (6.38) become

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P})\mathbf{x}$$

and substituting this into (6.39) gives

$$-\dot{\mathbf{P}} = \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} + \mathbf{Q}, \quad \mathbf{P}(T) = \mathbf{Q}_f. \quad (6.40)$$

This quadratic matrix equation is called a *differential Riccati equation*, and it may be solved *backward* in time (starting at $t = T$) to determine $\mathbf{P}(t)$. The optimal input is then given by $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}$, where the time-varying gain is

$$\mathbf{K}(t) = \mathbf{R}^{-1}\mathbf{B}(t)^\top \mathbf{P}(t).$$

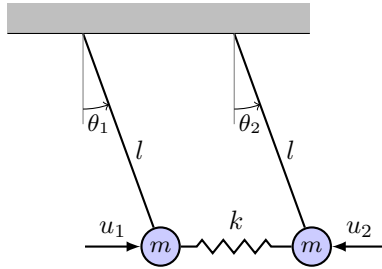


Figure 6.12: Coupled pendulums, for Exercise 6.3.

The matrix \mathbf{P} has meaning as well: the scalar quantity $V(\mathbf{x}, t) = \frac{1}{2}\mathbf{x}^T\mathbf{P}(t)\mathbf{x}$ represents the “cost to go” from the state \mathbf{x} and time t . To see this, one shows that

$$\frac{d}{dt}\mathbf{x}(t)^T\mathbf{P}(t)\mathbf{x}(t) = -\mathbf{x}^T(\mathbf{Q} + \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P})\mathbf{x} = -(\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}),$$

and hence, for all t_1 and t_2 ,

$$\mathbf{x}(t_1)^T\mathbf{P}(t_1)\mathbf{x}(t_1) = \int_{t_1}^{t_2} (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}) dt + \mathbf{x}(t_2)^T\mathbf{P}(t_2)\mathbf{x}(t_2).$$

Special case: Linear time-invariant systems

We can simplify even further if the matrices \mathbf{A} and \mathbf{B} do not depend on time, and if we let the time $T \rightarrow \infty$. In this case, the solution $\mathbf{P}(t)$ of (6.40) goes to a steady state as t marches backwards in time. At steady state we have $\dot{\mathbf{P}} = 0$, and so the constant, symmetric \mathbf{P} satisfies the *algebraic Riccati equation*

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = 0,$$

as we derived earlier in (6.29). The optimal gain is still $\mathbf{u} = -\mathbf{K}\mathbf{x}$, where the gain \mathbf{K} (now constant) is given by $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$.

Exercises

- 6.1.** Show that, if a system is controllable in the sense of Definition 6.3, then one can find an input $u(t)$ to drive the system from *any* initial state \mathbf{x}_0 (not just the origin) at time $t = 0$ to any final state \mathbf{x}_f at time $T > 0$.
- 6.2.** For the sprung beam considered in Example 6.5, consider the case $\delta \neq 0$ and show that the system is controllable unless $r = 0$ or $J = 1$. In each case, find the reachable subspace.

- 6.3.** Two pendulums, coupled by a spring, are to be controlled using applied forces u_1 and u_2 on the bobs, as shown in Figure 6.12. The system's linearized equations of motion are

$$\begin{aligned} ml^2\ddot{\theta}_1 &= -kl^2(\theta_1 - \theta_2) - mgl\theta_1 + lu_1 \\ ml^2\ddot{\theta}_2 &= -kl^2(\theta_2 - \theta_1) - mgl\theta_2 - lu_2. \end{aligned}$$

- Write the system in the state space form $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, where the state vector is $\mathbf{x} = (\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$ and the control input is $\mathbf{u} = (u_1, u_2)$.
- Can any arbitrary state $\mathbf{x} = (\alpha_1, \alpha_2, \beta_1, \beta_2)$ be reached from $\mathbf{x}(0) = \mathbf{0}$?
- Consider the case where only u_1 is available, and u_2 is fixed at 0. Can any arbitrary state $\mathbf{x} = (\alpha_1, \alpha_2, \beta_1, \beta_2)$ be reached from $\mathbf{x}(0) = \mathbf{0}$?
- Consider the case that we are restricted to a single control input of the form $u = u_1 = u_2$. Can any arbitrary state $\mathbf{x} = (\alpha_1, \alpha_2, \beta_1, \beta_2)$ be reached from $\mathbf{x}(0) = \mathbf{0}$? If not, physically describe the reachable subspace.

- 6.4.** Consider the system

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u, \quad y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x},$$

with the control law

$$u = -k_1x_1 - k_2x_2.$$

Show that the eigenvalues of the system cannot be assigned to arbitrary values. What effect can state feedback have on the eigenvalues of this system?

- 6.5.** Consider the double integrator defined by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u. \end{aligned}$$

Determine gains k_1, k_2 such that the state feedback $u = -k_1x_1 - k_2x_2$ moves the closed-loop poles to $-1 \pm i$.

- 6.6.** Consider a state-space system $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, with

$$\mathbf{A} = \begin{bmatrix} -200 & -15 & -12 \\ 32 & 0 & 0 \\ 0 & 8 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}.$$

We wish to design a state-feedback controller $u = -\mathbf{Kx}$, with \mathbf{K} chosen by pole placement. Compare the following choices of closed-loop pole locations:

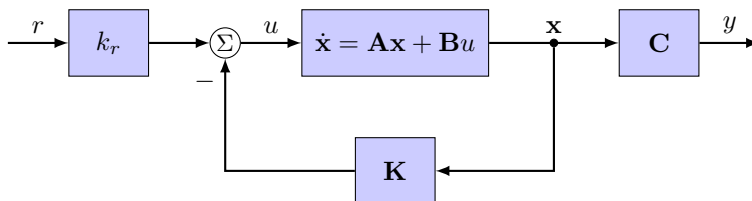
- (a) $-5 \pm 5i, -10$.
- (b) $-5 \pm 5i, -200$.
- (c) $-1 \pm i, -10$.

Which would you choose, and why? Justify your answer with suitable Bode and/or Nyquist plots.

- 6.7.** A state-space realization for a spring-mass damper as shown in Example 5.6 is given by $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, $y = \mathbf{C}\mathbf{x}$, with

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & -b \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ b \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix},$$

where the input u is the position of the wall, the output y is the position of the mass, and b is the amount of damping. We wish to design a controller to track a given reference position r , using the setup in the following diagram:



- (a) For what values of b is this realization controllable?
- (b) Choose appropriate closed-loop pole locations so that the closed-loop system has a step response with a rise time of about 2.5 seconds, and an overshoot less than 10%.
- (c) Take $b = 1$. Using pole placement, find gains \mathbf{K} and k_r so that the closed-loop poles lie at the locations determined in part (b), and so there is no steady-state error to a step reference. Simulate the response of the closed-loop system to a step reference. Does your controller behave as expected?
- (d) Now suppose we want a faster controller, with a rise time of 0.1 sec. Find appropriate pole locations, and repeat part (c) for these pole locations. Does your controller behave as expected and meet the specifications? If not, explain why not, and try adjusting the desired closed-loop pole locations, to obtain a controller that does meet the specifications.
- (e) Plot the Nyquist plots of the loop gain, for the two controllers you obtained. Comment on the stability margins. Do either of the controllers have problems with robustness?

- 6.8.** Show that the closed-loop transfer function from r to y in the block diagram in Figure 6.2 may be written

$$\frac{k_r P(s)}{1 + L(s)},$$

where $P(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is the plant transfer function and $L(s) = \mathbf{K}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is the loop transfer function.

- 6.9.** For each cost function J below, find symmetric matrices \mathbf{Q} and \mathbf{R} such that, for $\mathbf{x} = (x_1, x_2)$ and $\mathbf{u} = (u_1, u_2)$,

$$J(\mathbf{x}, \mathbf{u}) = \int_0^\infty (\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt.$$

(a) $J(\mathbf{x}, \mathbf{u}) = \int_0^\infty (3x_2^2 + 2u_1^2 + u_2^2) dt.$

(b) $J(\mathbf{x}, \mathbf{u}) = \int_0^\infty (ax_1^2 + (x_1 + bx_2)^2 + 10u_1^2 + 2u_2^2) dt.$ For what values of a and b does the cost function satisfy the requirements of the LQR problem?

(c) $J(\mathbf{x}, \mathbf{u}) = \int_0^\infty (\mathbf{y}^\top \mathbf{M} \mathbf{y} + 2u_1^2 + 5u_2^2) dt$, where $\mathbf{y} = \mathbf{C}\mathbf{x}$ is the sensor output and \mathbf{M} is a symmetric positive-definite matrix.

- 6.10.** Consider the one-dimensional linear state-space equation

$$\dot{x} = ax + bu.$$

The LQR cost function is

$$J(x, u) = \int_0^\infty (q x(t)^2 + r u(t)^2) dt,$$

where $q, r > 0$.

- (a) Consider a control law of the form $u = -kx$. Assuming that the closed-loop system is stable, show that the cost function has the form

$$J = p x(0)^2,$$

where p satisfies the equation

$$2p(a - bk) + q + rk^2 = 0. \quad (6.41)$$

- (b) Recall from elementary calculus that a necessary condition for a continuously differentiable function $J(k)$ to have a minimum is

that $dJ/dk = 0$. Show that the gain k that minimizes this cost function, for any $x(0)$, satisfies

$$k = \frac{bp}{r}.$$

Also show that p is a solution of the scalar continuous algebraic Riccati equation,

$$a^\top p + pa - pbr^{-1}b^\top p + q = 0. \quad (6.42)$$

[*Hint:* When showing the minimization, it is easier to differentiate (6.41) implicitly than to solve for p and compute dJ/dk explicitly.]

- (c) Equation (6.42) has a positive and a negative solution for p . Why must we take the positive solution for the optimal control problem? What would happen to the closed-loop system if the negative solution were used?
- (d) Consider the “minimum energy” case, in the limit as $q \rightarrow 0$. Show that this leads to the following possibilities:
- If the plant is stable, then the optimal gain is $k = 0$.
 - If the plant is unstable, then the closed-loop pole is the plant pole reflected about the imaginary axis.

6.11. Let $\mathbf{L}(s) = \mathbf{K}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ be the loop gain, where \mathbf{K} is the solution of an LQR problem, and let $\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, where $\mathbf{Q} = \mathbf{C}^\top\mathbf{C}$. By suitably manipulating the Riccati equation, obtain the *Kalman equality*

$$(\mathbf{I} + \mathbf{L}(-s)^\top)\mathbf{R}(\mathbf{I} + \mathbf{L}(s)) = \mathbf{R} + \mathbf{G}(-s)^\top\mathbf{G}(s),$$

and use it to prove (6.19) for a system with a single input. [*Hint:* Start with the Riccati equation, add and subtract $s\mathbf{P}$, and multiply on the right by $(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ and on the left by $((-s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B})^\top$.]

Chapter 7

Output feedback

In the last chapter, we saw some useful ways of designing controllers using state feedback of the form $\mathbf{u} = -\mathbf{K}\mathbf{x}$. In particular, given a controllable linear system, with a suitable choice of \mathbf{K} , we can place the closed-loop poles anywhere we like in the complex plane; or, using LQR, we can find an optimal controller that minimizes a quadratic cost function of our choice.

However, there is a catch with these methods: in order to compute the feedback law $\mathbf{u} = -\mathbf{K}\mathbf{x}$, we need to know the state \mathbf{x} , and usually we cannot measure the entire state $\mathbf{x}(t)$; we have access only to the outputs $\mathbf{y}(t)$. This then raises the following question: can we reconstruct or estimate the state $\mathbf{x}(t)$ given only measurements $\mathbf{y}(t)$ (and the input $\mathbf{u}(t)$, which of course we choose)? The answer, fortunately, is *yes*, as long as the system is *observable*. In the next section, we quantify this notion of observability, and in Section 7.2, we describe a practical way of reconstructing the state, using an *observer*. We then investigate what happens when we use this state estimate in a feedback loop, in conjunction with a state-feedback controller.

7.1 Observability

As before, we will consider a linear time-invariant (LTI) system of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, & \mathbf{x}(0) &= \mathbf{x}_0 \in \mathbb{R}^n \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u},\end{aligned}\tag{7.1}$$

where the input $\mathbf{u}(t)$ and output $\mathbf{y}(t)$ may be vectors. We first quantify what it means for this system to be *observable*. Intuitively, a system is observable if we are able to reconstruct the state $\mathbf{x}(t)$ given only the measurements $\mathbf{y}(t)$. We also assume we know the inputs $\mathbf{u}(t)$ —in a real control system, we always know $\mathbf{u}(t)$, since it is the controller’s job to provide $\mathbf{u}(t)$. The precise definition of observability below involves reconstructing only the *initial* state:

Definition 7.1. The system (7.1) is *observable* if the initial state \mathbf{x}_0 can be deduced from knowledge of $\mathbf{u}(t)$ and $\mathbf{y}(t)$ for $0 < t < T$, for any $T > 0$.

A system is *unobservable* if it is not observable. Note that if we know \mathbf{x}_0 , then $\mathbf{x}(t)$ can be deduced for all time, from (5.8):

$$\mathbf{x}(t) = e^{\mathbf{A}t} \mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau. \quad (7.2)$$

This is not how we reconstruct the state in practice—see §7.2 for a better way—but it is possible in theory.

It turns out that observability of the system (7.1) does not depend on the input $\mathbf{u}(t)$, or even on the \mathbf{B} and \mathbf{D} matrices. Therefore, when considering observability, we can restrict ourselves to the case $\mathbf{u} = \mathbf{0}$. This is a simple consequence of superposition: the response $\mathbf{y}(t)$ that satisfies (7.1) is the sum of the zero-input response with $\mathbf{x}(0) = \mathbf{x}_0$ (the first term of (7.2)) and the zero-initial-state response with input $\mathbf{u}(t)$ (the second term of (7.2)). Since we know $\mathbf{u}(t)$, we can always determine the zero-initial-state response, and subtract this from $\mathbf{y}(t)$, to get the corresponding output $\mathbf{y}(t)$ with $\mathbf{u}(t) = \mathbf{0}$. Thus, observability of the system (7.1) depends only on the \mathbf{A} and \mathbf{C} matrices, and one sometimes refers to this as observability of the *pair* (\mathbf{A}, \mathbf{C}) .

Unobservable states

Definition 7.1 explains what it means for a *system* to be observable. We now define a related notion, that of an unobservable *state*. This concept will be useful both in developing an intuitive understanding of observability, and for deriving a convenient test for observability.

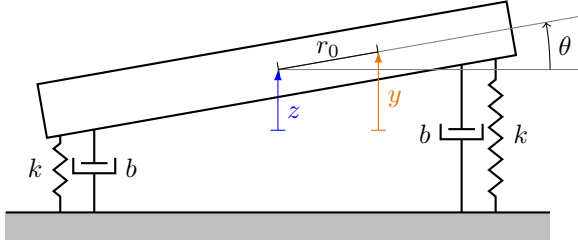
Definition 7.2. A state \mathbf{x}_0 is *unobservable* if the zero-input solution $\mathbf{y}(t)$ with $\mathbf{x}(0) = \mathbf{x}_0$ is zero for all $t \geq 0$:

$$\mathbf{x}_0 \text{ unobservable} \iff \mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t} \mathbf{x}_0 = \mathbf{0} \quad \text{for all } t \geq 0.$$

(Compare this expression with the corresponding expressions for an uncontrollable state, in Theorem 6.7.) This notion is indeed quite intuitive: a state is unobservable if the sensors read zero for all time. We illustrate with some examples below.

Example 7.3. Consider a DC motor, with the state $\mathbf{x} = (\theta, \dot{\theta})$, corresponding to the shaft's angle θ and angular velocity $\dot{\theta}$. Suppose the sensor is a tachometer, which measures the shaft speed, $y = \dot{\theta}$. Then the state $\mathbf{x}_0 = (1, 0)$ is unobservable: it corresponds to a nonzero initial angle $\theta = 1$, but a zero initial velocity $\dot{\theta} = 0$. Since the shaft is initially at rest, it will remain at rest (since there is no input), and the sensor measurement will be $y = 0$ for all time. Therefore, the state $(1, 0)$ is unobservable, as is any state $(\theta, 0)$. \diamond

Example 7.4 (Sprung beam). Consider the sprung beam, as in Example 6.5, but now with an output $y(t)$ corresponding to the height of the beam offset a distance r_0 from the center of mass, as shown in the diagram below:



We nondimensionalize all quantities as in Example 6.5, and choose the state $\mathbf{x} = (z, \theta, \dot{z}, \dot{\theta})$, where z is the height of the center of mass (normalized by the beam half-length L), and θ is the angle of the beam relative to the horizontal. The output is then $y = z + r \sin \theta \approx z + r\theta$ for small θ . The linearized equations are given by

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -\delta & 0 \\ 0 & -1/J & 0 & -\delta/J \end{bmatrix} \mathbf{x}$$

$$y = [1 \quad r \quad 0 \quad 0] \mathbf{x}.$$

Does this system have any unobservable states? Consider first the case $r = 0$. Then we expect intuitively that there should be unobservable states $\mathbf{x}_0 = (0, 1, 0, 0)$ (corresponding to a nonzero angle θ) and $\mathbf{x}_0 = (0, 0, 0, 1)$ (corresponding to a nonzero $\dot{\theta}$), for if the system is released with these initial conditions, the height z will be zero for all time, and so $y(t) = z(t) = 0$ for all time.

What about the case when $r \neq 0$? Are there unobservable states? And is the system observable? The answers to these questions are not so clear at this point. Intuitively, if we have two outputs $\mathbf{y} = (z, \theta)$, we would expect the system to be observable, for then we can figure out the initial state $(z, \theta, \dot{z}, \dot{\theta})$ simply by differentiating \mathbf{y} . But if $y = z + r\theta$, then can we determine z and θ separately (along with their derivatives)? This seems like a challenging task. \diamond

These two notions of observability seem initially to be quite different: according to Definition 7.1, a *system* is observable if we can reconstruct the initial state \mathbf{x}_0 given the output $\mathbf{y}(t)$ (and input $\mathbf{u}(t)$), while according to Definition 7.2 a *state* \mathbf{x}_0 is unobservable if the resulting output $\mathbf{y}(t)$ (with $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbf{u}(t) = 0$) is zero for all t . The following theorem provides an important connection between these concepts. First, we define an $n \times n$

matrix called the *observability Gramian*:

$$\mathbf{W}_o(T) := \int_0^T e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} dt. \quad (7.3)$$

Theorem 7.5 (Observability). *The following are equivalent:*

1. *The system (7.1) is observable.*
2. *The system (7.1) has no unobservable states.*
3. *The observability Gramian $\mathbf{W}_o(T)$ defined by (7.3) is positive definite for all $T > 0$.*

Proof. We show $(1) \implies (2) \implies (3) \implies (1)$.

To show $(1) \implies (2)$, we show the contrapositive: that is, we assume there *is* an unobservable state $\mathbf{x}_0 \neq \mathbf{0}$, and show that the system is *not* observable. This is straightforward to see: consider the zero-input response $\mathbf{y}(t)$ for an initial condition $\mathbf{v} \in \mathbb{R}^n$. But since \mathbf{x}_0 is an unobservable state, this is the same as the zero-input response $\mathbf{y}(t)$ for the initial condition $\mathbf{x}(0) = \mathbf{v} + \mathbf{x}_0$ (by superposition, since the response with initial condition $\mathbf{x}(0) = \mathbf{x}_0$ is $\mathbf{y}(t) = \mathbf{0}$). Thus, using only the output $\mathbf{y}(t)$, we cannot distinguish between the initial conditions \mathbf{v} and $\mathbf{v} + \mathbf{x}_0$, so the system is unobservable.

To show $(2) \implies (3)$, we assume there are no unobservable states, and show that for any $T > 0$, $\mathbf{W}_o(T)$ is positive definite, and hence invertible (see, for instance, Theorem A.5). Fix $T > 0$ and consider the product

$$\mathbf{x}_0^\top \mathbf{W}_o(T) \mathbf{x}_0 = \int_0^T \mathbf{x}_0^\top e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} \mathbf{x}_0 dt,$$

where \mathbf{x}_0 is an arbitrary vector in \mathbb{R}^n . Letting $\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0$, this becomes

$$\int_0^T \mathbf{y}(t)^\top \mathbf{y}(t) dt = \int_0^T \|\mathbf{y}(t)\|^2 dt \geq 0,$$

so $\mathbf{W}_o(T)$ is at least positive semidefinite. If there is some $\mathbf{x}_0 \neq \mathbf{0}$ for which this integral is zero, then $\mathbf{y}(t)$ must be zero for all $t \in [0, T]$, and thus \mathbf{x}_0 is an unobservable state. But we have assumed that there are no unobservable states, so $\mathbf{x}_0^\top \mathbf{W}_o(T) \mathbf{x}_0 > 0$ for all $\mathbf{x}_0 \neq \mathbf{0}$, and thus $\mathbf{W}_o(T)$ is positive definite.

Finally, to show $(3) \implies (1)$, let \mathbf{x}_0 be an arbitrary initial condition in (7.1), with $\mathbf{u}(t) = \mathbf{0}$. The corresponding output $\mathbf{y}(t)$ is then

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0.$$

We show that, if $\mathbf{W}_o(T)$ is invertible, we may solve uniquely for \mathbf{x}_0 , given $\mathbf{y}(t)$ for $0 < t < T$. Multiplying both sides by $(\mathbf{C}e^{\mathbf{A}t})^\top = e^{\mathbf{A}^\top t} \mathbf{C}^\top$ and integrating from 0 to T gives

$$\int_0^T e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{y}(t) dt = \int_0^T e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} \mathbf{x}_0 dt = \mathbf{W}_o(T) \mathbf{x}_0,$$

so since $\mathbf{W}_o(T)$ is invertible, we may solve uniquely for \mathbf{x}_0 in terms of \mathbf{y} , as

$$\mathbf{x}_0 = \mathbf{W}_o(T)^{-1} \int_0^T e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{y}(t) dt,$$

and thus the system is observable. \square

The observability Gramian defined above can be used to quantify *how observable* a particular state is. Let us fix $T > 0$, and for an output $\mathbf{y}(t)$, define

$$\|\mathbf{y}\|^2 := \int_0^T \mathbf{y}(t)^\top \mathbf{y}(t) dt.$$

If $\mathbf{y}(t)$ is the solution of (7.1) with $\mathbf{u}(t) = \mathbf{0}$ and initial condition $\mathbf{x}(0) = \mathbf{x}_0$, then $\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0$, and so

$$\|\mathbf{y}\|^2 = \int_0^T \mathbf{x}_0^\top e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} \mathbf{x}_0 dt = \mathbf{x}_0^\top \mathbf{W}_o(T) \mathbf{x}_0.$$

Given a state \mathbf{x}_0 , the product $\mathbf{x}_0^\top \mathbf{W}_o(T) \mathbf{x}_0$ therefore describes how large the corresponding output $\mathbf{y}(t)$ is, and we may consider states for which this product is large to be “more observable” than states for which this product is small. Of course, if this product is zero, then the output is zero, and \mathbf{x}_0 is an unobservable state; similarly, if this product is very small, then we may consider the state \mathbf{x}_0 to be “nearly unobservable.” This discussion is precisely analogous to the discussion of controllability on page 173.

Observability matrix

The above connection between observability of a system and the existence of unobservable states lets us obtain a useful test for observability.

Theorem 7.6. *The system (7.1) is observable if and only if the observability matrix*

$$\mathbf{O} := \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (7.4)$$

has full rank ($\text{rank } \mathbf{O} = n$).

Proof. Suppose $\mathbf{x}_0 \neq \mathbf{0}$ is an unobservable state. Then

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_0 = \mathbf{0}, \quad \forall t.$$

In particular, $\mathbf{y}(t)$ is zero at $t = 0$, so

$$\mathbf{C}e^{\mathbf{A} \cdot 0} \mathbf{x}_0 = \mathbf{C}\mathbf{x}_0 = \mathbf{0}. \quad (7.5a)$$

Furthermore, all of the derivatives are zero, so

$$\begin{aligned}\dot{\mathbf{y}}(0) &= \mathbf{CA}e^{\mathbf{A}\cdot 0}\mathbf{x}_0 = \mathbf{CA}\mathbf{x}_0 = \mathbf{0} \\ \ddot{\mathbf{y}}(0) &= \mathbf{CA}^2e^{\mathbf{A}\cdot 0}\mathbf{x}_0 = \mathbf{CA}^2\mathbf{x}_0 = \mathbf{0} \\ &\vdots \\ \mathbf{y}^{(n-1)}(t) &= \mathbf{CA}^{n-1}e^{\mathbf{A}\cdot 0}\mathbf{x}_0 = \mathbf{CA}^{n-1}\mathbf{x}_0 = \mathbf{0}.\end{aligned}\tag{7.5b}$$

Writing (7.5) together in matrix form gives

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \mathbf{x}_0 = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}.\tag{7.6}$$

Note that the matrix on the left-hand side of (7.6) is $\mathbf{\Theta}$. Thus, if $\text{rank } \mathbf{\Theta} = n$, then by the rank and nullity theorem, the nullspace of $\mathbf{\Theta}$ is $\{\mathbf{0}\}$, so (7.6) has a unique solution $\mathbf{x}_0 = \mathbf{0}$. Thus, there are no unobservable states, and hence the system (7.1) is observable. On the other hand, if $\text{rank } \mathbf{\Theta} < n$, then the nullspace of $\mathbf{\Theta}$ is nontrivial, so (7.6) has a nonzero solution \mathbf{x}_0 . It turns out that \mathbf{x}_0 is an unobservable state, and thus the system is unobservable. In order to show that \mathbf{x}_0 is an unobservable state, there is one more technicality: we need to show that $\mathbf{CA}^k = \mathbf{0}$ for *all* k , while (7.6) implies only that $\mathbf{CA}^k\mathbf{x}_0 = \mathbf{0}$ for $k = 0, \dots, n-1$. As in the proof of Theorem 6.8, however, we may apply the Cayley-Hamilton theorem (Corollary A.7) to see that \mathbf{A}^n may be written as a linear combination of $\mathbf{A}^0, \mathbf{A}^1, \dots, \mathbf{A}^{n-1}$. It then follows that $\mathbf{CA}^n\mathbf{x}_0 = \mathbf{0}$, and applying this recursively, we see $\mathbf{CA}^k\mathbf{x}_0 = \mathbf{0}$ for all k . Hence, \mathbf{x}_0 is an unobservable state, so the system is unobservable. \square

Example 7.7 (Sprung beam). Now that we have a test for observability, let us return to the sprung beam we considered in Example 7.4. We will assume $\delta = 0$ (for $\delta \neq 0$, see Exercise 7.1), and compute the observability matrix

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \end{bmatrix} = \begin{bmatrix} 1 & r & 0 & 0 \\ 0 & 0 & 1 & r \\ -1 & -r/J & 0 & 0 \\ 0 & 0 & -1 & -r/J \end{bmatrix}.$$

When $r = 0$, the second and fourth columns are zero, so in this case, the matrix is not full rank, and the system is not observable (as we expect based on the intuitive explanation in Example 7.4). The corresponding unobservable states \mathbf{x}_0 satisfy

$$\mathbf{\Theta}\mathbf{x}_0 = \mathbf{0} \implies \mathbf{x}_0 = \begin{bmatrix} 0 \\ \theta_0 \\ 0 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dot{\theta}_0 \end{bmatrix}.$$

This result agrees with our intuitive explanation from before: if the beam has an initial angle or angular velocity, but no vertical displacement or vertical velocity, then the output y will be zero for all time.

As in Example 6.5, there is another way in which \mathbf{O} can lose rank: when $J = 1$, the second column is r times the first column, and the fourth column is r times the third. In this case, the unobservable states \mathbf{x}_0 satisfy

$$\mathbf{O}\mathbf{x}_0 = \mathbf{0} \implies \mathbf{x}_0 = \begin{bmatrix} -r\theta_0 \\ \theta_0 \\ 0 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 \\ 0 \\ -r\dot{\theta}_0 \\ \dot{\theta}_0 \end{bmatrix}.$$

This result also makes intuitive sense: we saw in Example 6.5 that when $J = 1$, the frequency of “bouncing” in the z -direction equals the frequency of “rocking” in the θ -direction. The unobservable states above represent initial conditions for which $y = 0$, and since the frequencies of bouncing and rocking are the same, the bouncing and rocking remain in phase, and y remains zero for all time.

These are the only cases when \mathbf{O} is not full rank, as one can check by calculating when $\det \mathbf{O} = 0$:

$$\det \mathbf{O} = -\frac{r^2}{J^2}(J-1)^2,$$

which is zero only when $r = 0$ or $J = 1$. ◇

Duality

You have probably noticed that many of the concepts that arose in our discussion of observability are similar to concepts we used to discuss controllability: for instance, the matrix \mathbf{O} in (7.4) resembles the controllability matrix \mathbf{C} from (6.2) in some ways. There is actually a precise “duality” between observability and controllability, which turns out to be useful in a number of situations.

We have been considering the system (7.1). Now, consider another linear system, called the *dual system* of (7.1), given by

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{A}^\top \mathbf{z} + \mathbf{C}^\top \mathbf{u} \\ \mathbf{y} &= \mathbf{B}^\top \mathbf{z} + \mathbf{D}^\top \mathbf{u}. \end{aligned} \tag{7.7}$$

Let us look at the controllability matrix for this dual system. We have

$$\mathbf{c} = [\mathbf{C}^\top \quad \mathbf{A}^\top \mathbf{C}^\top \quad \dots \quad (\mathbf{A}^\top)^{n-1} \mathbf{C}^\top] = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}^\top. \tag{7.8}$$

So, comparing with (7.4), we see that the controllability matrix for the dual system (7.7) is simply the transpose of the observability matrix for the original system (7.1). Conversely, the observability matrix for the dual system is the transpose of the controllability matrix for the original system. Since the rank of a matrix is the same as the rank of its transpose (or, in other words, the row and column ranks are equal), this implies the following:

- The system (7.1) is observable \iff the dual system (7.7) is controllable.
- The system (7.1) is controllable \iff the dual system is observable.

Notions of duality such as this pervade many areas of mathematics, and they can be extremely useful. In our case, the duality above will let us apply ideas we have developed for design of state feedback laws (in particular, pole placement) to the design of observers, to be discussed in the next section.

7.2 Observers

An *observer* is a device that reconstructs an estimate of the state \mathbf{x} , given only the inputs \mathbf{u} and the outputs \mathbf{y} . An abstract notion of an observer is shown in the following diagram:



Here, $\hat{\mathbf{x}}$ represents an estimate of the state, and we shall try to design an observer so that $\hat{\mathbf{x}}$ is a good approximation to \mathbf{x} . This is not an easy thing to do. A good example to keep in mind is the sprung beam, from Examples 7.4 and 7.7: this has four states, $(z, \theta, \dot{z}, \dot{\theta})$, and an observer must reconstruct all four from the single output $y = z + r\theta$. We know this is possible in principle, since the system is observable (except in certain special cases—see Example 7.7), but what is a practical way to estimate the state?

Perhaps the most obvious approach is the following: we know that, as long as the system (7.1) is observable, we can (at least in principle) construct the initial state \mathbf{x}_0 from the output $\mathbf{y}(t)$. So let us determine this initial state \mathbf{x}_0 , and then compute our state estimate $\hat{\mathbf{x}}(t)$ by running a simulation

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u}, \quad \hat{\mathbf{x}}(0) = \mathbf{x}_0. \quad (7.9)$$

This idea seems reasonable, but it has a few serious problems:

- The model we use in our simulation might differ slightly from the actual system. (In practice, no model is ever perfect.) The resulting errors between the true state and our state estimate will accumulate over time.

- Real systems usually have external disturbances, which we do not measure. These will drive the true state away from our state estimate, because our simulation does not include the effects of the disturbances.

The underlying source of these problems is that, once we determine the initial state \mathbf{x}_0 , we no longer use our sensor measurements $\mathbf{y}(t)$. It would be better to devise a way of incorporating information from our sensor measurements as time goes on.

Let us now consider a modification to this approach that does incorporate information from the output \mathbf{y} . In particular, let our state estimate $\hat{\mathbf{x}}$ satisfy

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}) \quad (7.10)$$

where \mathbf{L} is a matrix of “observer gains” that we will select, $\mathbf{y} = \mathbf{C}\mathbf{x}$ is the (actual) output, and $\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}}$ is the predicted output: that is, it is the output we would have measured, if the actual state \mathbf{x} were equal to our estimate $\hat{\mathbf{x}}$. The idea is that, if our state estimate is correct and $\hat{\mathbf{x}} = \mathbf{x}$, then (7.10) agrees with (7.9). However, if our state estimate differs from the true state, then the last term of (7.10) provides a correction that should push $\hat{\mathbf{x}}$ closer to the true state \mathbf{x} . In this way, we should be able to overcome both of the problems with the previous approach (7.9): the sensor corrections can correct for errors in our state estimate, whether they arise from errors in our model, or from external disturbances. Note, however, that we must choose \mathbf{L} appropriately—with the wrong choice of \mathbf{L} , the additional terms could also drive us *further* from the actual state.

Before discussing how to choose \mathbf{L} , let us show how to write the dynamics (7.10) as a state-space system with input (\mathbf{u}, \mathbf{y}) and output $\hat{\mathbf{x}}$, as in the diagram at the beginning of this section. We can write (7.10) as

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \\ &= (\mathbf{A} - \mathbf{LC})\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}\mathbf{y} \\ &= (\mathbf{A} - \mathbf{LC})\hat{\mathbf{x}} + \begin{bmatrix} \mathbf{B} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix}. \end{aligned} \quad (7.11)$$

A state-space realization for our observer is therefore

$$\left[\begin{array}{c|cc} \mathbf{A} - \mathbf{LC} & \mathbf{B} & \mathbf{L} \\ \hline \mathbf{I} & \mathbf{0} & \mathbf{0} \end{array} \right].$$

Now let us return to the question of how to choose \mathbf{L} . To see this, let us look at the quantity $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, which is the error between the actual state \mathbf{x}

and the state estimate $\hat{\mathbf{x}}$. Using (7.11), we find

$$\begin{aligned}\dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} \\ &= \mathbf{Ax} + \mathbf{Bu} - ((\mathbf{A} - \mathbf{LC})\hat{\mathbf{x}} + \mathbf{Bu} + \mathbf{Ly}) \\ &= \mathbf{Ax} - \mathbf{LCx} - (\mathbf{A} - \mathbf{LC})\hat{\mathbf{x}} \\ &= (\mathbf{A} - \mathbf{LC})(\mathbf{x} - \hat{\mathbf{x}}) \\ &= (\mathbf{A} - \mathbf{LC})\mathbf{e}.\end{aligned}$$

By now, we are very familiar with this type of equation: its solution is

$$\mathbf{e}(t) = e^{(\mathbf{A} - \mathbf{LC})t} \mathbf{e}(0).$$

We would like this error $\mathbf{e}(t)$ to go to zero as time goes on. In order to achieve this, then, *we should choose \mathbf{L} such that the eigenvalues of $\mathbf{A} - \mathbf{LC}$ are in the left half-plane*: then the error \mathbf{e} will converge to zero as $t \rightarrow \infty$. The further in the left half-plane the poles of $\mathbf{A} - \mathbf{LC}$ are, the faster the error will converge to zero.

Pole placement

We now know what criteria we would like \mathbf{L} to satisfy (namely, that the eigenvalues of $\mathbf{A} - \mathbf{LC}$ be in the left half-plane), but this still does not tell us precisely how to choose \mathbf{L} . Fortunately, we can borrow some methods from §6.2 for designing state feedback gains, using the duality principle explained in § 7.1. It turns out that, as long as the pair (\mathbf{A}, \mathbf{C}) is observable, we can place poles of $\mathbf{A} - \mathbf{LC}$ *anywhere we like*, using the same pole placement technique we used for state feedback in §6.2.

Suppose the system

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} \\ \mathbf{y} &= \mathbf{Cx}\end{aligned}\tag{7.12}$$

is observable. Then the dual system

$$\dot{\mathbf{z}} = \mathbf{A}^T \mathbf{z} + \mathbf{C}^T \mathbf{u}\tag{7.13}$$

is controllable (see §7.1). So we can place poles of $\mathbf{A}^T - \mathbf{C}^T \mathbf{K}$ anywhere we like, by Theorem 6.9. But the eigenvalues of $\mathbf{A}^T - \mathbf{C}^T \mathbf{K}$ are the same as the eigenvalues of $(\mathbf{A}^T - \mathbf{C}^T \mathbf{K})^T = \mathbf{A} - \mathbf{K}^T \mathbf{C}$. Thus, if we choose $\mathbf{L} = \mathbf{K}^T$, the eigenvalues of $\mathbf{A} - \mathbf{LC}$ are the same as those of $\mathbf{A}^T - \mathbf{C}^T \mathbf{K}$, so we can place the eigenvalues of $\mathbf{A} - \mathbf{LC}$ anywhere we like.

In MATLAB, then, one can design observer gains \mathbf{L} using the same **place** command we used for designing state feedback gains \mathbf{K} , as follows:

```
K = place(A', C', p);  
L = K';
```

As with the design of state feedback controllers, there is still the question of where to choose the poles of $\mathbf{A} - \mathbf{L}\mathbf{C}$. Many of the same guidelines we learned in § 6.4 about choosing closed-loop poles for state feedback also apply to observers: for instance, if the plant has fast poles or slow zeros, we should place poles of $\mathbf{A} - \mathbf{L}\mathbf{C}$ at the same locations. In general, the further in the left half-plane we place the poles, the faster the error will converge to zero. However, as with the state feedback case, there is a tradeoff: if we place the poles too far in the left half-plane, the resulting gains \mathbf{L} will be large, and our observer will amplify sensor noise. A well-designed observer will achieve a compromise between fast tracking (poles far in the left half-plane) and small gains \mathbf{L} .

Example 7.8 (Observer for sprung beam). Recall that the sprung beam considered in Example 7.7 is observable as long as $r \neq 0$ and $J \neq 1$. Here, we design an observer to reconstruct the full state $\mathbf{x} = (z, \theta, \dot{z}, \dot{\theta})$ from a single sensor measurement $y = z + r\theta$. In particular, we consider the beam starting from rest, with initial state $(0, 0, 0, 0)$, excited by an (unknown) disturbance, a vertical force applied at a distance r from the center of mass, as in Example 6.5. The equations of motion are thus $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$, and $y = \mathbf{C}\mathbf{x}$, with

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -\delta & 0 \\ 0 & -1/J & 0 & -\delta/J \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ r/J \end{bmatrix}, \quad \mathbf{C} = [1 \quad r \quad 0 \quad 0],$$

where we now regard u as an unknown disturbance (rather than a known control input). We consider parameter values $r = 0.5$, $J = 0.5$, $\delta = 0.1$, for which the single-input single-output system has poles at $-0.05 \pm 1.0i$, $-0.1 \pm 1.4i$, and a pair of complex zeros at $-0.08 \pm 1.3i$.

We design the observer gains using pole placement, specifying the eigenvalues of $\mathbf{A} - \mathbf{L}\mathbf{C}$ as described above. In particular, we compare the performance of the observer for the following choices of pole locations:

$$\begin{aligned} -1 \pm i, \quad -2 \pm 2i &\implies \mathbf{L}_1 = [16.5 \quad -21.6 \quad -0.41 \quad 27.4]^\top \\ -5 \pm 5i, \quad -10 \pm 10i &\implies \mathbf{L}_2 = [1970 \quad -3881 \quad 9554 \quad -18232]^\top \\ -2 \pm 2i, \quad -0.08 \pm 1.3i &\implies \mathbf{L}_3 = [2.6 \quad 2.5 \quad 4.4 \quad 3.5]^\top. \end{aligned}$$

Observe that the gains \mathbf{L}_2 correspond to poles that are “faster” than those of \mathbf{L}_1 (i.e., eigenvalues of $\mathbf{A} - \mathbf{L}_2\mathbf{C}$ are further in the left half-plane than those of $\mathbf{A} - \mathbf{L}_1\mathbf{C}$), so we might expect the error to converge faster, and thus the observer to perform better. However, also observe that the gains \mathbf{L}_2 are much larger than those of \mathbf{L}_1 . The third set of gains was chosen so that a pair of poles lies at the locations of the zeros of the plant, and note that the gains \mathbf{L}_3 are much smaller than those of \mathbf{L}_1 or \mathbf{L}_2 .

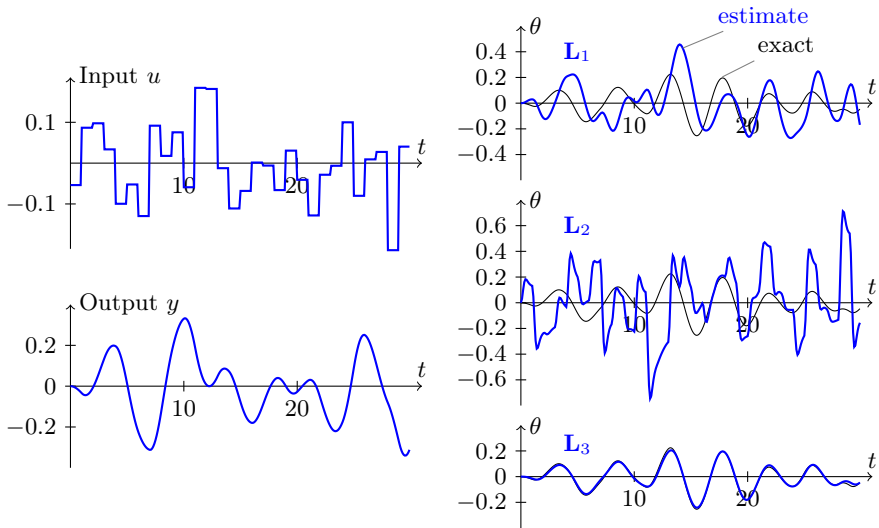


Figure 7.1: Comparison of three different observer designs in Example 7.8, showing the (random) input and corresponding plant output (left), and the observer estimates of θ compared with the true θ (right).

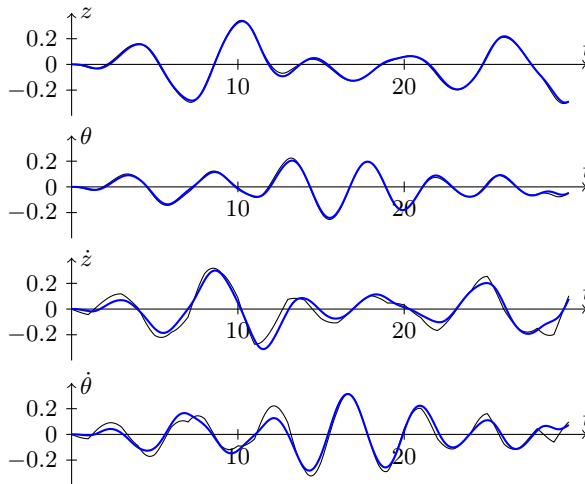


Figure 7.2: Comparison of state estimates with the true states, for Example 7.8, observer L_3 , for the same random disturbance shown in Figure 7.1.

The performance of the three different observer designs is shown in Figure 7.1. The plots at the left of the figure show the input disturbance u , and the corresponding output y . Here, the input is chosen to be piecewise constant, with values at each time interval chosen from a zero-mean normal distribution with standard deviation 0.1. The observer tries to reconstruct all four states $(z, \theta, \dot{z}, \dot{\theta})$ using only this output signal y . We emphasize that, because this input u is being regarded as an unknown disturbance, the observer has no knowledge of it (that is, u is taken to be zero in equation (7.11)).

The corresponding state estimates of θ are shown in the plots at the right of Figure 7.1. The observer using gains \mathbf{L}_1 gets the general trend correct, but the state estimate seems to lag behind the true value, the amplitude is also off, and the overall accuracy of the estimate is poor. One might think that \mathbf{L}_2 should perform better, since the observer poles are further in the left half-plane, but actually the estimate is far worse! However, the choice \mathbf{L}_3 performs extremely well, as shown in the plot at the bottom right.

The state estimates for all of the states $(z, \theta, \dot{z}, \dot{\theta})$ are shown in Figure 7.2, for the gains \mathbf{L}_3 . Note that all the states are estimated with excellent accuracy. It is remarkable that the estimates of all four of these signals were reconstructed from the single signal y shown in Figure 7.1.

This example illustrates that it is critical to choose suitable pole locations when designing observers using pole placement: merely placing the poles further in the left half-plane does not always yield a better design. The criteria discussed in Section 6.4 for choosing pole locations for state-feedback controllers also apply to observer design. For instance, in this example, there is a pair of “slow zeros,” as in Example 6.14, and placing a pair of poles at the location of these zeros dramatically improves the performance of the observer. \diamond

7.3 Observer-based feedback

The primary reason we are interested in constructing an estimate of the state with an observer is to use the state estimate for feedback, for instance using the tools developed in Chapter 6. Consider again the plant described by (7.1), and assume that we can measure the full state \mathbf{x} . Figure 7.3 shows the block diagram associated with the feedback law $\mathbf{u} = -\mathbf{K}\mathbf{x}$. This configuration is called *full-state feedback*, because the full state \mathbf{x} is measured and used in the feedback law.

In most situations, however, we do not have access to the full state; instead, we have access only to the output \mathbf{y} . However, we may use an observer to reconstruct a state estimate $\hat{\mathbf{x}}$, and use it for a feedback law $\mathbf{u} = -\mathbf{K}\hat{\mathbf{x}}$. The corresponding block diagram is shown in Figure 7.4. This configuration is called *output feedback* (or *observer-based feedback*), because only the output \mathbf{y} is used for feedback (instead of the full state \mathbf{x}).

The orange box in Figure 7.4 is called an *observer-based compensator*:

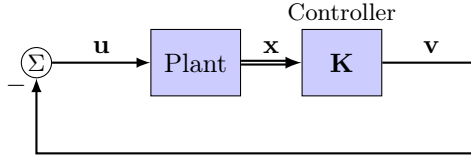


Figure 7.3: Block diagram of a controller with full state feedback: we assume we can measure the full state \mathbf{x} .

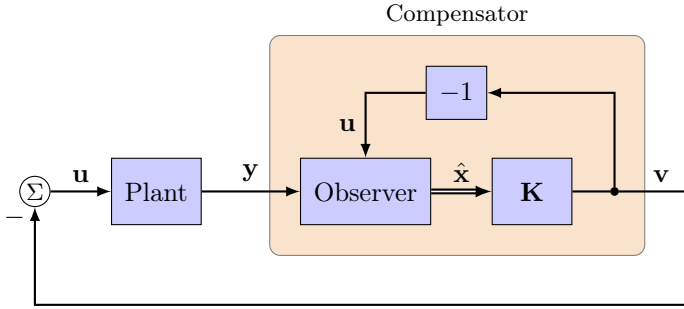


Figure 7.4: Block diagram of a controller with output feedback: we measure the output \mathbf{y} , and reconstruct a state estimate using an observer.

its input is \mathbf{y} , its output is $\mathbf{v} = -\mathbf{u}$, and it encloses both the observer and the state feedback controller. Note that, if u and y are both scalars (i.e., if the plant is a SISO system), then the compensator is also a SISO system, and can therefore be represented as a (scalar) transfer function, just as the compensators we considered in Chapter 4.

Let us determine a state-space realization for the compensator. We begin with the dynamics of the observer, which are given by

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{LC})\hat{\mathbf{x}} + \mathbf{Bu} + \mathbf{Ly}.$$

Substituting the feedback law $\mathbf{u} = -\mathbf{K}\hat{\mathbf{x}}$, this becomes

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{BK} - \mathbf{LC})\hat{\mathbf{x}} + \mathbf{Ly}. \quad (7.14)$$

The output of the compensator in Figure 7.4 is $\mathbf{v} = \mathbf{K}\hat{\mathbf{x}}$, so combining this with (7.14) gives the state-space realization

$$\left[\begin{array}{c|c} \mathbf{A} - \mathbf{BK} - \mathbf{LC} & \mathbf{L} \\ \hline \mathbf{K} & 0 \end{array} \right]. \quad (7.15)$$

In the single-input single-output case, we can treat this observer-based compensator just like any other classical controller, for instance plotting Bode or Nyquist plots, and using it for reference tracking or disturbance rejection.

Separation principle

Now, let us take a closer look at the stability of the closed-loop system shown in Figure 7.4. In particular, suppose the poles of $\mathbf{A} - \mathbf{BK}$ are in the left half-plane, so the closed-loop system for full-state feedback shown in Figure 7.3 is stable. Suppose further that the poles of $\mathbf{A} - \mathbf{LC}$ are in the left half-plane, so the observer is stable and the state estimate $\hat{\mathbf{x}}$ converges to the true state \mathbf{x} . Under what conditions is the overall feedback loop in Figure 7.4 stable? This is not so clear. One might expect that, as long as the state estimate converges fast enough, the overall feedback loop should be stable. But what if the observer converges slowly, while the state feedback drives the state to zero rapidly? One would imagine this might cause problems, because the state feedback controller would be working with an inaccurate estimate of the state.

The (perhaps surprising) result is that the closed-loop system in Figure 7.4 is *always* stable, as long as poles of $\mathbf{A} - \mathbf{BK}$ and of $\mathbf{A} - \mathbf{LC}$ are in the left half-plane. The precise result is stated in the following theorem.

Theorem 7.9 (Separation principle). *The poles of the closed-loop system shown in Figure 7.4 are the poles of the full-state feedback system (eigenvalues of $\mathbf{A} - \mathbf{BK}$) together with the poles of the observer (eigenvalues of $\mathbf{A} - \mathbf{LC}$).*

Proof. The key here is to write the governing equations in coordinates in which the result is obvious. In particular, we use the state variables (\mathbf{x}, \mathbf{e}) , where $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$. Using the feedback law $\mathbf{u} = -\mathbf{K}\hat{\mathbf{x}}$ from Figure 7.4, the state \mathbf{x} evolves as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ &= \mathbf{Ax} - \mathbf{BK}\hat{\mathbf{x}} \\ &= \mathbf{Ax} - \mathbf{BK}(\mathbf{x} + \mathbf{e}) \\ &= (\mathbf{A} - \mathbf{BK})\mathbf{x} - \mathbf{BKe}.\end{aligned}$$

Now, using this result together with (7.14), we find

$$\begin{aligned}\dot{\mathbf{e}} &= \dot{\hat{\mathbf{x}}} - \dot{\mathbf{x}} \\ &= (\mathbf{A} - \mathbf{BK} - \mathbf{LC})\hat{\mathbf{x}} + \mathbf{Ly} - [(\mathbf{A} - \mathbf{BK})\mathbf{x} - \mathbf{BKe}] \\ &= (\mathbf{A} - \mathbf{BK})(\hat{\mathbf{x}} - \mathbf{x}) - \mathbf{LC}\hat{\mathbf{x}} + \mathbf{LCx} + \mathbf{BKe} \\ &= (\mathbf{A} - \mathbf{BK} - \mathbf{LC})\mathbf{e} + \mathbf{BKe} \\ &= (\mathbf{A} - \mathbf{LC})\mathbf{e}.\end{aligned}$$

These are two coupled linear systems. Writing them together, we have

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK} & -\mathbf{BK} \\ \mathbf{0} & \mathbf{A} - \mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}.$$

Note that the matrix in the above equation is block upper triangular. Its eigenvalues are therefore the eigenvalues of the matrices on the diagonals,

namely those of $\mathbf{A} - \mathbf{BK}$ together with the eigenvalues of $\mathbf{A} - \mathbf{LC}$, which was to be shown. \square

This result is known as the *separation principle* because the state feedback controller and the observer can be designed separately, and as long as each is stable, the resulting closed-loop system is guaranteed to be stable. This is a remarkable result, and it means that we now have a systematic procedure by which we can stabilize *any* linear system, as we summarize below.

Systematic way to stabilize any plant

Suppose we are given a (proper) transfer function $P(s)$, which may be unstable, and our goal is to design a feedback controller such that the closed-loop system is stable. Consider the following procedure:

1. Obtain a state-space realization for $P(s)$ (§5.3):

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} + Du.\end{aligned}$$

2. Design a state-feedback controller $u = -\mathbf{K}\mathbf{x}$, for instance using pole placement (§6.2) or LQR (§6.5).
3. Design an observer, for instance using pole placement (§7.2):

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{LC})\hat{\mathbf{x}} + \mathbf{B}u + \mathbf{L}y,$$

4. Connect them, as in Figure 7.4. The resulting compensator is given by (7.15).

Because of the separation principle, the poles of the closed-loop system are precisely the eigenvalues of $\mathbf{A} - \mathbf{BK}$ (chosen in step 2 to be stable) together with the eigenvalues of $\mathbf{A} - \mathbf{LC}$ (chosen in step 3 to be stable), so the closed-loop system is guaranteed to be stable. Note that, as long as our state-space realization is controllable and observable, we can assign these poles to be wherever we like. Note that this procedure works even for multiple-input, multiple-output (MIMO) plants, for which \mathbf{u} and \mathbf{y} are vectors and $P(s)$ is a matrix of transfer functions.

In the above procedure, the order of the compensator in (7.15) is the same as the order of the plant model. Thus, if we have a very complex model of the plant $P(s)$, we will also get a very complex controller $C(s)$. It might not be desirable to have such a complex controller, because a complex controller will involve more calculations in order to compute the input \mathbf{u} at each time. However, often one can get a simpler, almost identical controller by *model reduction*, in which one tries to find a lower-order system that has approximately the same input-output behavior. There are various systematic

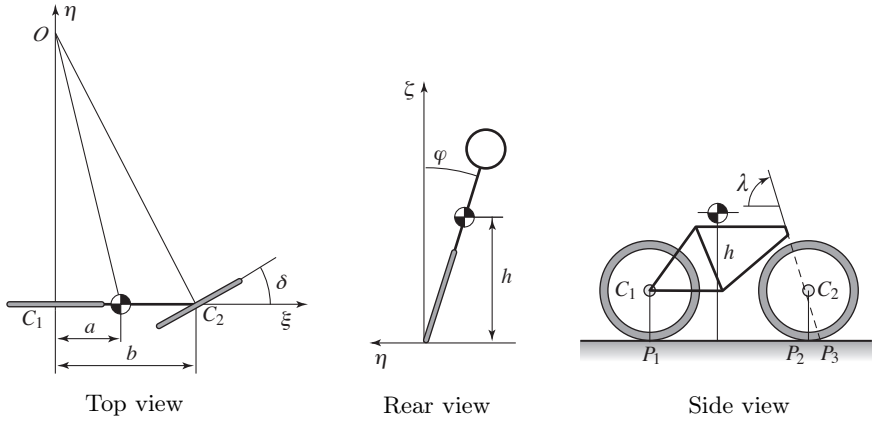


Figure 7.5: Parameters for the bicycle model used in Example 7.10 (from [1, p. 70]). The steering angle is δ and the tilt angle is φ .

procedures for model reduction of linear systems, such as balanced truncation and Hankel norm reduction: to learn about these, see, for instance, Skogestad and Postlethwaite [22, Ch. 11].

Though the above procedure is systematic, and is always guaranteed to produce a stabilizing controller, the choices of state feedback gains \mathbf{K} and observer gains \mathbf{L} are still important. Some choices are much better than others, for instance in terms of performance (e.g., tracking or disturbance rejection) or in terms of robustness (e.g., how well the controller works if our model of the plant is not perfect). How can one evaluate the performance or robustness of one controller over another? For SISO plants, we can use the same tools we developed in preceding chapters, for instance looking at Bode plots of the sensitivity function and complementary sensitivity function (§2.6 and §6.4), or Bode and Nyquist plots of the loop transfer function (§4.4). The next example illustrates how the choice of observer gains \mathbf{L} can have a profound effect on the performance and robustness of a feedback controller.

Example 7.10 (Bicycle stabilization). The following is simple model of the dynamics of a bicycle, where δ is the steering angle, φ is the tilt angle, and the other parameters are as shown in Figure 7.5:

$$J \frac{d^2 \varphi}{dt^2} - mgh\varphi = \frac{Dv_0}{b} \frac{d\delta}{dt} + \frac{mv_0^2 h}{b} \delta.$$

Here, v_0 is the forward velocity, $J \approx mh^2$ and $D \approx mah$ are moments of inertia, and the model is linearized about the upright position ($\varphi = 0$). (See Åström and Murray [1, §3.2] for a derivation of this model.) Taking Laplace transforms, we find the transfer function from δ to φ to be

$$P(s) = \frac{av_0}{bh} \frac{s + v_0/a}{s^2 - g/h}.$$

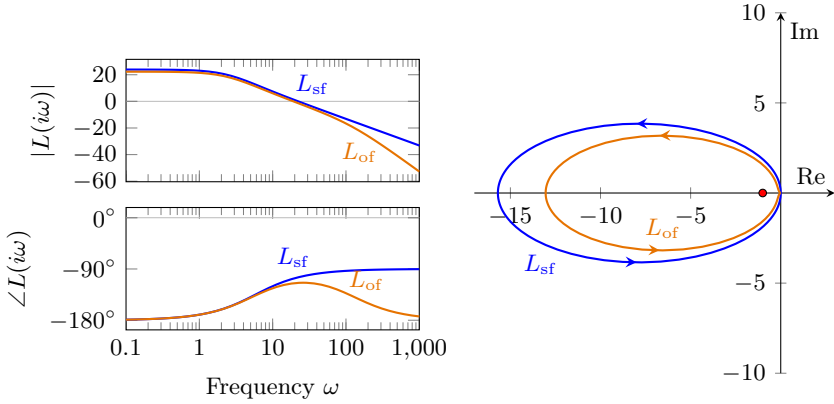


Figure 7.6: Bode and Nyquist plots of loop gain $L(s)$ for bicycle example, comparing full-state feedback (L_{sf}) and output feedback (L_{of}), using the “robust” observer from Table 7.1.

Note that the transfer function has poles at $s = \pm\sqrt{g/h}$, just like an inverted simple pendulum with length h . We will design an observer-based compensator to stabilize the bicycle, given measurements of the tilt angle φ .

Using the controllable canonical form (5.15), we obtain a state-space realization

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ g/h & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \delta \\ \varphi &= \frac{av_0}{bh} \begin{bmatrix} v_0/a & 1 \end{bmatrix} \mathbf{x}.\end{aligned}$$

Let us choose parameter values $m = 75$, $a = 0.5$, $b = 1.5$, $h = 1.2$, and $v_0 = 2$, and design a state-feedback controller: using pole placement, we choose \mathbf{K} such that the closed-loop poles of $\mathbf{A} - \mathbf{BK}$ are at $-10, -12$, obtaining

$$\mathbf{K} = \begin{bmatrix} 128 & 22 \end{bmatrix}. \quad (7.16)$$

At this point, we may examine the performance and robustness of the state-feedback controller $u = -\mathbf{K}\mathbf{x}$ by looking at Bode and Nyquist plots of the loop gain (6.13), as shown in Figure 7.6. From the Bode plot, we see that we have large gain at low frequencies (good for disturbance rejection), and from the Bode and Nyquist plots we see that we have good stability margins: the phase margin is $\varphi_m = 75^\circ$, and the stability margin from the Nyquist plot $s_m = 1$. This controller should therefore provide good performance, and should be robust: that is, it should work even if our model of the plant does not perfectly match the actual system.

In practice, however, we cannot measure the full state \mathbf{x} : we measure only φ . Therefore, we will design an observer to reconstruct \mathbf{x} from the

Description	Poles of $\mathbf{A} - \mathbf{LC}$	\mathbf{L}	φ_m	s_m
Robust	$-100, -4$	$\begin{bmatrix} 180 & 7.2 \end{bmatrix}^\top$	65°	0.88
Slow	$-1, -2$	$\begin{bmatrix} 3.7 & 1.7 \end{bmatrix}^\top$	29°	0.24
Fast	$-100, -120$	$\begin{bmatrix} 10,600 & -10,200 \end{bmatrix}^\top$	19°	0.21

Table 7.1: Observer pole locations and gain matrices for Example 7.10. Also shown is the stability margin s_m for the resulting compensator when used with state feedback gains \mathbf{K} from (7.16).

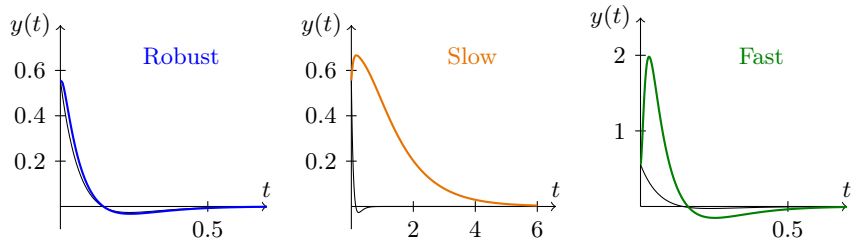


Figure 7.7: Impulse responses for Example 7.10, comparing state feedback (thin) with output feedback, for the three different choices of observer gains from Table 7.1.

sensor measurement φ . We choose the observer gains \mathbf{L} by pole placement, and several choices are listed in Table 7.1. For instance, for the design labeled “Robust,” the gains \mathbf{L} were chosen to place the closed-loop poles at -100 and -4 . The compensator $C(s)$ is given by (7.15), and we can again plot Bode and Nyquist plots of the loop gain $L(s) = P(s)C(s)$, and compare these with the state feedback case. (Note that the loop gain $L(s)$ is a transfer function, not to be confused with the observer gains \mathbf{L} in Table 7.1!) These plots are shown in Figure 7.6, for the “Robust” design. Note that the loop gain for observer-based feedback closely matches that for full-state feedback, until after the crossover frequency, at which point the output-feedback case rolls off more steeply. The Nyquist plot also indicates good stability margins for the output-feedback case, so we would expect this compensator to have good robustness.

We can also compare the impulse responses of the closed-loop systems, as shown in Figure 7.7. As we see in the leftmost plot, the impulse response for the system with the “Robust” observer closely matches that of full-state feedback. It appears this is a good control design.

However, not all observers perform this well. Let us compare the other designs shown in Table 7.1. For the “Slow” design, the poles of $\mathbf{A} - \mathbf{LC}$ are chosen at $-1, -2$. Compare these values to the controller poles (eigenvalues of $\mathbf{A} - \mathbf{BK}$), which we chose at $-10, -12$. The observer poles are slower than

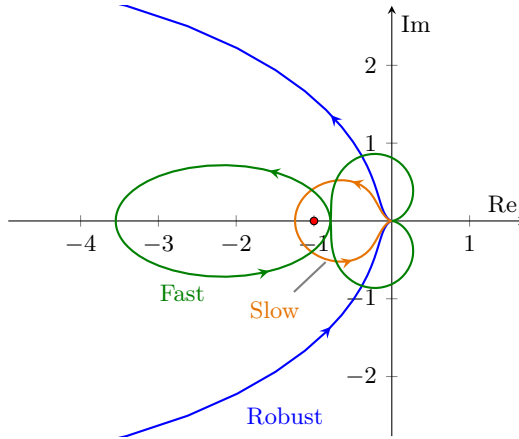


Figure 7.8: Nyquist plots for output feedback controllers in Table 7.1.

the controller poles (not as far in the left half-plane), so we might therefore expect the resulting observer to perform poorly, since the state estimate will converge slowly in comparison with the desired closed-loop system response. Figure 7.7 (middle) shows the impulse response for this system, and we see that indeed, it converges much more slowly than the full state-feedback case and the “Robust” observer.


In addition, Figure 7.8 compares the Nyquist plots of the loop gains for the different observer designs. The “Slow” design comes very close to the -1 point, indicating poor robustness. We would therefore expect the “Slow” design to perform poorly in practice, and probably not even stabilize the actual system (since our plant model is almost certainly not perfectly accurate).


Now, recall that our “Robust” observer has closed-loop poles at -100 and -4 . This pole at -4 might seem curious: since our state feedback poles (eigenvalues of $\mathbf{A} - \mathbf{BK}$) are at $-10, -12$, perhaps we can get better performance if we choose a faster pole, for instance at $-100, -120$, as in the “Fast” design in Table 7.1. This turns out, surprisingly, to be a terrible idea. First, we notice that the observer gains shown in Table 7.1 are much larger than for the other observers, indicating that this design will have much greater amplification of sensor noise. But the real problem is with robustness: we see from Table 7.1 and from the Nyquist plot in Figure 7.8 that the resulting stability margins are very small, even smaller than for the “Slow” observer. In addition, the impulse response shown in Figure 7.7 looks terrible: while the convergence is indeed fast, there is a large transient, initially in the wrong direction, that looks much worse than the full-state feedback case or the “Robust” observer design.

How can this be? How can an observer with poles at $-100, -120$ be *less* robust than one with poles at $-100, -4$? The poles of the closed-loop system

are further in the left half-plane, so one would think it should only be *more* robust (further from instability). What is happening here?

The key is to notice that the plant has a *zero* at $s = -v_0/a = -4$. Thus, we should place a closed-loop pole at $s = -4$ in order to cancel this zero, as we saw previously in Example 6.14. If all of the closed-loop poles are “faster” than this zero, we will always have poor robustness, as we now explain.

 Since the plant $P(s)$ has a zero at $s = -4$, the loop gain $L(s) = P(s)C(s)$ also has a zero at $s = -4$, and so does the complementary sensitivity function $T = L/(1 + L)$. Consider the Bode magnitude plot of T : for low frequencies, the slope is zero, with magnitude about 1, and then the plot “breaks up” (increases slope) at $\omega = 4$ rad/sec, because of the zero at $s = -4$. The magnitude plot does not “break down” again until we reach the break points corresponding to our closed-loop poles. Of course, we get to *choose* the closed-loop poles, and for the “Fast” observer design, we have chosen them to be at $-10, -12, -100, -120$. So the Bode magnitude plot of T breaks up at $\omega = 4$ rad/sec and does not break down again until $\omega = 10$ rad/sec, and will therefore have a peak. Since $S + T = 1$, if $|T|$ has a large peak, then $|S|$ will also have a large peak, implying poor robustness: recall from §4.7 that $1/|S(i\omega)|$ is the distance from the -1 point to the point $L(i\omega)$ on the Nyquist plot of L , so a peak in $|S|$ indicates that the Nyquist plot comes close to the -1 point, indicating poor robustness. From this reasoning, we see that, surprisingly, the further in the left half-plane we place the closed-loop poles, the larger the peak in $|S|$ will be, and thus the worse the robustness will be.

 The solution is to place a closed-loop pole at $s = -4$: this pole will cancel with the closed-loop zero at $s = -4$, and thus $|T|$ and $|S|$ will no longer have large peaks, and we will avoid the robustness problems described above. This is the reasoning behind selecting an observer pole at -4 for the “Robust” observer design. \diamond

The previous example illustrates that observer design is not always simple. We can use many of the same criteria for pole placement as described in §6.4, but poor choices of closed-loop poles can lead to bad performance. Furthermore, it is often *not* a good idea to place observer poles far in the left half-plane, as this can lead to large amplification of sensor noise, as well as extremely poor robustness (e.g., stability margins).

7.4 Fundamental limitations

As described in the previous section, the tools we have developed can be used to find an observer-based compensator that stabilizes *any* linear time-invariant system (as long as we have a state-space realization that is both controllable and observable). However, this statement is a bit misleading, for it is not always possible to stabilize a given system *robustly*, i.e., with reasonable stability margins. Some systems are inherently difficult to control. In particular, systems can be difficult to control if they have any of the following characteristics:

- Unstable dynamics (RHP poles)
- RHP zeros
- Time delays

In this section, we will quantify what it means for these systems to be “difficult to control,” in particular by looking at some fundamental limitations these systems have. By *fundamental* limitations, we mean limitations of *any* controller one could possibly design, not just of one particular control design. For instance, we may have constraints on the bandwidth or crossover frequency, as summarized below, for a system with a real RHP pole at $s = p$, a real RHP zero at $s = z$, or a time delay τ :

In order to have a reasonable (30°) phase margin, we have the following constraints on the bandwidth or crossover frequency:

- RHP pole p : *minimum* bandwidth $\omega_c > 1.7p$.
- RHP zero z : *maximum* bandwidth $\omega_c < 0.6z$.
- Time delay τ : *maximum* bandwidth $\omega_c < 1.1/\tau$.

Note that, if a system has a right-half-plane zero z near a right-half-plane pole p (that is, $z \approx p$), then these constraints are in conflict, and it is *impossible* to stabilize the system *robustly* (i.e., with reasonable stability margins). We will see a more quantitative expression of these limitations later in this section (equation (7.20)). We will soon investigate where the above constraints come from, but first let us show why we cannot simply cancel right-half-plane poles and zeros with a suitable controller.

Right-half-plane pole/zero cancellation is bad

Suppose we have an unstable plant, such as

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

that we wish to stabilize with feedback control. Of course, this plant is easy to stabilize: a simple proportional controller $C(s) = 2$ does the job. However, it may be tempting to simply *cancel* the RHP pole at $s = 1$ with a RHP zero in our controller. For instance, if we choose

$$C(s) = \frac{s-1}{s+1},$$

then the loop gain is

$$L(s) = P(s)C(s) = \frac{s-1}{(s-1)(s+1)} = \frac{1}{s+1},$$

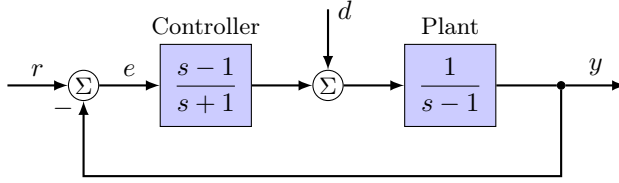


Figure 7.9: Block diagram showing cancellation of a right-half-plane pole.

which has no RHP pole. If we consider the block diagram shown in Figure 7.9, the closed-loop transfer function from r to y is

$$T = \frac{PC}{1 + PC} = \frac{1}{s + 2},$$

which is perfectly well behaved. Furthermore, the transfer function from r to e is

$$S = \frac{1}{1 + PC} = \frac{s + 1}{s + 2},$$

which is also fine. So far, it appears there is no problem at all.

However, suppose we have a disturbance d , as indicated in the diagram. Then the transfer function from d to y is

$$\frac{P}{1 + PC} = \frac{s + 1}{(s - 1)(s + 2)}.$$

This transfer function contains the original RHP pole at $s = 1$, so disturbances will still blow up. A similar problem happens if we try to cancel a RHP plant zero by including a RHP pole in the controller: the transfer function from r to y will appear fine, but in this case, the transfer function from sensor noise (entering at the plant output y) to u will have a RHP pole.

So even if we could cancel RHP poles and zeros perfectly, this would be a terrible idea. In practice, of course, we do not know the plant poles and zeros perfectly, so we cannot exactly cancel these anyway. This inexact cancellation would result in a RHP pole very close to a RHP zero, which is then impossible to control robustly, as we have mentioned above: in fact, the closer the RHP zero is to the RHP pole, the worse the performance limitations will be (for instance as shown below in (7.20)). Overall, attempting to cancel RHP poles and zeros is a very bad idea, and will only make the system more difficult to control. If our plant has RHP poles or zeros, we are stuck with them, and they are what give rise to the fundamental limitations we now study.

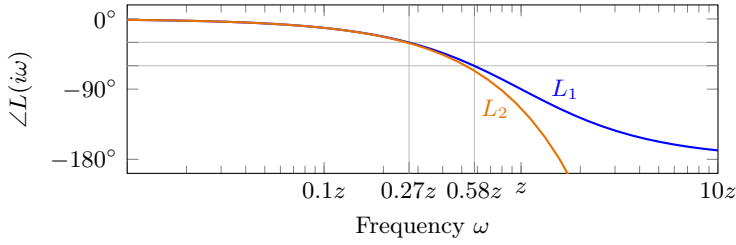


Figure 7.10: Bode phase plot of all-pass transfer function (7.18), and of a time delay $e^{-s\tau}$, for $\tau = 2/z$, showing frequencies for a phase lag of 30° and 60° .

Right-half-plane zeros

Consider the case of a plant $P(s)$ with a single RHP zero at $s = z$. If $P(s)$ has no other right-half-plane poles or zeros, then we may write

$$P(s) = P_{mp}(s) \underbrace{\frac{z-s}{z+s}}_{\text{all pass}}, \quad (7.17)$$

where P_{mp} is minimum-phase (no RHP zeros or poles), and the transfer function

$$P_{ap}(s) = \frac{z-s}{z+s} \quad (7.18)$$

is all-pass (i.e., $|P_{ap}(i\omega)| = 1$ for all ω). Thus, P_{mp} is the same as $P(s)$, except that the right-half-plane zero at $s = z$ has been exchanged for a left-half-plane zero at $s = -z$. Since P_{ap} is all-pass, it follows that $|P(i\omega)| = |P_{mp}(i\omega)|$. The idea behind the bandwidth constraints summarized previously is as follows: for any controller $C(s)$, the loop gain $L = PC$ must cross 0 dB at some point, with some negative slope. If the slope is -1 , then the phase of $P_{mp}(i\omega)$ will be about -90° . If we want a phase margin of at least 30° , then the additional phase lag of P_{ap} at crossover must be at most 60° (that is, $\angle P_{ap}(i\omega_c) > -60^\circ$). From (7.18), we find

$$\angle P_{ap}(i\omega) = -2 \tan^{-1} \frac{\omega}{z},$$

so if φ_l is the allowable phase lag at the crossover frequency (for instance, $\varphi_l = 60^\circ$ for a 30° phase margin), then we must have

$$\omega_c < z \tan(\varphi_l/2).$$

For $\varphi_l = 30^\circ$, this gives $\omega_c < 0.27z$, and for $\varphi_l = 60^\circ$, this gives $\omega_c < 0.58z$, as shown in Figure 7.10.

Type	Constraint	$\varphi_l < 30^\circ$	$\varphi_l < 60^\circ$
RHP pole p	minimum bandwidth	$\omega_c > 3.7p$	$\omega_c > 1.7p$
RHP zero z	maximum bandwidth	$\omega_c < 0.3z$	$\omega_c < 0.6z$
Time delay τ	maximum bandwidth	$\omega_c < 0.5/\tau$	$\omega_c < 1.1/\tau$

Table 7.2: Constraints imposed by time delays and real right-half plane poles and zeros. Constraints on the crossover frequency ω_c are shown for a phase lag of $\varphi_l < 30^\circ$ (for better robustness) and $\varphi_l < 60^\circ$ (if the plant is more certain, so less robustness is required).

Time delays

Next, consider a plant with time delay τ . Recall from Table 2.2 that the transfer function for a time delay is $e^{-s\tau}$. The phase at frequency ω is therefore $-\omega\tau$, so for an maximum phase lag of φ_l at crossover, we require

$$\omega_c < \varphi_l/\tau.$$

For $\varphi_l = 30^\circ$, this corresponds to $\omega_c < 0.52/\tau$, and for $\varphi_l = 60^\circ$, we have $\omega_c < 1.05/\tau$. These values are summarized in Table 7.2.

It is useful to note that a time delay is in many ways similar to a right-half-plane zero. To see why this is the case, consider the Taylor series expansion about $s = 0$:

$$e^{-s\tau} = 1 - s\tau + \frac{1}{2}s^2\tau^2 + O(s^3).$$

For comparison, the Taylor series expansion of P_{ap} in (7.18) is

$$\frac{z-s}{z+s} = 1 - \frac{2}{z}s + \frac{2}{z^2}s^2 + O(s^3).$$

Note that the first three terms match when $\tau = 2/z$. Thus, a time delay τ can be reasonably well approximated by a RHP zero at $z = 2/\tau$, at least for frequencies for which $\omega\tau \ll 1$. Figure 7.10 also shows the Bode phase plot for $e^{-s\tau}$, with $\tau = 2/z$: note that the phase matches closely for $\omega < z$.

Right-half-plane poles

Now, consider an unstable plant $P(s)$ with single RHP pole at $s = p$, and no other RHP poles or zeros. Then we may write

$$P(s) = P_{mp}(s) \frac{s+p}{s-p},$$

where P_{mp} is minimum-phase, and

$$P_{ap}(s) = \frac{s+p}{s-p} \quad (7.19)$$

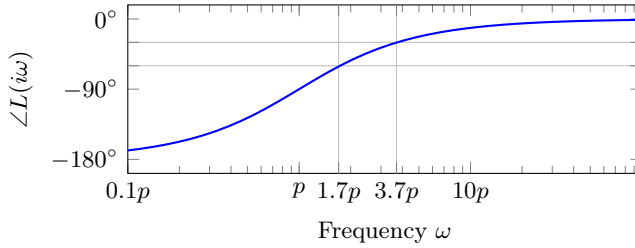


Figure 7.11: Bode phase plot of all-pass transfer function (7.19) for a RHP pole at $s = p$, showing frequencies for a phase lag of 60° and 30° .

is all-pass. The Bode plot of P_{ap} is shown in Figure 7.11. As with Figure 7.10, we look for frequencies for which the phase lag does not exceed some maximum value φ_l . For (7.19), we find

$$\angle P_{ap}(i\omega) = -2 \tan^{-1} \frac{p}{\omega}$$

so for a maximum phase lag of φ_l , we require

$$\omega_c > p / \tan(\varphi_l/2).$$

For a phase margin of 30° , we take $\varphi_l = 60^\circ$, and the constraint becomes $\omega_c > 3.73p$, and for a phase margin of 60° ($\varphi_l = 30^\circ$), we require $\omega_c > 1.7p$. The Bode plot of P_{ap} is shown in Figure 7.11, and the corresponding constraints are summarized in Table 7.2.

Example 7.11. Let us compare three plants given by

$$P_1(s) = \frac{5}{s+1}, \quad P_2(s) = \frac{5}{s+1} \frac{10-s}{10+s}, \quad P_3(s) = \frac{5}{s+1} \frac{s+10}{s-10}.$$

Note that these all have the same Bode magnitude plot, but P_1 is minimum phase, while P_2 has a RHP zero at $s = 10$, and P_3 has a RHP pole at $s = 10$. Figure 7.12 shows the Bode plots for the three plants. For P_1 , the phase never drops below -90° , and there are no fundamental limitations: the bandwidth can be arbitrarily large or small, without affecting robustness.

However, for P_2 , the situation is worse. If we were to use proportional feedback, then it is easy to see from the Bode plot that we could not increase the crossover frequency beyond about 5 rad/sec without making the phase margin too small (since the phase comes close to -180°). This is consistent with the constraint listed in Table 7.2: for a RHP zero at $s = 10$, for a 30° phase margin, we require $\omega_c < 6$ rad/sec, which is consistent with this Bode plot. It is also instructive to compare the Nyquist plots shown in Figure 7.13: for P_1 , the gain margin is infinite, while for P_2 , if we increase

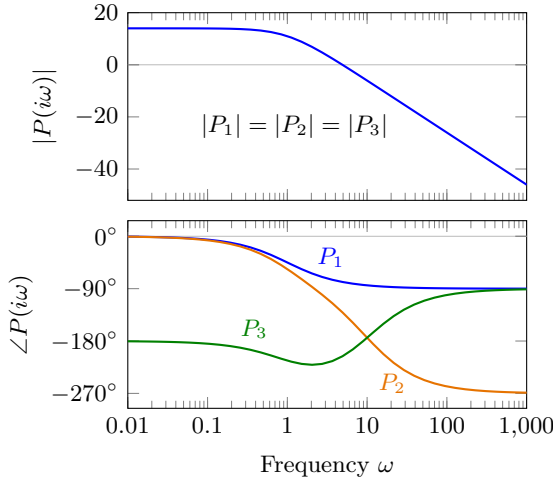


Figure 7.12: Bode plots for Example 7.11. The magnitude plots are identical, but the phase plots differ: P_2 has a maximum bandwidth requirement, and P_3 has a minimum bandwidth requirement.

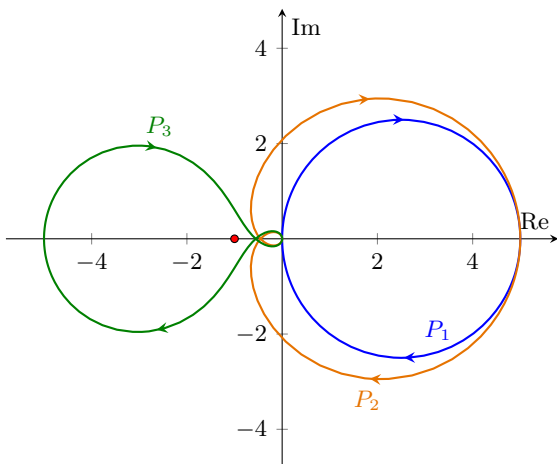
the gain beyond a certain amount, the Nyquist plot will encircle the -1 point and the closed-loop system will be unstable. The important point of this section, though, is that this bandwidth constraint is not just a limitation of proportional feedback: indeed, no controller can do much better than this without making the stability margins unacceptably small.

For P_3 , we see there is a minimum crossover frequency needed for stability. Table 7.2 specifies a constraint of $\omega_c > 17$ rad/sec in order to have a 30° phase margin, and this is consistent with the Bode plot shown in Figure 7.12: the phase does not climb much above -180° until about 20 rad/sec. This behavior is also evident from the Nyquist plot shown in Figure 7.13: since there is now one open-loop RHP pole, we need one *counter-clockwise* encirclement in order for the closed-loop system to be stable. For a unity gain (as shown in the figure), there is one *clockwise* encirclement, so the closed-loop system has two unstable poles. For sufficiently large gain, however, the Nyquist plot would have one counter-clockwise encirclement, and the closed-loop system would be stable. Again, the point of this section is that the bandwidth constraint evident from the Bode plot is a fundamental one, and no controller can stabilize the system robustly with a bandwidth much lower than 17 rad/sec. This result makes sense on intuitive grounds: if there is an unstable pole, then the “reaction time” of the closed-loop system must be sufficiently fast in order to stabilize the pole. Thus, if the bandwidth is not sufficiently high, no controller can robustly stabilize the system. \diamond

Right-half-plane poles and zeros

If a system has a right-half-plane pole near a right-half-plane zero, it will be especially difficult to control. For example, the following result is proved in

Figure 7.13: Nyquist plots for P_1 , P_2 , and P_3 from Example 7.11.



Example B.14 in Appendix B.

Consider a loop transfer function L with a single RHP zero at $z > 0$ and a single RHP pole at $p > 0$. Then the sensitivity function $S = 1/(1 + L)$ satisfies

$$\sup_{\omega \in \mathbb{R}} |S(i\omega)| \geq \left| \frac{p + z}{p - z} \right|. \quad (7.20)$$

Let us see why (7.20) implies severe robustness limitations when a system has a RHP zero close to a RHP pole. If z is close to p , then the right-hand side of (7.20) is large, so $|S(i\omega)|$ must have a large peak at some frequency ω . But recall that $1/|S(i\omega)|$ is the distance from the -1 point to the point $L(i\omega)$ on the Nyquist plot. So if $|S(i\omega)| \gg 1$, then $1/|S(i\omega)| \ll 1$, so $L(i\omega)$ comes very close to the -1 point, and we have poor robustness.

Example 7.12. Suppose we have a system with a right-half-plane pole at $p = 4$ and a right-half-plane zero at $z = 4.1$. Then (7.20) implies

$$\sup_{\omega} |S(i\omega)| \geq \frac{4 + 4.1}{|4 - 4.1|} = 81.$$

Thus, the distance from the -1 point to the Nyquist plot of the loop transfer function $L(i\omega)$ will be *at most* $1/81$, *no matter what controller we use*. \diamond

The methods used in Example B.14 can be used to show many other variants of the fundamental limitations shown here: for instance, even if the plant is stable (no RHP poles), one cannot expect to have good performance ($|S(i\omega)| \ll 1$) at frequencies close to a right-half-plane zero. For a more detailed study of these fundamental limitations, see Doyle et al. [7, Ch. 6].

Exercises

- 7.1.** For the sprung beam considered in Examples 7.4 and 7.7, consider the case $\delta \neq 0$ and show that the system is observable unless $r = 0$ or $J = 1$. In each case, find the unobservable states.
- 7.2.** The linearized equations for a satellite's orbital motion may be described by a state-space system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 5\omega^2 & 0 & 0 & 3\omega \\ 0 & 0 & -3\omega & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Here, the state is $\mathbf{x} = (x_1, x_2, \dot{x}_1, \dot{x}_2)$, where x_1 and x_2 are radial and angular deviations from a circular reference orbit, the inputs $\mathbf{u} = (u_1, u_2)$ are radial and tangential thrusts. Assume $\omega > 0$.

- (a) Show that the system is observable using the measurements $\mathbf{y} = (x_1, x_2)$.
- (b) Suppose that we can make only one measurement, either x_1 or x_2 . Which one should we use? If there are any unobservable states with either choice of measurement, describe what they are.

Appendix A

Linear algebra

In this appendix, we give a brief review of some important results from linear algebra that arise in the study of control systems.

A.1 Eigenvectors and eigenvalues

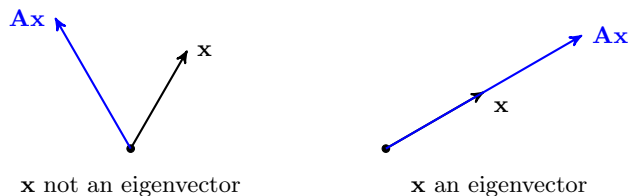
We begin with the concept of eigenvectors and eigenvalues of a square matrix.

Definition A.1. Consider an $n \times n$ matrix \mathbf{A} . A nonzero vector $\mathbf{x} \in \mathbb{C}^n$ is called an *eigenvector* of \mathbf{A} if

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}, \quad \text{for some } \lambda \in \mathbb{C}.$$

The constant λ is called an *eigenvalue* of \mathbf{A} .

Thus, an eigenvector is a special vector for which applying the linear transformation \mathbf{A} changes only the *length* of \mathbf{x} , not its direction. Geometrically, the picture is as follows:



This seems like a very special property, and at first, it is not at all obvious that such a special vector exists, given a particular matrix \mathbf{A} . How could we find such a vector? We wish to find a nonzero \mathbf{x} that solves

$$\begin{aligned} \lambda\mathbf{x} - \mathbf{A}\mathbf{x} &= \mathbf{0} \\ \implies (\lambda\mathbf{I} - \mathbf{A})\mathbf{x} &= \mathbf{0}. \end{aligned} \tag{A.1}$$

(Observe that we needed to insert an identity matrix \mathbf{I} , writing $\lambda \mathbf{x} = \lambda \mathbf{I} \mathbf{x}$, since one cannot add a scalar to a matrix.) One obvious solution to the above equation is $\mathbf{x} = \mathbf{0}$. However, we are interested in a *nonzero* vector \mathbf{x} that satisfies the above. So the only way we can have a nonzero solution is when the matrix $\lambda \mathbf{I} - \mathbf{A}$ fails to be invertible. (If $\lambda \mathbf{I} - \mathbf{A}$ is invertible, then the unique solution is $\mathbf{x} = (\lambda \mathbf{I} - \mathbf{A})^{-1} \mathbf{0} = \mathbf{0}$.) This happens if and only if

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0, \quad (\text{A.2})$$

where \det denotes the determinant of a square matrix. You are probably familiar with the basic properties of determinants, but here are a few of the most important properties:

1. $\det(\mathbf{AB}) = (\det \mathbf{A})(\det \mathbf{B})$
2. If \mathbf{A} is the identity matrix, then $\det \mathbf{A} = 1$.
3. \mathbf{A} is invertible if and only if $\det \mathbf{A} \neq 0$, and in this case $\det(\mathbf{A}^{-1}) = (\det \mathbf{A})^{-1}$.
4. $\det \mathbf{A}^T = \det \mathbf{A}$, where \mathbf{A}^T denotes the transpose of \mathbf{A} .

Note that, if \mathbf{A} is an $n \times n$ matrix, $\det(\lambda \mathbf{I} - \mathbf{A})$ is a polynomial of degree n in λ (called the *characteristic polynomial* of \mathbf{A}). Thus, there is always a solution of $\det(\lambda \mathbf{I} - \mathbf{A}) = 0$ (actually, there are n solutions, counted with multiplicity), so *every matrix \mathbf{A} has an eigenvalue*, and a corresponding eigenvector.

To summarize, the procedure for finding eigenvalues and eigenvectors is as follows:

- First, find λ that satisfies (A.2). These solutions are the eigenvalues of \mathbf{A} .
- For each eigenvalue λ , find a nonzero \mathbf{x} that satisfies (A.1).

Note that the normalization of eigenvectors is arbitrary: if \mathbf{x} satisfies (A.1), then so does $\alpha \mathbf{x}$, for any $\alpha \in \mathbb{C}$.

Example A.2. If \mathbf{A} is diagonal or triangular, one can read the eigenvalues off the diagonal: suppose

$$\mathbf{A} = \begin{bmatrix} a_1 & * & * \\ 0 & a_2 & * \\ 0 & 0 & a_3 \end{bmatrix}.$$

Then

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \det \begin{bmatrix} \lambda - a_1 & * & * \\ 0 & \lambda - a_2 & * \\ 0 & 0 & \lambda - a_3 \end{bmatrix} = (\lambda - a_1)(\lambda - a_2)(\lambda - a_3)$$

and $\det(\lambda \mathbf{I} - \mathbf{A}) = 0$ has solutions $\lambda = a_1, a_2, a_3$. ◇

Symmetric matrices

In general, even if \mathbf{A} is a real matrix, its eigenvalues may be complex: for instance, the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

has characteristic polynomial

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \det \begin{bmatrix} \lambda & -1 \\ 1 & \lambda \end{bmatrix} = \lambda^2 + 1,$$

which has roots $\lambda = \pm i$. However, there is a common special case for which the eigenvalues are guaranteed to be real, namely when \mathbf{A} is *symmetric* (or, for a complex matrix, *Hermitian*).

Definition A.3. A real matrix \mathbf{A} is *symmetric* if $\mathbf{A}^\top = \mathbf{A}$, where \mathbf{A}^\top denotes the transpose. A complex matrix is *Hermitian* if $\mathbf{A}^* = \mathbf{A}$, where \mathbf{A}^* denotes the complex-conjugate transpose.

Clearly, for a real matrix, symmetric and Hermitian are the same thing.

Before stating the main result about eigenvalues and eigenvectors of symmetric or Hermitian matrices, recall that two vectors \mathbf{x} and \mathbf{y} are *orthogonal* if $\mathbf{x} \cdot \mathbf{y} = 0$, where \cdot denotes the dot product. For real vectors, the dot product may be written in matrix form as $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^\top \mathbf{y}$, a row vector \mathbf{x}^\top times a column vector \mathbf{y} . For complex vectors, the dot product is defined by $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^* \mathbf{y}$, where again \mathbf{x}^* denotes the complex-conjugate transpose of \mathbf{x} (the complex conjugate is there so that $\mathbf{x} \cdot \mathbf{x}$ will be real and positive, for any \mathbf{x} ; we can then define the length of a vector \mathbf{x} as $\|\mathbf{x}\| := (\mathbf{x} \cdot \mathbf{x})^{1/2}$). We are now ready for the following important property of symmetric (or Hermitian) matrices:

Theorem A.4. If \mathbf{A} is real and symmetric ($\mathbf{A}^\top = \mathbf{A}$) or complex and Hermitian ($\mathbf{A}^* = \mathbf{A}$), then its eigenvalues are real, and eigenvectors corresponding to distinct eigenvalues are orthogonal.

Proof. Suppose

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \lambda\mathbf{x}, \\ \mathbf{A}\mathbf{y} &= \mu\mathbf{y}, \end{aligned}$$

with $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$. (That is, \mathbf{x} and \mathbf{y} are eigenvectors corresponding to eigenvalues λ and μ .) Then

$$\mathbf{A}\mathbf{x} \cdot \mathbf{y} = (\mathbf{A}\mathbf{x})^* \mathbf{y} = (\lambda\mathbf{x})^* \mathbf{y} = \bar{\lambda}\mathbf{x}^* \mathbf{y} = \bar{\lambda}(\mathbf{x} \cdot \mathbf{y}),$$

where $\bar{\lambda}$ denotes the complex conjugate of λ . But since $\mathbf{A} = \mathbf{A}^*$ (and recalling that $(\mathbf{AB})^* = \mathbf{B}^* \mathbf{A}^*$), we could also have computed

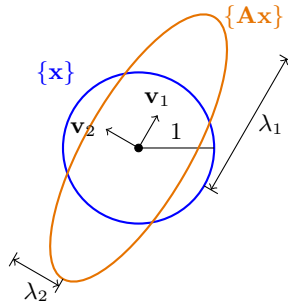
$$\mathbf{A}\mathbf{x} \cdot \mathbf{y} = (\mathbf{A}\mathbf{x})^* \mathbf{y} = \mathbf{x}^* \mathbf{A}^* \mathbf{y} = \mathbf{x}^* \mathbf{A} \mathbf{y} = \mathbf{x}^* \mu \mathbf{y} = \mu(\mathbf{x} \cdot \mathbf{y}).$$

Since these two must be equal, this means

$$(\bar{\lambda} - \mu)(\mathbf{x} \cdot \mathbf{y}) = 0.$$

If we take $\lambda = \mu$ and $\mathbf{x} = \mathbf{y}$, this means $(\lambda - \bar{\lambda})\|\mathbf{x}\|^2 = 0$, but since $\mathbf{x} \neq 0$, this implies $\lambda = \bar{\lambda}$, so λ is real. On the other hand, if $\lambda \neq \mu$ (and therefore $\lambda \neq \bar{\mu}$, since μ is real), then we have $\mathbf{x} \cdot \mathbf{y} = 0$, so \mathbf{x} and \mathbf{y} are orthogonal. \square

In this special case in which \mathbf{A} is symmetric, there is also a nice geometric interpretation of the eigenvalues and eigenvectors. Consider the set of all vectors \mathbf{x} of length 1: that is, all vectors on the “unit ball” $\|\mathbf{x}\| = 1$. Now, map each such \mathbf{x} by the transformation \mathbf{A} . The set of all vectors \mathbf{Ax} forms an ellipsoid; the major axes of the ellipsoid are given by the eigenvectors of \mathbf{A} , and the length of each axis of the ellipsoid is specified by the corresponding eigenvalue. This is illustrated in the figure below:



If $\lambda < 0$, then one gets, in addition, a reflection about the corresponding axis.

A.2 Positive-definite matrices

A real, symmetric matrix \mathbf{M} is called *positive definite* if $\mathbf{x}^T \mathbf{M} \mathbf{x} > 0$ for all $\mathbf{x} \neq 0$. Similarly, if $\mathbf{x}^T \mathbf{M} \mathbf{x} \geq 0$ (instead of being strictly > 0), then \mathbf{M} is called *positive semidefinite*. The expression $\mathbf{x}^T \mathbf{M} \mathbf{x}$ is called a *quadratic form*, and observe that it is a scalar: it is a row vector times a column vector. Positive-definite matrices and their associated quadratic forms arise often in control theory, for instance in the definitions of cost functions for optimal control (see §6.5).

It is common to use the notation $\mathbf{M} > 0$ to mean that the real, symmetric matrix \mathbf{M} is positive definite, and $\mathbf{M} \geq 0$ means that \mathbf{M} is positive semidefinite. (Note that this is *not* the same as all the entries of the matrix \mathbf{M} being positive or non-negative!)

Why do we restrict ourselves to *symmetric* matrices? It turns out that the quadratic form $\mathbf{x}^T \mathbf{M} \mathbf{x}$ depends only on the symmetric part of \mathbf{M} (see Exercise A.1), so we do not gain anything by considering matrices that are not symmetric. Furthermore, as we have seen in the previous section, symmetric

matrices have nice properties, such as having real eigenvalues and orthogonal eigenvectors, so it is convenient to work with them.

The following gives a useful test for when a matrix is positive definite. The proof is straightforward, and is left as an exercise.

Theorem A.5. *A real, symmetric matrix is positive definite if and only if all its eigenvalues are > 0 . A real, symmetric matrix is positive semi-definite if and only if all its eigenvalues are ≥ 0 .*

Any positive-semidefinite matrix \mathbf{M} can be written as $\mathbf{M} = \mathbf{C}^T \mathbf{C}$ for some matrix \mathbf{C} (possibly rectangular), and any positive-definite matrix \mathbf{A} has a unique positive-definite square root \mathbf{B} such that $\mathbf{A} = \mathbf{B}^2$.

A.3 Coordinate changes

Let us now recall how matrices transform under a change of coordinates. Consider solving a system of linear equations given by

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (\text{A.3})$$

Let us change coordinates, writing

$$\mathbf{x} = \mathbf{T}\tilde{\mathbf{x}}, \quad \mathbf{b} = \mathbf{T}\tilde{\mathbf{b}},$$

where \mathbf{T} is an invertible matrix, and $\tilde{\mathbf{x}}$, $\tilde{\mathbf{b}}$ represent \mathbf{x} and \mathbf{b} in the new coordinates. Substituting into (A.3) gives

$$\mathbf{A}\mathbf{T}\tilde{\mathbf{x}} = \mathbf{T}\tilde{\mathbf{b}}$$

or, multiplying by \mathbf{T}^{-1} ,

$$\mathbf{T}^{-1}\mathbf{A}\mathbf{T}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}.$$

This may then be written $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, where

$$\tilde{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}. \quad (\text{A.4})$$

Thus, in the new coordinates, \mathbf{A} transforms to $\mathbf{T}^{-1}\mathbf{A}\mathbf{T}$.

Next, let us look at a system of linear ODEs given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \quad (\text{A.5})$$

and consider the same change of coordinates, $\mathbf{x} = \mathbf{T}\tilde{\mathbf{x}}$. Our ODE becomes

$$\mathbf{T}\dot{\tilde{\mathbf{x}}} = \mathbf{A}\mathbf{T}\tilde{\mathbf{x}}$$

or, multiplying by \mathbf{T}^{-1} ,

$$\dot{\tilde{\mathbf{x}}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\tilde{\mathbf{x}},$$

which may be written $\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}$, with $\tilde{\mathbf{A}}$ again given by (A.4).

This transformation $\mathbf{A} \mapsto \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ is called a *similarity transformation*, and it is how matrices transform under a change of coordinates, either for the linear system (A.3) or for the system of linear ODEs (A.5). An important property of similarity transformations is that they do not change the eigenvalues of a matrix. To show this, note that eigenvalues of $\tilde{\mathbf{A}}$ are roots of

$$\begin{aligned}\det(\lambda\mathbf{I} - \mathbf{T}^{-1}\mathbf{A}\mathbf{T}) &= \det(\mathbf{T}^{-1}(\lambda\mathbf{I} - \mathbf{A})\mathbf{T}) \\ &= (\det \mathbf{T}^{-1}) \det(\lambda\mathbf{I} - \mathbf{A}) (\det \mathbf{T}) \\ &= \det(\lambda\mathbf{I} - \mathbf{A}),\end{aligned}$$

but these roots are eigenvalues of \mathbf{A} . (Note that the last equality holds since $(\det \mathbf{T}^{-1})(\det \mathbf{T}) = \det(\mathbf{T}^{-1}\mathbf{T}) = 1$.)

It is also worth pointing out that both the determinant of a matrix and the trace of a matrix (which is the sum of the diagonal elements) are independent of coordinates (i.e., they do not change under a similarity transformation). These follow from the basic properties $\det(\mathbf{AB}) = (\det \mathbf{A})(\det \mathbf{B})$ and $\text{trace}(\mathbf{AB}) = \text{trace}(\mathbf{BA})$, which hold for any matrices \mathbf{A} and \mathbf{B} of compatible dimension.

Diagonalization

A useful similarity transformation is to consider coordinates aligned with the eigenvectors of \mathbf{A} . In these coordinates, the transformed matrix becomes *diagonal*, so is particularly easy to work with. For instance, we may use this procedure (called *diagonalization*) to compute the matrix exponential $e^{\mathbf{A}t}$ (see §5.1).

Suppose λ_j are eigenvalues of \mathbf{A} , with corresponding eigenvectors \mathbf{v}_j , so $\mathbf{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j$, for $j = 1, \dots, n$. We may write these n relations together in matrix form as

$$\mathbf{A} \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{bmatrix} = \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix},$$

or simply

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda},$$

where \mathbf{V} is a matrix whose columns are eigenvectors of \mathbf{A} , and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. If the eigenvectors \mathbf{v}_j are linearly independent (for instance, this is always true if the n eigenvalues λ_j are distinct), then the matrix \mathbf{V} is invertible, and we can write

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}. \tag{A.6}$$

That is, \mathbf{A} and $\mathbf{\Lambda}$ are related via a similarity transform. This procedure of transforming \mathbf{A} to a set of coordinates (given by \mathbf{V}) in which it is diagonal (given by $\mathbf{\Lambda}$) is called *diagonalization*.

We may illustrate an application of diagonalization by proving the following theorem, which is useful in obtaining tests for controllability and observability:

Theorem A.6 (Cayley-Hamilton theorem). *A square matrix \mathbf{A} satisfies its own characteristic equation.*

That is, if $p(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A})$ denotes the characteristic polynomial of \mathbf{A} , then $p(\mathbf{A})$ is the zero matrix (where in evaluating the polynomial $p(\mathbf{A})$, for instance, $\mathbf{A}^2 = \mathbf{A}\mathbf{A}$, using the usual matrix multiplication). We will prove Theorem A.6 for the special case when \mathbf{A} a diagonalizable matrix, although the theorem holds even if \mathbf{A} is not diagonalizable.

Proof. Assuming \mathbf{A} is diagonalizable, we may write $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where \mathbf{V} is a matrix whose (linearly independent) columns are eigenvectors of \mathbf{A} , and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ is a diagonal matrix whose entries are eigenvalues of \mathbf{A} . Write the characteristic polynomial of \mathbf{A} as

$$p(\lambda) := \lambda^n + c_{n-1}\lambda^{n-1} + \dots + c_1\lambda + c_0,$$

and note that if λ_j is an eigenvalue of \mathbf{A} , then $p(\lambda_j) = 0$. Then

$$\begin{aligned} p(\mathbf{A}) &= p(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}) = (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})^n + c_{n-1}(\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1})^{n-1} + \dots + c_1\mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1} + c_0\mathbf{I} \\ &= \mathbf{V}(\mathbf{\Lambda}^n + c_{n-1}\mathbf{\Lambda}^{n-1} + \dots + c_1\mathbf{\Lambda} + c_0\mathbf{I})\mathbf{V}^{-1}. \end{aligned}$$

Each term in the sum in parentheses is a diagonal matrix, and the sum is a diagonal matrix whose entries are precisely $p(\lambda_j)$, where λ_j are the eigenvalues of \mathbf{A} . Since $p(\lambda_j) = 0$, we therefore have $p(\mathbf{A}) = \mathbf{0}$. \square

In addition, the following corollary follows immediately:

Corollary A.7. *If \mathbf{A} is an $n \times n$ matrix, then \mathbf{A}^n may be written as a linear combination of $\{\mathbf{I}, \mathbf{A}, \mathbf{A}^2, \dots, \mathbf{A}^{n-1}\}$.*

Exercises

A.1. Show that any real square matrix may be written *uniquely* as the sum of a symmetric matrix and a skew-symmetric matrix. That is, for any \mathbf{M} , we may write $\mathbf{M} = \mathbf{S} + \mathbf{A}$, where $\mathbf{S}^T = \mathbf{S}$ and $\mathbf{A}^T = -\mathbf{A}$, and there is only one choice of \mathbf{S}, \mathbf{A} with these properties. Furthermore, show that the quadratic form $\mathbf{x}^T\mathbf{M}\mathbf{x}$ depends only on the symmetric part of \mathbf{M} : in particular, $\mathbf{x}^T\mathbf{M}\mathbf{x} = \mathbf{x}^T\mathbf{S}\mathbf{x}$.

A.2. Prove Theorem A.5.

- A.3.** Show that a real, symmetric 2×2 matrix is positive definite if and only if both its determinant and its trace are positive.

Appendix B

Complex variables

This appendix gives an overview of some results from complex analysis that are useful in control theory. For instance, Bode's gain-phase relation discussed in Section 4.3 follows from Cauchy's theorem, the Nyquist stability theorem in Section 4.5 is a consequence of the argument principle, and many of the fundamental limitations given in Section 7.4 follow from the maximum modulus principle. Here, we give only the highlights necessary to understand these important results in control theory, and we omit proofs of the fundamental results of complex analysis, for which we refer the reader to standard texts, such as [4, 18, 23].

B.1 Basic properties

A *complex number* has the form $z = x + iy$, where x and y are real numbers, and i is an imaginary number that satisfies $i^2 = -1$. The numbers x and y are called the real and imaginary parts of z , respectively, and are denoted

$$x = \operatorname{Re}(z), \quad y = \operatorname{Im}(z).$$

Complex numbers obey the usual rules for addition and multiplication: if $z_1 = x_1 + iy_1$ and $z_2 = x_2 + iy_2$, then

$$z_1 + z_2 = (x_1 + x_2) + i(y_1 + y_2)$$

and

$$\begin{aligned} z_1 z_2 &= (x_1 + iy_1)(x_2 + iy_2) \\ &= x_1 x_2 + ix_1 y_2 + iy_1 x_2 + i^2 y_1 y_2 \\ &= (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + y_1 x_2). \end{aligned}$$

The *complex conjugate* of $z = x + iy$ is defined as $\bar{z} = x - iy$: that is, one just changes the sign of the imaginary part. The *absolute value* of a complex

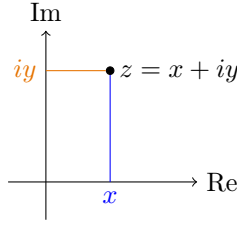


Figure B.1: The complex plane.

number is defined as

$$|z| = (z\bar{z})^{1/2} = (x^2 + y^2)^{1/2},$$

and it is easy to see that $|z_1 z_2| = |z_1| |z_2|$.

The set of all complex numbers is denoted \mathbb{C} . A complex number $z = x + iy \in \mathbb{C}$ may be identified with a point $(x, y) \in \mathbb{R}^2$, and it is common to visualize complex numbers in this manner, as shown in Figure B.1. The real numbers $z = x$ thus correspond to the horizontal axis (called the real axis), and the pure imaginary numbers $z = iy$ correspond to the vertical axis (called the imaginary axis).

A complex number may also be expressed in *polar form* $z = re^{i\theta}$, where $r \geq 0$ is called the *magnitude* (or *modulus*) of z and $\theta \in \mathbb{R}$ is called the *phase* (or *argument*) of z . Here,

$$e^{i\theta} = \cos \theta + i \sin \theta,$$

so since $|e^{i\theta}| = 1$, it follows that $r = |z|$ is the distance from the point z to the origin, and θ is the angle between the point z and the positive real axis (with counterclockwise orientation), as shown in Figure B.2. It is common to write $\theta = \angle z$, but observe that the phase of complex number is determined only up to an additive multiple of 2π , since $e^{2\pi i} = 1$, and hence, for any integer n ,

$$z = re^{i\theta} = re^{i\theta} e^{2\pi i n} = re^{i(\theta + 2\pi n)}.$$

For instance, if $z = i$, the phase of z can be taken to be $\pi/2$ (90°) or $-3\pi/2$ (-270°).

Polar form is particularly useful when multiplying complex numbers: if $z_1 = r_1 e^{i\theta_1}$ and $z_2 = r_2 e^{i\theta_2}$, then

$$z_1 z_2 = (r_1 e^{i\theta_1})(r_2 e^{i\theta_2}) = r_1 r_2 e^{i(\theta_1 + \theta_2)}.$$

That is, the magnitudes r_1, r_2 multiply, and the phases θ_1, θ_2 add.

Finally, we define some terminology. The *open disc* of radius r centered at z_0 , denoted $D_r(z_0)$, is the set of points whose distance from z_0 is strictly less than r :

$$D_r(z_0) = \{z \in \mathbb{C} : |z - z_0| < r\}.$$

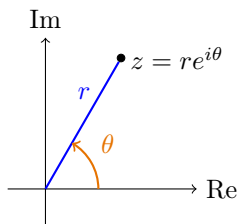


Figure B.2: Polar form of a complex number.

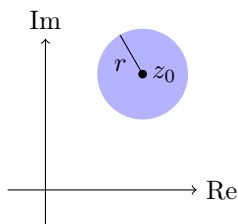


Figure B.3: A neighborhood $D_r(z_0)$ of z_0 .

An open disc centered at z_0 is also called a *neighborhood* of z_0 (see Figure B.3). If Ω is a subset of \mathbb{C} , a point z_0 is called an *interior point* of Ω if there is a neighborhood $D_r(z_0)$ contained in Ω . A set $\Omega \subset \mathbb{C}$ is called *open* if all its points are interior points. A subset Ω is *closed* if its complement $\mathbb{C} - \Omega$ is open. A point $z \in \mathbb{C}$ is a *limit point* of the set Ω if there exists a sequence of points $z_n \in \Omega$ with $z_n \neq z$ such that $\lim_{n \rightarrow \infty} z_n = z$. The *closure* of a set Ω is denoted $\overline{\Omega}$, and consists of the set Ω together with all of its limit points. The *boundary* of a set Ω is the closure of Ω minus the interior of Ω , and is denoted $\partial\Omega$. An open set Ω is called *connected* if it is not possible to find two disjoint non-empty open sets Ω_1 and Ω_2 such that $\Omega = \Omega_1 \cup \Omega_2$. A *region* is a connected open set.

B.2 Functions of a complex variable

We now consider complex-valued functions of a complex variable. We will in particular be interested in functions that are differentiable in the complex sense: such functions are called *holomorphic*. In particular, consider a function f defined on an open set Ω in \mathbb{C} . The function f is holomorphic at the point z_0 if the quotient

$$\frac{f(z_0 + h) - f(z_0)}{h} \quad (\text{B.1})$$

converges to a limit as $h \rightarrow 0$. Here, h is complex and nonzero, with $z_0 + h \in \Omega$, so that the quotient is well defined. If the limit exists, then it is called

the *derivative* of f at z_0 , denoted

$$f'(z_0) = \lim_{h \rightarrow 0} \frac{f(z_0 + h) - f(z_0)}{h}. \quad (\text{B.2})$$

Note that the limit must exist as h approaches zero from any direction (for instance, along the real axis, or along the imaginary axis). Below are some examples of holomorphic functions:

- The function $f(z) = z$ is holomorphic on any open set in \mathbb{C} , with $f'(z) = 1$.

- Any polynomial

$$p(z) = a_0 + a_1z + \cdots + a_nz^n$$

is holomorphic in all of \mathbb{C} , with

$$p'(z) = a_1 + 2a_2z + \cdots + na_nz^{n-1}.$$

- The function $f(z) = 1/z$ is holomorphic on any open set that does not contain the origin, with $f'(z) = -1/z^2$.
- Any rational function $f(z) = n(z)/d(z)$, where n and d are polynomials, is holomorphic everywhere except where $d(z) = 0$.
- The complex exponential $f(z) = e^z$ defined by the power series

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!}$$

is holomorphic in all of \mathbb{C} , with $f'(z) = e^z$. This can be seen by differentiating the series term-by-term (which is justified by Theorem B.1 below).

The above definition and examples might lead one to think that differentiability in the complex sense is not much different from differentiability of a function of a real variable. However, a holomorphic function actually satisfies much stronger properties than a differentiable function of a real variable, such as the following:

- *Regularity:* If f is holomorphic, then derivatives of *all* orders will exist. By contrast, a differentiable function of a real variable may not be twice differentiable, and the first derivative may not even be continuous (see Exercise B.1).
- *Analyticity:* Any holomorphic function is *analytic*, meaning that it has a power series expansion near every point. The converse is also true, namely that any complex function with a power series expansion that converges in an open set Ω is holomorphic in Ω , and for this reason, the

term analytic is often used as a synonym for holomorphic. This is also in contrast with functions of a real variable, which may be infinitely differentiable, yet may not be able to be expanded in a power series (see Exercise B.2).

- *Contour integration:* If f is holomorphic, then for appropriate closed paths γ ,

$$\int_{\gamma} f(z) dz = 0.$$

- *Analytic continuation:* If f and g are holomorphic functions in Ω that are equal in an arbitrarily small disc in Ω , then $f = g$ everywhere in Ω .

Power series

A *power series* centered at $z_0 \in \mathbb{C}$ is an expression of the form

$$\sum_{n=0}^{\infty} a_n (z - z_0)^n \quad (\text{B.3})$$

for some coefficients $a_n \in \mathbb{C}$. Given coefficients a_n , for what values of z does the power series (B.3) converge? It turns out that, for any power series, there is an $0 \leq R \leq \infty$ such that the series converges *absolutely* (i.e., $\sum_n |a_n| |z - z_0|^n$ converges) for $|z - z_0| < R$, and diverges for $|z - z_0| > R$. Thus the region of convergence is a disc centered about z_0 , called the *disc of convergence*, and the radius R is called the *radius of convergence*.

Theorem B.1. *The power series*

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n$$

defines a holomorphic function inside its disc of convergence. Furthermore, the derivative of f can be obtained by differentiating term by term:

$$f'(z) = \sum_{n=0}^{\infty} n a_n (z - z_0)^{n-1},$$

and f' has the same radius of convergence as f .

(For a proof, see, for instance, [23, Theorem 2.5] or [18, Theorem 3.2.4].)

Zeros and poles

A complex number z_0 is a *zero* of f if $f(z_0) = 0$. If f has a zero at z_0 , then in some neighborhood of z_0 , there is a non-vanishing holomorphic function g , and a unique positive integer n such that

$$f(z) = (z - z_0)^n g(z). \quad (\text{B.4})$$

(For a proof, see [23, Chapter 3, Theorem 1.1].) The integer n is called the *order* of the zero, and if $n = 1$, z_0 is called a *simple zero*.

In order to define a pole, we first introduce the notion of a *deleted neighborhood* of a point z_0 , which is simply a neighborhood of z_0 with the point z_0 removed:

$$\{z \in \mathbb{C} : 0 < |z - z_0| < r\}, \quad \text{for some } r > 0.$$

A *point singularity* of a function f is a complex number z_0 such that f is defined in a deleted neighborhood of z_0 , but not at z_0 itself. A *pole* is a particular type of point singularity, for which the function $1/f$, defined to be zero at z_0 , is holomorphic in a full neighborhood of z_0 . For instance, $f(z) = 1/(z - 3)$ has a pole at $z = 3$. If f has a pole at z_0 , then in a neighborhood of z_0 there is a nonvanishing holomorphic function h and a unique positive integer n such that

$$f(z) = (z - z_0)^{-n} h(z). \quad (\text{B.5})$$

The integer n is called the *order* of the pole, and if $n = 1$, z_0 is called a *simple pole*. If f has a pole of order n at z_0 , then

$$f(z) = \frac{a_{-n}}{(z - z_0)^n} + \frac{a_{n-1}}{(z - z_0)^{n-1}} + \cdots + \frac{a_{-1}}{z - z_0} + G(z) \quad (\text{B.6})$$

where G is holomorphic in a neighborhood of z_0 . The coefficient a_{-1} in this expansion plays a special role, and is called the *residue* of f at the pole z_0 , and is denoted $\text{res}_{z_0} f = a_{-1}$.

Functions of a complex variable may have other types of point singularities that are not poles: for instance, $z^{1/2}$, $e^{1/z}$, and $\log z$ (to be discussed below) all have point singularities at the origin, but these are not poles. Point singularities that are not poles are called *essential singularities*. We will be primarily interested in functions whose only singularities are poles, and such functions are called *meromorphic*. Note that rational functions (ratios of polynomials) are meromorphic, and the transfer functions we normally encounter in control theory are rational functions, since they arise from Laplace transforms of ordinary differential equations.

Integration

In order to define integration on the complex plane, we first need to introduce the notion of a curve in \mathbb{C} . A *parameterization* of a curve is a function $z(t)$ that maps a closed interval $[a, b] \subset \mathbb{R}$ to the complex plane. A *curve* $\gamma \subset \mathbb{C}$ is then the image of $[a, b]$ under such a parameterization. A curve has an orientation corresponding to t increasing from a to b . A parameterization $z(t)$ is *smooth* if $z'(t)$ exists and is continuous, and $z'(t) \neq 0$ for $t \in (a, b)$. Two parameterizations $z : [a, b] \rightarrow \mathbb{C}$ and $\tilde{z} : [c, d] \rightarrow \mathbb{C}$ are *equivalent* if there

exists a continuously differentiable bijection $t(\tau)$ from $[c, d]$ to $[a, b]$ such that $t'(\tau) > 0$ and

$$\tilde{z}(\tau) = z(t(\tau)),$$

and equivalent parameterizations define the same curve. A curve γ is smooth if any parameterization $z(t)$ is smooth; a curve is *closed* if $z(a) = z(b)$, and it is *simple* if it does not intersect itself.

Given a smooth curve γ and a function f , one can define the integral of f along γ by

$$\int_{\gamma} f(z) dz = \int_a^b f(z(t)) z'(t) dt,$$

which is independent of the parameterization $z(t)$ of γ . Such an integral is called a *contour integral*. It is straightforward to define integration over piecewise smooth curves as well, by summing the integrals over each smooth portion.

Next, we define a *simply-connected region* of \mathbb{C} . Informally, a simply-connected region is a region without any holes. More precisely, a region Ω is simply connected if any two curves in Ω with the same endpoints are homotopic: that is, one can be deformed into the other by a continuous transformation, without leaving Ω , and with the endpoints fixed. Thus, the disc $D_r(z_0)$ is simply connected, as is the right half-plane. However, the punctured disc (the disc $D_r(z_0)$ with the point z_0 removed) is not simply connected. If a region is simply connected, then we have the following remarkable theorem:

Theorem B.2 (Cauchy's theorem). *If f is holomorphic in a simply connected region Ω , then*

$$\int_{\gamma} f(z) dz = 0 \tag{B.7}$$

for any closed curve γ in Ω .

Note that it is important that the region Ω be simply connected. For instance, if we consider the function $f(z) = 1/z$, and $\Omega = \mathbb{C} - \{0\}$, then f is holomorphic in Ω , although Ω is not simply connected. If we take γ to be the circle of radius r centered at the origin (parameterized by $z(\theta) = re^{i\theta}$ with $\theta \in [0, 2\pi]$), then

$$\int_{\gamma} \frac{1}{z} dz = \int_0^{2\pi} \frac{1}{re^{i\theta}} ire^{i\theta} d\theta = i \int_0^{2\pi} d\theta = 2\pi i.$$

This example in fact motivates a more general formula for evaluating integrals, known as the residue formula.

Theorem B.3 (Residue formula). *Let Ω be a simply connected region containing a simple closed curve γ and its interior, and suppose that f is holomorphic in Ω except for poles at the points z_1, \dots, z_N inside γ . Then*

$$\int_{\gamma} f(z) dz = 2\pi i \sum_{k=1}^N \text{res}_{z_k} f. \tag{B.8}$$

B.3 The logarithm and the argument principle

We now discuss a useful theorem called the argument principle, and use it to prove the Nyquist stability criterion discussed in Section 4.5.

We begin with a discussion of the complex logarithm. Recall that the logarithm is the inverse of the exponential function, so given a function $f(z)$, the logarithm $\log f(z)$ must satisfy $f(z) = e^{\log f(z)}$. Thus, recalling the polar form, we must have

$$\log f(z) = \log |f(z)| + i\angle f(z),$$

where \log on the right-hand side denotes the usual logarithm for real numbers. However, this definition is ambiguous, because as mentioned previously, the phase $\angle f(z)$ is defined only up to an additive multiple of 2π . Thus, the function \log is *multiple-valued*, since the phase is not uniquely specified.

In order to make $\log f(z)$ uniquely defined, we need to choose a unique value for the phase of $f(z)$, and this is called choosing a *branch* of the logarithm. However, this is not always possible. For instance, let Ω be the “punctured plane”, the complex plane with the origin removed. Can we define a function $\log z$ that is holomorphic in Ω ? The answer is no: whatever value we pick for the phase of $z = re^{i\theta}$ (say, $-\pi < \theta \leq \pi$), if z travels in a circle around the origin, the value of $\log z$ would need to be discontinuous at some point (e.g., $\theta = \pi$), so $\log z$ cannot be holomorphic. However, if we let Ω denote the *slit plane*

$$\Omega = \{re^{i\theta} : r > 0, -\pi < \theta < \pi\},$$

(i.e., the complex plane with the negative real axis removed), then the function $\log z = \log r + i\theta$ is holomorphic in Ω .

The key difference between these two examples is that the slit plane is simply connected, while the punctured plane is not. It turns out that, if Ω is simply connected, and f is holomorphic and non-vanishing in Ω , then there is a branch $\log f(z)$ that is holomorphic in Ω .

Regardless of our choice of branch, however, as long as f is holomorphic and nonzero, the derivative of $\log f$ is single valued, and is given by

$$\frac{d}{dz} \log f(z) = \frac{f'(z)}{f(z)}.$$

Thus, the imaginary part of the integral

$$\int_{\gamma} \frac{f'(z)}{f(z)} dz$$

is equal to the net change in the phase (or argument) of $f(z)$ as z travels around the contour γ . By Cauchy’s theorem, we immediately see that if γ is a simple closed contour, and f is holomorphic and non-vanishing inside γ ,

then the integral must be zero, and the net change in phase is zero. However, if f does have poles or zeros inside γ , then the integrand will fail to be holomorphic at those points, and the integral may not be zero. The argument principle relates the net change in phase of $f(z)$ to the number of poles and zeros inside γ .

To be more specific, recall that if f has a zero of order n at z_0 , then in some neighborhood of z_0 we may write

$$f(z) = (z - z_0)^n g(z)$$

for some function g that is holomorphic and non-vanishing. Then

$$\frac{f'(z)}{f(z)} = \frac{n(z - z_0)^{n-1}g(z) + (z - z_0)^n g'(z)}{(z - z_0)^n g(z)} = \frac{n}{z - z_0} + \frac{g'(z)}{g(z)}.$$

Since g is holomorphic and nonvanishing in a neighborhood of z_0 , it follows that g'/g is holomorphic, and we have

$$\operatorname{res}_{z_0}(f'/f) = n.$$

Similarly, if f has a pole of order n at p_0 , then in a neighborhood of p_0 we may write

$$f(z) = (z - p_0)^{-n} g(z),$$

where g is holomorphic and non-vanishing. We calculate

$$\frac{f'(z)}{f(z)} = \frac{-n}{z - p_0} + \frac{g'(z)}{g(z)},$$

so since g'/g is holomorphic, we have

$$\operatorname{res}_{p_0}(f'/f) = -n.$$

In summary, if f has a zero of order n at z_0 , the function f'/f has a pole at z_0 with residue $+n$, and if f has a pole of order n at p_0 , the function f'/f has a pole at p_0 with residue $-n$. The following result then follows directly from the residue formula.

Theorem B.4 (Argument principle). *Suppose f is meromorphic in an open set containing a simple closed curve γ and its interior. Let Z and P denote the number of zeros and poles inside γ , counted with multiplicity. If f has no poles and never vanishes on γ , then*

$$\frac{1}{2\pi i} \int_{\gamma} \frac{f'(z)}{f(z)} dz = Z - P. \quad (\text{B.9})$$

This theorem is called the argument principle because, as explained above, the integral on the left-hand side of (B.9) is simply i times the change in the argument (phase) of $f(z)$ as z traverses the curve γ in a counter-clockwise direction. Thus, we have the following corollary.

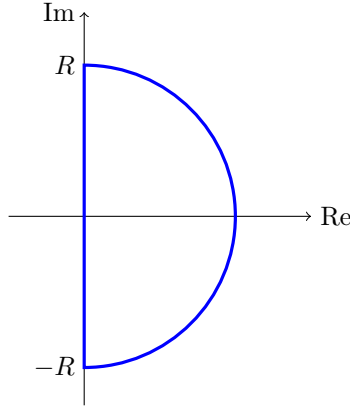


Figure B.4: Nyquist contour.

Corollary B.5. *Suppose f is meromorphic in an open set containing a simple closed curve γ and its interior. Let Z and P denote the number of zeros and poles inside γ , counted with multiplicity, and let N denote the number of times $f(z)$ encircles the origin in a counter-clockwise direction as z traverses the curve γ in a counter-clockwise direction. Then*

$$N = Z - P.$$

The argument principle has a number of interesting applications, among them being the Nyquist stability criterion, which we discuss next.

Proof of the Nyquist criterion

Given a loop transfer function $L(s)$, the Nyquist criterion relates the number of *closed-loop* poles in a particular region of the complex plane (typically the right half-plane) to the number of *open-loop* poles in that same region. The key observation is that the denominator of any closed-loop transfer function is $1 + L$ (see, e.g., Section 4.4), so the closed-loop poles are zeros of the function

$$f(s) = 1 + L(s).$$

On the other hand, open-loop poles (poles of L) are poles of $f = 1 + L$.

Although the Nyquist criterion applies to any simple closed curve γ , the curve is usually taken to be the D-shaped contour shown in Figure B.4, called the *Nyquist contour*. Note that, if we let $R \rightarrow \infty$, the curve encloses the entire right half-plane.

Now, applying the argument principle to our function f , we see that, as s traverses the curve γ , the number of times $f(s)$ encircles the origin is $Z - P$, where Z is the number of closed-loop RHP poles (zeros of f), and P is the number of open-loop RHP poles (poles of f).

Of course, the points of $f(s)$ are just the points of $L(s)$ shifted to the right by 1, so the number of times $f(s)$ encircles the origin is the same as the number of times $L(s)$ encircles the -1 point. Note that the convention for Nyquist plots is that the Nyquist contour is traversed in the *clockwise* direction, which is the opposite of the usual convention for contour integrals. (The reason for this convention for Nyquist plots is so that the portion of the Nyquist contour on the positive imaginary axis is in the direction of *increasing* frequency.) The Nyquist criterion is stated precisely below.

Theorem B.6 (Nyquist criterion). *Consider a loop transfer function L , and a simple closed curve γ . Assume L is meromorphic in an open set containing γ , and has no poles or zeros on γ . Let P be the number of open-loop poles inside γ , Z be the number of closed-loop poles inside γ , and N be the number of times $L(s)$ encircles the -1 point in a clockwise direction as s traverses γ in a clockwise direction. Then*

$$Z = P + N.$$

B.4 Bode's theorems

We now prove some of Bode's key results, namely the integral formula (or area rule), which is given in Theorem 4.20 in Section 4.7, and Bode's gain-phase relation, which is discussed in Section 4.3.

Integral formula

Theorem B.7 (Integral formula). *Let L be a function that satisfies the following:*

- (i) L is holomorphic in the right half-plane except possibly for a finite number of poles p_1, \dots, p_n .
- (ii) $1 + L(s)$ is nonzero for all s with $\operatorname{Re} s \geq 0$,
- (iii) $sL(s) \rightarrow 0$ as $s \rightarrow \infty$ with $\operatorname{Re} s \geq 0$, and
- (iv) $L(-i\omega) = \overline{L(i\omega)}$ for all real ω ,

and let $S = 1/(1 + L)$. Then

$$\int_0^\infty \log |S(i\omega)| d\omega = \pi \sum_{k=1}^n \operatorname{Re} p_k. \quad (\text{B.10})$$

When applying this theorem, normally L is the loop transfer function, so $S = 1/(1 + L)$ is the sensitivity function. Assumption (ii) means that the closed-loop system must be stable (no RHP *closed-loop* poles), and assumption (iii) is equivalent to saying that the loop transfer function “rolls

off" faster than $1/s$ for high frequencies. Assumption (iv) must hold in order for the response to be real for all real input signals, and this always holds if L is a rational function with real coefficients. The proof of the theorem is a straightforward application of Cauchy's theorem, although one needs to be careful about branches of the logarithm.

Proof. First, consider the case in which L has no poles in the right half-plane. Then, using assumption (ii), $1 + L$ has no poles or zeros in the right half-plane, so there is a branch of \log such that $\log S(s) = \log(1/(1 + L(s)))$ is holomorphic in the right half-plane, and $\operatorname{Re} \log S(s) = \log |S(s)|$. Consider the integral of $\log S(s)$ over the D-shaped contour shown in Figure B.4 (in the clockwise direction). Because $\log S$ is holomorphic in the right half-plane, the integral around this closed contour is zero (for any R), by Cauchy's theorem. Furthermore, the integral along the semicircle with radius R goes to zero as $R \rightarrow \infty$. To see this, let $s = Re^{i\theta}$, so the integral becomes

$$\int_{\pi/2}^{-\pi/2} iRe^{i\theta} \log S(Re^{i\theta}) d\theta.$$

Note that $L(s) \rightarrow 0$ as $s \rightarrow \infty$ (by assumption (iii)), and for small $L(s)$, we have $\log(1/(1 + L(s))) = -L(s) + O(L(s)^2)$, so the integrand is $-isL(s) + O(sL(s)^2)$, which goes to zero as $R \rightarrow \infty$ (again using assumption (iii)).

The integral along the imaginary axis is

$$I_\omega = \int_{-R}^R i \log S(i\omega) d\omega,$$

so we have

$$\operatorname{Im} I_\omega = \int_{-R}^R \log |S(i\omega)| d\omega = 2 \int_0^R \log |S(i\omega)| d\omega, \quad (\text{B.11})$$

since $|S(-i\omega)| = |S(i\omega)|$, by assumption (iv). Thus, since $I_\omega \rightarrow 0$ as $R \rightarrow \infty$, (B.10) then follows (with the right-hand side equal to zero).

Next, consider the case in which L does have right-half-plane poles at p_1, \dots, p_n . Then $S = 1/(1 + L)$ has right-half-plane *zeros* at these points, so these are branch points of $\log S$. Let Ω be a region consisting of the (closed) right half-plane with horizontal slits to the left of each pole p_k removed (see Figure B.5). We will define a branch of $\log S$ on the simply connected region Ω . Since S has zeros at p_1, \dots, p_n , we may write

$$S(s) = (s - p_1) \cdots (s - p_n) f(s)$$

where f is holomorphic and nonzero in the right half-plane. Letting $s - p_k = r_k e^{i\theta_k}$, with $-\pi < \theta_k < \pi$, we may define a branch of $\log S$ by

$$\log S(s) = \sum_{k=1}^n \log r_k + i \sum_k \theta_k + \log f(s), \quad (\text{B.12})$$

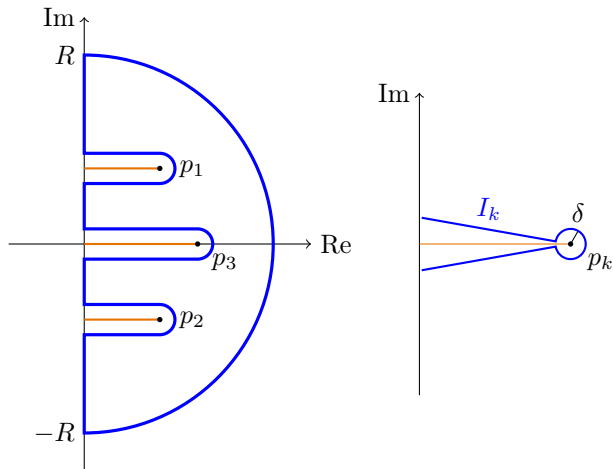


Figure B.5: Contour used in the proof of Theorem B.7 (left), and a detail of the “detour” around pole p_k .

where $\log f(s)$ is holomorphic in the entire right half-plane. Consider now the contour integral shown in Figure B.5, similar to the D-shaped contour from before (also in the clockwise direction), but with “detours” around each of the poles, so that the contour stays in the region Ω . As before, the integral around the entire contour is zero, by Cauchy’s theorem, and the integral around the large semicircle of radius R also goes to zero as $R \rightarrow \infty$. Let I_k denote the integral along the “detour” around pole p_k , as shown in Figure B.5. The integral around the circular portion encircling p_k is

$$\int_{-\pi+\varepsilon}^{\pi-\varepsilon} i\delta e^{i\theta} \log S(p_k + \delta e^{i\theta}) d\theta.$$

As $\delta, \varepsilon \rightarrow 0$, this goes to

$$\int_{-\pi}^{\pi} i\delta e^{i\theta} \log \delta d\theta \rightarrow 0.$$

The integrals along the straight segments of this detour become

$$\int_{\operatorname{Re} p_k}^{\delta} e^{i(-\pi+\varepsilon)} \log S(p_k + r e^{i(-\pi+\varepsilon)}) dr + \int_{\delta}^{\operatorname{Re} p_k} e^{i(\pi-\varepsilon)} \log S(p_k + r e^{i(\pi-\varepsilon)}) dr.$$

As $\varepsilon \rightarrow 0$, all the terms in (B.12) cancel out except for the term $i\theta_k$, and we are left with

$$I_k = \lim_{\delta \rightarrow 0} \left[\int_{\operatorname{Re} p_k}^{\delta} e^{-i\pi} (-i\pi) dr + \int_{\delta}^{\operatorname{Re} p_k} e^{i\pi} (i\pi) dr \right] = -2\pi i \operatorname{Re} p_k.$$

Finally, letting I_ω denote the portions of the integral along the imaginary axis, (B.11) still holds, so since $I_\omega + \sum_k I_k = 0$, we have

$$\int_0^\infty \log |S(i\omega)| d\omega = -\frac{1}{2} \sum_{k=1}^n \operatorname{Im} I_k = \pi \sum_{k=1}^n \operatorname{Re} p_k,$$

as was to be shown. \square

Gain-phase relation

We now turn to Bode's gain-phase relation, discussed previously in Section 4.3, which relates the phase of a nonminimum-phase transfer function to the slope of its magnitude plot. We begin with the following result, due to Bode [3, §14.3], which describes a relationship between the real and imaginary parts of a complex function.

Theorem B.8. *Assume the function f satisfies the following:*

- *f is holomorphic in the open right half-plane;*
- *f has no poles on the imaginary axis (but may have logarithmic singularities)*
- *$f(s)/s \rightarrow 0$ as $s \rightarrow \infty$ with $\operatorname{Re} s \geq 0$;*
- *$f(-i\omega) = \overline{f(i\omega)}$ for all real ω .*

Letting $f(i\omega) = u(\omega) + iv(\omega)$, where u and v are real-valued functions of a real variable, then for any ω_0 ,

$$v(\omega_0) = \frac{2\omega_0}{\pi} \int_0^\infty \frac{u(\omega) - u(\omega_0)}{\omega^2 - \omega_0^2} d\omega. \quad (\text{B.13})$$

Proof. Let γ be the contour shown in Figure B.6, and consider the integral

$$\int_\gamma \frac{f(s) - u(\omega_0)}{s^2 + \omega_0^2} ds.$$

Note that the integrand is holomorphic inside the contour, so by Cauchy's theorem, the integral around all of γ is zero. Furthermore, the integral around the large circle goes to zero as $R \rightarrow \infty$, since $f(s)/s \rightarrow 0$ as $s \rightarrow \infty$. The integral around the semicircle centered at $i\omega_0$ is

$$\lim_{\varepsilon \rightarrow 0} \int_{-\pi/2}^{\pi/2} \frac{f(i\omega_0 + \varepsilon e^{i\theta}) - u(\omega_0)}{(i\omega_0 + \varepsilon e^{i\theta})^2 + \omega_0^2} i\varepsilon e^{i\theta} d\theta = \int_{-\pi/2}^{\pi/2} \frac{iv(\omega_0)}{2\omega_0} d\theta = \frac{i\pi}{2\omega_0} v(\omega_0),$$

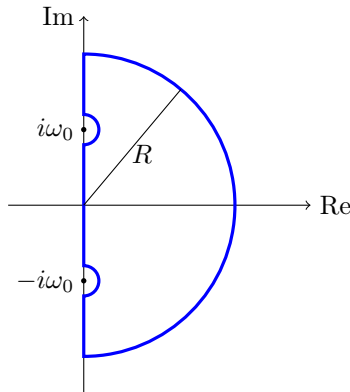


Figure B.6: Contour γ used in the proof of Theorem B.8.

and the integral about the other semicircle centered at $-i\omega_0$ is the same. Thus, since the overall contour integral must be zero, we have

$$\int_{-\infty}^{\infty} \frac{f(i\omega) - u(\omega_0)}{-\omega^2 + \omega_0^2} i d\omega + \frac{i\pi}{\omega_0} v(\omega_0) = 0,$$

and hence

$$v(\omega_0) = \frac{\omega_0}{\pi} \int_{-\infty}^{\infty} \frac{u(\omega) + iv(\omega) - u(\omega_0)}{\omega^2 - \omega_0^2} d\omega.$$

Because $f(-i\omega) = \overline{f(i\omega)}$, it follows that $u(-\omega) = u(\omega)$ and $v(-\omega) = -v(\omega)$, so splitting the integral into positive and negative values of ω gives (B.13). \square

As Bode points out, it is not actually necessary to include the term $-u(\omega_0)$ in the numerator of (B.13), since this term does not contribute to the integral, but including this term makes the integrand better behaved (in particular, Riemann integrable), and avoids needing to use fancier notions of integration, such as the Cauchy principal value integral.

Theorem B.9. Assume f satisfies the conditions of Theorem B.8, with $f(i\omega) = u(\omega) + iv(\omega)$. Fix $\omega_0 \in \mathbb{R}$, and let $\lambda = \log(\omega/\omega_0)$. Then

$$v(\omega_0) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{du}{d\lambda} \log \coth \frac{|\lambda|}{2} d\lambda. \quad (\text{B.14})$$

Proof. Apply Theorem B.8, letting $\tilde{u}(\lambda) = u(\omega) = u(\omega_0 e^\lambda)$, and noting that

$d\lambda = d\omega/\omega$, to obtain

$$\begin{aligned}
 v(\omega_0) &= \frac{2}{\pi} \int_0^\infty \frac{u(\omega) - u(\omega_0)}{\omega/\omega_0 - \omega_0/\omega} \frac{d\omega}{\omega} \\
 &= \frac{2}{\pi} \int_{-\infty}^\infty \frac{\tilde{u}(\lambda) - u(\omega_0)}{e^\lambda - e^{-\lambda}} d\lambda \\
 &= \frac{1}{\pi} \int_{-\infty}^\infty \frac{\tilde{u} - u(\omega_0)}{\sinh \lambda} d\lambda. \\
 &= \frac{1}{\pi} \lim_{\varepsilon \rightarrow 0} \left(\int_{-\infty}^{-\varepsilon} \frac{\tilde{u} - u(\omega_0)}{\sinh \lambda} d\lambda + \int_{\varepsilon}^\infty \frac{\tilde{u} - u(\omega_0)}{\sinh \lambda} d\lambda \right).
 \end{aligned}$$

Denoting the integrals in parentheses by I_1 and I_2 and integrating by parts, we find

$$\begin{aligned}
 I_1 &= \left[(\tilde{u}(\lambda) - u(\omega_0)) \log \coth \frac{-\lambda}{2} \right]_{-\infty}^{-\varepsilon} + \int_{-\infty}^{-\varepsilon} \frac{d\tilde{u}}{d\lambda} \log \coth \frac{-\lambda}{2} d\lambda, \\
 I_2 &= - \left[(\tilde{u}(\lambda) - u(\omega_0)) \log \coth \frac{\lambda}{2} \right]_{\varepsilon}^{\infty} + \int_{\varepsilon}^{\infty} \frac{d\tilde{u}}{d\lambda} \log \coth \frac{\lambda}{2} d\lambda.
 \end{aligned}$$

First, let us look at the boundary terms. As $\lambda \rightarrow 0$, we have $\log \coth(|\lambda|/2) \approx -\log(|\lambda|/2)$, while $\tilde{u}(\lambda) - u(\omega_0)$ is linear in λ . Overall, then, this term is $O(-\lambda \log \lambda) \rightarrow 0$ as $\lambda \rightarrow 0$, so these boundary terms vanish as $\varepsilon \rightarrow 0$, for both I_1 and I_2 .

On the other hand, as $\lambda \rightarrow \infty$, we have

$$\log \coth \frac{\lambda}{2} = \log \frac{1 + e^{-\lambda}}{1 - e^{-\lambda}} = \log(1 + e^{-\lambda}) - \log(1 - e^{-\lambda}) = 2e^{-\lambda} + O(e^{-2\lambda}),$$

since $\log(1 + x) = x + O(x^2)$ as $x \rightarrow 0$. Noting that $e^{-\lambda} = \omega_0/\omega$, the boundary term in I_2 becomes $\lim_{\omega \rightarrow \infty} 2\omega_0 u(\omega)/\omega$, which vanishes, since we have assumed $f(s)/s \rightarrow 0$ as $s \rightarrow \infty$. As $\lambda \rightarrow -\infty$, we have $\log \coth(-\lambda/2) \approx 2e^\lambda = 2\omega/\omega_0$, so this boundary term in I_1 also vanishes as $\lambda \rightarrow -\infty$ (i.e., $\omega \rightarrow 0^+$), since $f(s)s \rightarrow 0$ as $s \rightarrow 0$ (which is true because f does not have a pole at $s = 0$).

Altogether, we have

$$\begin{aligned}
 v(\omega_0) &= \frac{1}{\pi} \lim_{\varepsilon \rightarrow 0} (I_1 + I_2) \\
 &= \frac{1}{\pi} \left(\int_{-\infty}^{0^-} \frac{d\tilde{u}}{d\lambda} \log \coth \frac{-\lambda}{2} d\lambda + \int_{0^+}^{\infty} \frac{d\tilde{u}}{d\lambda} \log \coth \frac{\lambda}{2} d\lambda \right),
 \end{aligned}$$

which agrees with (B.14). □

Corollary B.10 (Bode's gain-phase relation). *Assume $G(s)$ is a rational function with real coefficients and no poles or zeros in the open right half-plane, and such that $G(x_0) > 0$ for some real $x_0 > 0$. Letting $\lambda = \log(\omega/\omega_0)$*

for some ω_0 , then

$$\angle G(i\omega_0) = \frac{\pi}{2} \int_{-\infty}^{\infty} \frac{d \log |G(i\omega)|}{d\lambda} \rho(\lambda) d\lambda,$$

where

$$\rho(\lambda) = \frac{2}{\pi^2} \log \coth \frac{|\lambda|}{2} = \frac{2}{\pi^2} \log \left| \frac{\omega + \omega_0}{\omega - \omega_0} \right|.$$

Proof. The result follows immediately from Theorem B.9, with

$$f(s) = \log G(s) = \log |G(s)| + i\angle G(s).$$

It remains only to verify that f satisfies the conditions of the theorem. Let Ω denote the open right half-plane, and note that f is holomorphic in Ω , because $G(s)$ has no poles or zeros in Ω . Furthermore, f has no poles on the imaginary axis: even if G has poles or zeros on the imaginary axis, f will have logarithmic singularities at these locations, and these are allowable. Now, consider whether $f(s)/s \rightarrow 0$ as $s \rightarrow \infty$. Since G is a rational function, G grows no faster than s^n for some integer n , so $\log G(s)/s \rightarrow 0$ as $s \rightarrow \infty$.

Finally, we show that $f(\bar{s}) = \overline{f(s)}$ for all s in Ω . There is a subtlety here, in that we must choose an appropriate branch of the logarithm. In particular, define f by

$$f(s) = \int_{\gamma} \frac{G'(z)}{G(z)} dz + \log G(x_0),$$

where x_0 is the real value mentioned in the statement of the theorem, $\log G(x_0)$ is the standard logarithm of the real number $G(x_0) > 0$, and γ is a curve in Ω connecting x_0 to s . (The integral is independent of the path chosen, because G is holomorphic and nonvanishing in Ω , and hence G'/G is holomorphic.) In order to verify that f defines a branch of $\log G$, first note that $f' = G'/G$. An easy calculation then shows that

$$\frac{d}{ds} (G(s)e^{-f(s)}) = 0,$$

so $G(s)e^{-f(s)}$ is constant on Ω , and since $e^{f(x_0)} = G(x_0)$, we have $e^{f(s)} = G(s)$ for all $s \in \Omega$.

Finally, to show that $f(\bar{s}) = \overline{f(s)}$, note that $G(\bar{s}) = \overline{G(s)}$, since G is a rational function with real coefficients, and hence

$$\begin{aligned} f(\bar{s}) &= \int_{\bar{\gamma}} \frac{G'(z)}{G(z)} dz + \log G(x) \\ &= \int_{\gamma} \frac{G'(\bar{z})}{G(\bar{z})} dz + \log G(x) \\ &= \int_{\gamma} \frac{\overline{G'(z)}}{\overline{G(z)}} dz + \log G(x) \\ &= \overline{f(s)}. \end{aligned}$$

□

Note that it is important that $G(x_0) > 0$ for some $x_0 > 0$. For instance, consider the functions

$$G(s) = \frac{1}{s+1}, \quad H(s) = \frac{-1}{s+1},$$

for which $|G(i\omega)| = |H(i\omega)|$, but $\angle G(i\omega)$ differs from $\angle H(i\omega)$ by 180° . Bode's gain-phase theorem applies to G , but not to H , since $G(1) = 1/2 > 0$, yet $H(1) = -1/2 < 0$. (Of course, since G has no zeros in the right half-plane, $G(x_0)$ will have the same sign for any $x_0 > 0$.)

It is also worth pointing out that the function ρ satisfies

$$\int_{-\infty}^{\infty} \rho(\lambda) d\lambda = 1,$$

so the formula (B.14) may be viewed as a weighted average of the slope $d \log |G| / d \log \omega$, and if this slope is roughly constant (say m), then $\angle G(i\omega_0) = m\pi/2$.

B.5 Maximum modulus principle

The following useful result places constraints on the maximum of a function f in a given region (which means the maximum of its modulus, or absolute value $|f(z)|$).

Theorem B.11 (Maximum modulus principle). *If f is a non-constant function that is holomorphic in a region Ω , then f cannot attain a maximum in Ω .*

A set is called *compact* if it is closed and bounded, and a standard result from real analysis is that a continuous function on a compact set achieves its maximum and minimum. If the region Ω has compact closure, then we have the following corollary, which says that f achieves its maximum on the boundary:

Corollary B.12. *If Ω is a region with compact closure $\overline{\Omega}$, and f is holomorphic on Ω and continuous on $\overline{\Omega}$, then*

$$\sup_{z \in \Omega} |f(z)| \leq \max_{z \in \partial\Omega} |f(z)|.$$

An intuitive way to think about this result is as a heat conduction problem: if Ω is a region with compact closure, and the temperature is specified on the boundary of Ω , then the steady-state temperature on the interior of Ω cannot exceed the maximum temperature on the boundary. This analogy can even be made precise, since the real and imaginary parts of holomorphic functions satisfy Laplace's equation, which also describes steady-state heat conduction.

Note that in this corollary, it is crucial that the region Ω be compact. For instance, if we take the region Ω to be the open right half-plane (which is obviously not bounded), and $f(z) = e^z$, which is holomorphic in all of \mathbb{C} , then we might be tempted to conclude that $|f(z)| \leq 1$ for all z in the right half-plane, since the boundary of Ω is the imaginary axis, and on that boundary $z = iy$, $|f(z)| = |e^{iy}| = 1$. But actually $|f(z)|$ is unbounded in the right half-plane, since $|f(x + iy)| = e^x$ which is unbounded as $x \rightarrow \infty$. However, with some restrictions on f , we can prove a version of the corollary that applies to the right half-plane.

Corollary B.13. *If G is a real-rational function (that is, a rational function with real coefficients) with no poles in the open right half-plane, then*

$$\sup_{\operatorname{Re} s > 0} |G(s)| \leq \sup_{\omega \in \mathbb{R}} |G(i\omega)|. \quad (\text{B.15})$$

Proof. If $|G(i\omega)|$ is unbounded, then obviously (B.15) holds, so assume this quantity is bounded. The idea is to use a conformal mapping to map the right half-plane to the unit disc. Consider the mapping

$$z = f(s) = \frac{1 - s}{1 + s},$$

which is its own inverse: $s = f(z)$. It is straightforward to see that f maps the right half-plane to the unit disc D , since if s is in the right half-plane, then s is closer to $+1$ than to -1 , and hence $|1 - s| < |1 + s|$, so $|f(s)| < 1$. Furthermore, if $s = i\omega$ for real ω , then $|z| = 1$. If $z = e^{i\theta}$, then

$$s = \frac{1 - e^{i\theta}}{1 + e^{i\theta}} = \frac{e^{-i\theta/2} - e^{i\theta/2}}{e^{-i\theta/2} + e^{i\theta/2}} = -i \tan(\theta/2).$$

so $s = i\omega$ corresponds to $z = e^{i\theta}$ with $\omega = -\tan(\theta/2)$.

We now define a function \tilde{G} , defined on the closed unit disc, such that

$$\tilde{G}(z) = G(f(z)) = G(s).$$

The idea is to show that \tilde{G} is holomorphic on the open unit disc D and continuous on the unit circle, so that we can apply Corollary B.12, to obtain

$$\sup_{\operatorname{Re} s > 0} |G(s)| = \sup_{z \in D} |\tilde{G}(z)| \leq \max_{\theta \in [-\pi, \pi]} |\tilde{G}(e^{i\theta})| = \sup_{\omega \in \mathbb{R}} |G(i\omega)|.$$

Since both G and f are holomorphic in the right half-plane, \tilde{G} is holomorphic in the unit disc. All that remains is to show that \tilde{G} is continuous on the unit circle. Since we are assuming $G(i\omega)$ is bounded, it follows that $\tilde{G}(e^{i\theta})$ is bounded. Furthermore, G is continuous on the imaginary axis, so $\tilde{G}(e^{i\theta})$ is continuous for $|\theta| < \pi$. Finally, since G is a real-rational function, $\lim_{\omega \rightarrow \infty} G(i\omega)$ is real, and thus

$$\lim_{\omega \rightarrow \infty} G(-i\omega) = \lim_{\omega \rightarrow \infty} \overline{G(i\omega)} = \lim_{\omega \rightarrow \infty} G(i\omega),$$

so $\tilde{G}(e^{i\theta})$ is continuous at $\theta = \pi$. □

The example below demonstrates how the maximum modulus theorem can be used to provide useful results in control theory.

Example B.14. Consider a loop transfer function L that is a real-rational function with no poles or zeros in the open right half-plane, except for a single zero at $z_0 > 0$ and a single pole at $p_0 > 0$. Also assume that $1 + L$ has no zeros in the open right half-plane. Then the sensitivity function $S = 1/(1 + L)$ satisfies

$$\sup_{\omega \in \mathbb{R}} |S(i\omega)| \geq \left| \frac{p_0 + z_0}{p_0 - z_0} \right|. \quad (\text{B.16})$$

First, a remark about why this result is useful. The assumptions of the theorem guarantee that the closed-loop system is stable (i.e., S has no right-half-plane poles). However, if L has a right-half-plane pole p_0 close to a right-half-plane zero z_0 , then the right-hand side of (B.16) is large, so the sensitivity function $|S(i\omega)|$ must have a large peak for some ω . This then implies that the feedback system will necessarily have very poor robustness (see, for instance, Sections 4.7 and 7.4), regardless of other details of the plant and controller, such as phase margin, bandwidth, etc.

To see why (B.16) is true, first observe that, since $S = 1/(1 + L)$ and $1 + L$ has no zeros in the right half-plane, it follows that S is a real-rational function with no poles in the right half-plane. Furthermore, since $L(z_0) = 0$ and L has a pole at p_0 , this implies

$$S(z_0) = 1, \quad S(p_0) = 0.$$

Since S has a zero at p_0 , we can write

$$S(s) = S_{mp}(s) \frac{s - p_0}{s + p_0},$$

where S_{mp} has no poles or zeros in the right half-plane (i.e., it is “minimum phase,” to use Bode’s terminology). Note that, for real ω , we have

$$|S(i\omega)| = |S_{mp}(i\omega)| \left| \frac{i\omega - p_0}{i\omega + p_0} \right| = |S_{mp}(i\omega)|.$$

Furthermore, since $S(z_0) = 1$, we have

$$S_{mp}(z_0) = \frac{z_0 + p_0}{z_0 - p_0}.$$

Finally, applying Corollary B.13, we obtain

$$\sup_{\omega \in \mathbb{R}} |S(i\omega)| = \sup_{\omega \in \mathbb{R}} |S_{mp}(i\omega)| \geq |S_{mp}(z_0)| = \left| \frac{z_0 + p_0}{z_0 - p_0} \right|.$$

◇

This example implies that a plant with a right-half-plane zero close to a right-half-plane pole cannot be stabilized robustly, no matter what controller one uses. In this sense, it implies that such a plant has *fundamental* limitations, as discussed in Section 7.4.

Exercises

B.1. Consider the following function f of a real variable x :

$$f(x) = \begin{cases} x^2 \sin(1/x) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0. \end{cases}$$

Show that f is differentiable for all $x \in \mathbb{R}$, but the derivative f' is not continuous at $x = 0$.

B.2. Consider the following function f of a real variable x :

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ e^{-1/x^2} & \text{if } x > 0. \end{cases}$$

Show that f is infinitely differentiable for all $x \in \mathbb{R}$, yet f does not have a converging power series expansion $\sum_{n=0}^{\infty} a_n x^n$ for x near the origin.

B.3. Suppose L satisfies the assumptions of Theorem B.7, and L has a finite number of right-half-plane zeros given by z_1, \dots, z_n . If $T = L/(1 + L)$ denotes the complementary sensitivity function, prove that

$$\int_0^{\infty} \frac{\log |T(i\omega)|}{\omega^2} d\omega = \pi \sum_{k=1}^n \frac{1}{z_k}.$$

Appendix C

MATLAB commands

MATLAB has extensive support for control systems, mostly through its toolboxes (mainly the Control System Toolbox, but also including the Robust Control Toolbox and System Identification Toolbox). This appendix summarizes some commands especially useful for MAE 433, but of course, feel free to explore other commands that may be useful to you.

Getting help

MATLAB is very well documented software. To get help for a command, run

```
>> help [command]
```

to display the help information in the command window, or

```
>> doc [command]
```

to display it in a separate window. Sometimes, `doc` will give information that is more detailed and easier to read. At the end of help texts, you can find links to related functions. The MATLAB documentation is also available at <http://www.mathworks.com/help/techdoc/>.

Generally useful functions

- `linspace`: Linearly spaced vector.
Example: `t = linspace(0,2*pi); plot(t, sin(t))`
- `logspace`: Logarithmically spaced vector.
- `db2mag`: dB to magnitude conversion.
- `mag2db`: Magnitude to dB conversion.
- `conv`: Convolution and polynomial multiplication.
- `eye`: Identity matrix.

- **rank**: Matrix rank.
- **eig**: Eigenvalues and eigenvectors.
- **polyfit**: Fit polynomial to data.
- **polyval**: Evaluate polynomial.
- **residue**: Partial-fraction expansion (residues).

System construction and conversion

- **ss**: Construct state-space model or convert model to state space.
- **tf**: Construct transfer function or convert to transfer function.
- **c2d**: Converts continuous-time dynamic system to discrete time.

System information

- **ssdata**: Quick access to state-space data.
Example: `[A, B, C, D] = ssdata(sys)`
This is equivalent to `A = sys.a, B = sys.b`, etc.
- **tfddata**: Quick access to transfer function data.
Example: `[num, den] = tfdata(sys)`
This is equivalent to `num = sys.num, den = sys.den`.
- **pole**: Computes the poles of linear systems.
- **zero**: Computes zeros and gain of a SISO linear system.
- **pzmap**: Pole-zero map of dynamic systems.

Arithmetic and feedback interconnections

- **+, -**: Addition/subtraction for input/output models.
This is equivalent to parallel interconnection.
- *****: Multiplies two input/output models together.
This is equivalent to series interconnection.
- **feedback**: Feedback connection of two input/output systems.
Note that `G = feedback(P, K)` mathematically cancels poles and zeros properly, whereas `G = P / (1 + P * K)` does not.

Time domain analysis

- `step`: Step response of dynamic systems.
- `impulse`: Impulse response of dynamic systems.
Also may be viewed as computing an inverse Laplace transform.
- `initial`: Initial condition response of state-space models.
- `lsim`: Simulate time response of dynamic systems to arbitrary inputs.

Frequency domain analysis

- `freqresp`: Frequency response evaluated on a grid.
- `bode`: Bode plot of frequency response.
- `nyquist`: Nyquist plot of frequency response.
- `margin`: Gain and phase margins and crossover frequencies.
(returned as arrays and printed on a Bode plot)
- `allmargin`: All stability margins and crossover frequencies.
(returned in a structure and printed to the screen)
- `bandwidth`: Frequency response bandwidth.
- `rlocus`: Root locus plot.

Modern control design

- `ctrb`: Compute the controllability matrix.
- `obsv`: Compute the observability matrix.
- `place`: Closed-loop pole assignment using state feedback.
- `lqr`: Linear-quadratic regulator design for state space systems.
- `lqe`: Kalman estimator design for continuous-time systems.
- `lqg`: Synthesis of LQG regulators and servo-controllers.

Bibliography

- [1] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.
- [2] P. R. Bélanger. *Control Engineering: A Modern Approach*. Saunders College Publishing, 1995.
- [3] H. W. Bode. *Network analysis and feedback amplifier design*. Van Nostrand, New York, 1945.
- [4] R. V. Churchill and J. W. Brown. *Complex Variables and Applications*. McGraw-Hill, fifth edition, 1990.
- [5] E. A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. McGraw-Hill, 1955.
- [6] J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Trans. Automat. Contr.*, 23(4):756–757, 1978.
- [7] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback Control Theory*. Macmillan, 1992.
- [8] J. J. Duistermaat and J. A. C. Kolk. *Lie Groups*. Springer-Verlag, 1999.
- [9] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*, volume 36 of *Texts in Applied Mathematics*. Springer-Verlag, 1999.
- [10] W. R. Evans. Graphical analysis of control systems. *Transactions of the American Institute of Electrical Engineers*, 67(1):547–551, 1948.
- [11] G. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice-Hall, 6th edition, 2009.
- [12] B. Friedland. *Control System Design: An Introduction to State-Space Methods*. Dover, 2005.
- [13] H. Goldstein. *Classical Mechanics*. Addison-Wesley, second edition, 1980.

- [14] L. N. Hand and J. D. Finch. *Analytical Mechanics*. Cambridge Univ. Press, 1998.
- [15] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [16] H. James, N. Nichols, and R. Phillips. *Theory of Servomechanisms*. McGraw-Hill, 1947.
- [17] J. B. Marion and S. T. Thornton. *Classical Dynamics of Particles and Systems*. Harcourt Brace Jovanovich, third edition, 1988.
- [18] J. E. Marsden and M. J. Hoffman. *Basic Complex Analysis*. W. H. Freeman, 1998.
- [19] J. E. Marsden and T. S. Ratiu. *Introduction to mechanics and symmetry*, volume 17 of *Texts in Applied Mathematics*. Springer-Verlag, second edition, 1994.
- [20] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.*, 45(1):3–49, Mar. 2003.
- [21] T. Shinbrot, C. Grebogi, J. Wisdom, and J. A. Yorke. Chaos in a double pendulum. *Am. J. Phys.*, 60(6):491–499, June 1992.
- [22] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley and Sons, 2nd edition, 2005.
- [23] E. M. Stein and R. Shakarchi. *Complex Analysis*. Princeton University Press, 2003.
- [24] R. F. Stengel. *Optimal Control and Estimation*. Dover, 1994.

Index

Page numbers that are underlined denote the main definition of a term. Entries in typewriter type (e.g., `bode`) denote MATLAB commands.

- absolute convergence, 249
- absolute value, 245, 262
- actuator, 1, 9, 50, 128
- aircraft dynamics, 22
- algebraic Riccati equation, 190,
197, 201, 205
 solution, 190
- all-pass transfer function, 126,
230, 232
- `allmargin`, 269
- analytic, 248
- analytic continuation, 171, 249
- area rule, 128, 255
- argument principle, 253, 254
- argument, of a complex number,
29, 246
- asymptotes
 - Bode plot, 96, 97, 99, 101,
102
 - root locus plot, 79, 82, 195
- balanced truncation, 223
- bandwidth, 124, 125, 130, 132,
182, 185, 188
 constraints, 228, 230, 231
- `bandwidth`, 126, 269
- bicycle stabilization, 224
- block diagram, 1, 40
- block upper triangular, 221
- `bode`, 93, 269
- Bode gain-phase relation, 103,
104, 121, 255, 260
- Bode integral formula, 255
- Bode integral formula, 127, 191
- Bode plot, 26, 182, 223
 rules for sketching, 93–100
- boundary, 247
- break-in point, 83
- break-out point, 83
- Butterworth filter, 195
- Butterworth pattern, 195
-
- `c2d`, 268
- cancellation, of pole and zero,
228, 229
- Cauchy's theorem, 252
- causal system, 32
- Cayley-Hamilton theorem, 212,
243
- characteristic equation, 71, 243
- characteristic polynomial, 89,
164, 174, 176, 177, 238,
243
- circular reasoning, 5
- classical control, 67, 143
- closed curve, 251
- closed set, 247
- closed-loop control, 5
- closed-loop poles, 68, 71, 75, 77,
81, 91, 164, 174

- from loop gain, 254, 255
 - optimal locations, 194
 - placement, 174, 183, 185, 186, 192
 - separation principle, 221, 222
- closure, 247
- commuting matrices, 147
- compact set, 262
- companion matrix, 160, 177
- compensator, 101, 222
 - observer-based, 219, 224
- complementary sensitivity
 - function, 223
- complementary sensitivity
 - function, 48, 107, 124, 181
 - from loop gain, 108
- complex conjugate, 245
- complex exponential, 246, 248
- complex exponential inputs, 33
- complex inputs, 28
- complex number, 245
- complex poles, 96
- compromise, 217
- conformal mapping, 263
- connected, 247
- constrained optimization
 - problem, 198
- constraints, 232
- contour integral, 249, 251
- contrapositive, 171, 210
- control design, 181, 182, 186, 222
 - from Bode plots, 129
 - optimal, 189
- control effort, 173, 189, 192
- control objective, 188
- controllability, 164, 165, 201, 216
 - duality, 214
 - matrix, 166, 176
 - test, 166, 171
- controllability Gramian, 171, 171
- controllability matrix, 171
- controllable canonical form, 157, 176, 224
- conv, 267
- convolution, 32, 35, 50
- coordinate change, 160, 176, 241
- corner frequency, 96, 98, 100–103, 105, 131, 187
- correction, from sensors, 215
- cost function, 188, 193, 194, 198
- cost to go, 201
- critically damped, 59
- crossover frequency, 109, 110, 115, 119, 121, 122, 125, 133, 135, 228
 - constraints, 230–232
- cruise control, 1, 3, 9, 12, 36, 49
- ctrb, 269
- curve, 250
- damped natural frequency, 58
- damping ratio, 58, 98, 99
 - and phase margin, 120
- db2mag, 267
- DC motor, 40, 81, 82, 100, 182, 208
- decibels, 26
- degree of controllability, 173
- degree of observability, 211
- deleted neighborhood, 250
- derivative, 248
- derivative feedback, 73, 76, 84
- derivative gain, 72
- determinant, 238, 242, 244
- detrimental, 191
- diagonal matrix, 238
- diagonalizable matrix, 243
- diagonalization, 150, 242
- differential Riccati equation, 200
- Dirac delta function, 31, 34
- disc of convergence, 249
- discrete-time system, 8, 21
- distribution, 31
- disturbance rejection, 6, 7, 45–47, 107, 122, 124, 129, 181, 223, 224, 229
 - with state feedback, 179, 188
- doc, 267

- dot product, 170
- double integrator, 85–87, 109, 146, 150, 202
- double pendulum, 154
- dual system, 213, 216
- dynamics, 8, 9, 50, 53, 67
 - modifying, 58, 174, 175
- effective control, 124, 129
- eig**, 268
- eigenvalues, 153, 221, 237, 239, 241, 242
 - and poles, 155
- eigenvectors, 152, 153, 237, 239, 242
- ellipsoid, 240
- encirclements, 254
 - of -1 point, 113–116, 255
- envelope, 54
- equilibrium point, 14, 15, 18, 20, 23
- equivalent parameterizations, 250
- error convergence, 217
- error, in state estimate, 215
- exp**, 150
- expm**, 150, 160
- exponential inputs, 28
 - relation to sinusoidal inputs, 29
- exponential, complex, 252
- external disturbances, 12, 215, 217
- eye**, 267
- F-16, 2
- fast pole, 185, 192
- feedback
 - beneficial effects, 124, 129
 - detrimental effects, 124, 127
 - effect on dynamics, 71
- feedback**, 268
- feedback loop, 1, 43
- filtfilt**, 33
- final value theorem, 39
- first-order system, 57, 62, 75
- flight control, 2, 22
- forward gain, 43, 47
- freqresp**, 269
- frequency domain, 34, 67
- frequency response, 26
- full-state feedback, 219
- fundamental limitations, 228, 229, 233
- gain, 26, 27, 112
- gain margin, 119, 181, 184, 191, 232
- gang of four, 49
- guaranteed robustness, 191, 193
- guaranteed stability, 222
- Hamiltonian matrix, 159
- help**, 267
- Hermitian, 239
- holomorphic, 247, 262
- identity matrix, 238
- image, 167
- impossible to control, 81, 228, 229, 234
- impulse**, 269
- impulse response, 31, 53, 55, 57, 151, 155, 225
- inexact cancellation, 229
- initial**, 144, 150, 269
- input, 9, 12, 17, 45
- integral feedback, 69, 74, 85, 90, 130
- integrator, 11, 30, 69, 95, 109, 125, 158
- interconnection, 30, 40
 - and Bode plots, 94
 - state-space systems, 21
- interior point, 247
- inverse of a matrix, 238
- Jacobian matrix, 15
- Kalman equality, 205
- lag compensator, 101, 102

- Lagrange multiplier, 198
- Lagrangian mechanics, 19
- lane change, 186
- Laplace transform, 33
- lead compensator, 86, 87, 101, 101
- Lie algebra, 148
- Lie group, 148
- lightly damped poles, 88
- limit point, 247
- linear quadratic regulator, 188, 222
 - design, 192
 - guarantees, 190–192
 - solution, 190, 195, 201
- linear system, 241
- linear time-varying system, 200
- linearization, 13, 154
 - 1d example, 15
- linspace, 267
- load sensitivity function, 49
- logarithm, complex, 252, 261
- logspace, 267
- loop gain, 43, 47, 76, 92, 124, 224
- loop transfer function, 105, 107, 108, 113, 121, 181, 191, 223
 - for state-space system, 181
- loopshaping, 121
- low-pass filter, 135
- lqe, 269
- lqg, 269
- lqr, 190, 269
- lsim, 269
- LTI system, 144, 163, 195, 207
 - solution, 151
- Lyapunov equation, 173, 196, 196, 197
- mag2db, 267
- magnitude, 246
- margin, 269
- matrix exponential, 145, 145, 146
 - computing, 148
 - Laplace transform, 152
 - properties, 146
- maximum modulus principle, 262
- meromorphic function, 250, 253
- minimum phase, 94, 230–232
- minimum-energy controller, 194
- minimum-energy input, 173
- minimum-phase systems, 100, 103
- model inaccuracies, 214
- model reduction, 222
- modern control, 143
- modes, 153
- modulus, 246, 262
- monic polynomial, 174
- multiple-valued function, 252
- multiplication, of complex numbers, 30, 246
- natural frequency, 27, 58
- negative feedback, 43
- neighborhood, 247
- Newton-Raphson iteration, 199
- nilpotent, 148
- noise sensitivity function, 49
- non-minimum-phase systems, 105
- nondimensionalization, 20, 58, 167
- nonlinear system, 13
 - optimal control, 198
- normalization, 238
- notch filter, 88
- nyquist, 142, 269
- Nyquist contour, 112, 112, 254, 254
- Nyquist plot, 110, 112, 112, 181, 182, 191, 192, 223, 232
 - distance to -1 point, 120
 - from Bode plot, 114
- Nyquist stability criterion, 110, 113, 115, 121, 233, 252, 255
 - example, 116
 - simplified, 113, 115
- observability, 207, 208, 235
 - condition, 190, 197

- duality, 214
- matrix, 211
- test, 211
- observability Gramian, 210
- observable canonical form, 160
- observable pair, 208
- observer, 207, 214, 215, 220, 222, 224
 - poles, 217, 221
- observer gains, 215
- observer-based feedback, 191, 219, 219
- obsv, 269
- open disc, 246
- open set, 247
- open-loop control, 4
- open-loop poles, 68, 71, 75, 174, 254, 255
- operational amplifier, 11, 158
- optimal control, 173, 188, 194
 - nonlinear system, 198
- order, of zero or pole, 250
- orthogonal matrix, 148, 159
- orthogonal vectors, 170, 239
- oscillations, 57
- output, 9, 12, 17, 45
- output feedback, 207, 219
- overdamped, 59
- overshoot, 60, 62, 187
- parameterization, 250
- partial fractions, 56
- PD control, 72, 86, 133
- peak, in sensitivity function, 227, 234, 264
- pendulum, 10, 13
 - linearization, 13, 18
- performance, 126, 181, 184, 189, 223, 224
 - from loop gain, 109
 - from loop transfer function, 121
- phase, 26, 104, 111, 246
- phase crossover frequency, 115, 119
- phase lag, 102, 103, 131, 230–232
- phase lead, 101, 101
- phase margin, 119, 121, 132, 133, 142, 181, 184, 191, 224, 230, 232
 - and damping ratio, 120
- PI control, 82
- PI controller, 102, 103, 131
- PID controller, 70
- place, 176, 216, 269
- plant, 45
- point singularity, 250
- polar form, 246
- pole, 268
- pole locations, 62
- pole placement, 164, 186, 192, 202, 216, 222
 - choosing pole locations, 195, 202, 217
 - example, 175
 - optimal locations, 194
 - optimal pole locations, 193
 - theorem, 174, 177
 - where to choose poles, 217
 - where to place poles, 225
 - with fast pole, 185
 - with slow zero, 188
- poles, 39, 53, 55, 95, 250
 - cancellation, 228
 - complex, 96, 99, 100, 153
 - fast, 185
 - near zeros, 233
 - real, 105
 - right half plane, 232–234
 - right-half-plane, 105, 106, 232
- polyfit, 268
- polyval, 268
- positive-definite matrix, 171, 188, 190, 198, 200, 210, 240, 244
- positive-semidefinite matrix, 171, 210, 240
- power series, 171, 248, 249

- principles of feedback, 3, 7, 8, 58
- proper transfer function, 56, 78, 156
- proportional feedback, 67, 72, 75, 81, 84, 91, 232
- proportional gain, 72
- proportional-integral control, 65
- pzmap, 268
- quadratic form, 240, 243
- quadratic function, 188
- radius of convergence, 249
- range, 167
- rank, 213
- rank**, 268
- rational function, 53, 248, 263
- reachable subspace, 167, 170
- reference tracking, 5, 45–47, 69, 91, 107, 129, 181, 182, 186, 204, 223
 - from Bode plot, 187
 - with state feedback, 179, 188
- reflection, 240
- region, 247
- regularity, 248
- relative degree, 128, 191, 195
- residue, 250, 253
- residue**, 268
- resonant peak, 98–100
- resonant valley, 98
- right-half-plane pole, 231, 233
 - near right-half-plane zero, 228
- right-half-plane zero, 81, 233
- rise time, 61, 62
- rlocus**, 77, 269
- robustness, 226
 - from Nyquist plot, 234
 - to changes in the system, 5, 49, 118, 124, 183, 185, 188, 192, 223, 224, 227
- roll off, high frequency, 49
- roll off, high frequency, 27, 128, 225
- root-locus diagram, 74, 77
 - rules for sketching, 77–80
- root-locus form, 77, 90, 194
- root-locus plot, 111, 194
- rotation, 149
- satellite, 148
- second-order system, 57, 60, 75, 83, 96
- seesaw
 - control of, 192
 - dynamics, 19
 - nondimensionalization, 20
- sensitivity function, 6, 92
- sensitivity function, 48, 107, 121, 124, 181, 184, 223
 - constraints, 264
 - from loop gain, 108
 - from Nyquist plot, 122, 123
 - peak, 185, 234
- sensor, 1
- sensor measurements, 215
- sensor noise, 47, 49, 69, 92, 107, 109, 121, 184, 217, 226
- separation principle, 221, 222, 222
- servo motor, 182
- settling time, 61, 62
- sifting property, 31
- similarity transformation, 178, 242
- simple curve, 251
- simple pole, 54, 250
- simple zero, 250
- simply-connected region, 251
- sinusoidal inputs, 25, 33
 - relation to exponential inputs, 29
- skew-symmetric matrix, 148, 159, 243
- slope at crossover, 121
- slope, of Bode plot, 262
- slow zero, 187, 188, 194, 227
- smooth parameterization, 250
- spring-mass system, 9, 12, 26, 31, 36, 57, 70, 72, 156

- double, 10, 21
- forced at wall, 156
- sprung beam, 212
 - controllability, 167, 201
 - observability, 209, 212, 214, 235
 - observer design, 217
- ss, 178, 268
- ssdata, 268
- stability
 - closed-loop, from Bode plot, 111
 - of closed-loop system, 110, 111, 113, 116, 121
 - of second-order systems, 64
- stability augmentation, 2
- stability margin, 120, 123, 134, 181, 184, 188, 192, 224, 226, 228
- stabilization, 222
- stable, 54
- standard 2nd-order system, 58
- state, 11
- state estimate, 207, 214, 218, 219
- state feedback, 163, 174, 182, 220, 222
- state-space form, 11
- state-space realization, 144, 155, 157, 178, 182, 186, 215, 220, 222
- steady-state error, 69, 92, 130, 132, 134
- step, 269
- step function, 31, 34, 59
- step input, 59
- step response, 50, 59, 60, 62, 63, 183, 186
- strictly proper transfer function, 78, 113, 156
- superposition, 13, 25, 26, 28, 32, 46, 47, 159, 208, 210
- symmetric matrix, 239, 240, 243
- symmetric root locus, 194, 194, 195
- symplectic matrix, 159
- systematic design, 223
- tachometer, 208
- Taylor series, 15, 145, 231
- terminal cost, 198
- tf, 268
- tfdata, 268
- time delay, 8, 21, 35, 228, 230, 231
- time-domain specifications, 59, 62
- trace, 242, 244
- tracking error, 5, 6, 7, 45, 47, 109, 122, 124
 - from Bode plot, 92
- transfer function, 29, 33–35
 - closed-loop, 107
 - from a state-space realization, 38
 - open-loop, 106
- transient response, 187
- transients, 188
- triangular matrix, 238
- two-degree-of-freedom design, 45
- uncontrollable state, 170, 171
- uncontrollable system, 165
- underdamped, 59
- unit ball, 240
- unit disc, 263
- unobservable state, 208, 208
- unobservable states, 209, 212, 235
- unobservable system, 208
- unstable, 54, 175
- unstable dynamics
 - modifying, 164, 165
- unstable poles, 194, 228
- unstable system, 8
- vector margin, 120
- vehicle steering, 185, 194
- waterbed effect, 128
- weighted average, 104, 262
- weighting function, 104

zero, 268

zero-frequency gain, 38, 100–102,
180

zero-input solution, 144, 145

zeros, 53, 96, 103, 249

 cancellation, 228

 complex, 98–100, 217

 effect on dynamics, 63

 near poles, 228, 233

 right-half-plane, 63, 105, 228,
 230, 232–234

 slow, 187